

**AN ADVANCED PROCESS MANUFACTURING
SYSTEM - DESIGN AND APPLICATION TO A FOOD
PROCESSING PILOT PLANT**

by

LIU Zhen Hai

July 1995

A thesis submitted for the degree of Doctor of Philosophy of the University of London
and for the Diploma of Membership of the Imperial College

Department of Chemical Engineering and Chemical Technology
Imperial College of Science, Technology and Medicine
London SW7



ABSTRACT

The inherent variability and complexity of batch processes increases their difficulty of operation. In recent years, considerable effort has been made in developing methods and generic computer-aided tools to address various batch operational functions, such as process planning and scheduling, supervisory control and scheduling, dynamic simulation, and in advanced control techniques for nonlinear processes. However, it is generally acknowledged that the potential for flexibility, increased production, consistency of product and reduced operating costs will only be fully realised by considering a manufacturing process as a whole, and by applying these individual techniques in a properly integrated and coordinated manner. Furthermore, testing, validation and implementation of an integrated system is an intensive and expensive task and the lack of a unified environment for such a system is evident.

This work deals with the design, implementation and operation of an advanced manufacturing system for a multipurpose batch pilot plant at Imperial College which supports various on-line and off-line control and operations functions in an integrated platform. Hierarchical and modular models of the batch plant and enzymatic starch conversion processes (including cleaning-in-place operations) at various levels of detail are developed and validated by experiment. Coordination of control actions and mapping of production control models used at the different levels are established to ensure that information and local decisions can be transferred/shared among various application programs and/or control systems.

Off-line production planning and scheduling based on mathematical programming techniques is used to generate optimal production schedules. These optimal schedules are then used as inputs to a supervisory control system for on-line production control and monitoring. Due to the strong coupling between each level of control (*e.g.* supervisory, sequential and primitive continuous/discrete), the development of flexible and effective control sequences and model based nonlinear controllers is addressed. Tighter control of unit operations becomes more important in the integrated operation environment for the local suppression of disturbances and to provide a robust performance in response to wide ranging process variations. A partitioning space approach is presented which allows various nonlinear control techniques to be used for each nonlinear subsystem. This results in a more flexible control structure and better control performance. The effectiveness of the approach is demonstrated by the temperature control of a multipurpose batch reactor.

The nonlinear controllers, new control sequences and supervisory control procedures are developed and tested by simulation, utilising the operations support system environment developed in this work. The work also demonstrates a practical implementation of the proposed system on a multipurpose batch pilot plant.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Professor Sandro Macchietto, for his continuous encouragement, excellent guidance and support throughout this work.

I am very grateful for the help of Terrence Crombie in supporting me with SUPERBATCH. I must also thank Dr. Steve Walsh for his useful advice and comments on Chapter 5, and Dr. Nilay Shah for his support of *gBSS* package. I also wish to thank Nick Alsop for patiently reading the manuscript.

I would also like to thank all my friends in the Centre for Process System Engineering. In particular, I would like to mention Iqbal, Arturo, Antoine, Kamal, Yan, Xia and Min. I also wish to thank all the members of I.C. CSSA badminton club, who make the life at the I.C. more enjoyable.

This work would not have been possible without the financial support from APV Baker Ltd. and the Centre for Process Systems Engineering at Imperial College.

Finally, I would like to thank my wife, Lin Lin, for her unwavering support, love and encouragement throughout this work. I also wish to thank my family for their love and support.

TABLE OF CONTENTS

Abstract 2

Acknowledgements 4

List of Figures 8

List of Tables 11

Nomenclature 12

Chapter 1 Introduction 15

1.1 Introduction 15

1.2 Objective 16

1.3 Thesis Outline.....18

Chapter 2 Literature Review 19

2.1 Characteristics of Batch Processes.....19

2.2 Batch Process Operations.....20

2.2.1 Production Planning and Scheduling 22

2.2.2 Supervisory Control 24

2.2.3 Sequence Control 27

2.2.4 Model Based Control of Batch Processes.....29

2.2.4.1 Model Predictive Control.....30

2.2.4.2 Differential Geometric Approaches.....31

2.2.4.3 Applications to Batch Processes.....33

2.3 Integrated Operation of Batch Processes.....34

2.4 Operations Support Systems 37

2.5 Summary and Conclusions.....38

Chapter 3 Advanced Processes Manufacturing System - Design 41

3.1 Advanced Operations Support System - Design and Implementation.....41

3.1.1 The Batch Pilot Plant 44

3.1.2 Control Hardware Configuration 46

3.1.3 Communication Design.....48

3.2 Modelling of Batch Process Operations.....49

3.2.1 Model Aggregation 49

3.2.1.1 Transfer Tasks.....50

3.2.1.2 Sequential Tasks 52

3.2.2	Design of Control Sequences for Flexibility	55
3.2.3	Models Hierarchy and Coordination	57
3.3	Summary and Conclusions.....	58
Chapter 4	Modelling a Multipurpose Batch Pilot Plant	60
4.1	Model of Equipment	60
4.2	Model of Processes - Recipe Models.....	65
4.2.1	Process Descriptions.....	65
4.2.2	State-Task-Network Models	66
4.3	Supervisory Control Procedures - Control Recipe Models.....	69
4.4	Control Sequences and Modelling.....	73
4.5	Model of Plant Dynamics.....	76
4.5.1	Model Validation and Parameter Estimation.....	80
4.6	Summary	81
Chapter 5	Partitioning Space Approach for Nonlinear Process Control	83
5.1	Control Based on Exact Input-output Linearization	84
5.2	Partitioning Space Approach for Nonlinear Process Control.....	85
5.2.1	Characteristics of Cascade Control Structure	85
5.2.2	Partitioning Space	86
5.2.2.1	Nonlinear System Partitioning	89
5.2.3	Control System Design Based on Partitioned System	90
5.2.4	Handling Constraints.....	92
5.2.5	Design Procedure	94
5.2.6	Illustrative Example	95
5.3	Multipurpose Batch Reactor System	99
5.3.1	Batch Reactor System	100
5.3.2	Characteristics of the Reactor System.....	102
5.4	Controller Design and Synthesis	103
5.4.1	Synthesis of Control Law	104
5.4.2	Estimation of the Heat of Reaction	108
5.4.3	Disturbance Decoupling	109
5.4.4	Coordination Rule	110
5.5	Simulation and Experimental Results.....	111
5.5.1	Performance Criteria	112
5.5.2	Controller Tuning.....	112
5.5.3	Servo Response.....	113

5.5.4	Estimation of Q	115
5.5.5	Tracking Control Response	116
5.5.5.1	PI Control.....	116
5.5.5.2	GMC and PI Control	119
5.5.5.3	GLC and PI Control.....	122
5.5.5.4	GMC Control	124
5.5.6	Summary and Discussions.....	126
5.6	Conclusions	128
Chapter 6	Sequence Control of Batch Processes	131
6.1	Sequence Control of Cleaning-In-Place Operations	132
6.1.1	Cleaning Based On Fixed Time Policy	132
6.1.2	Cleaning Policy Based on Volume	137
6.1.3	Cleaning of Parallel Vessels	139
6.2	Emergency Control and Recovery Procedure.....	142
6.2.1	Emergency Control Sequence.....	143
6.2.2	Model Based Override Control.....	145
6.3	Conclusions	148
Chapter 7	Production Scheduling and Supervisory Control.....	150
7.1	Operator Controlled Operation	150
7.2	Production Planning and On-line Supervisory Control	153
7.2.1	Off-line Optimal Production Planning and Scheduling	154
7.2.2	SUPERBATCH Production Input.....	158
7.2.3	Supervisory Control of Batch Plant	160
7.3	Alternative Operating Procedure	170
7.4	Summary	176
Chapter 8	Conclusions and Future Work	178
8.1	Future Work	180
References		181
Appendices		191
A	Implementation of Nonlinear Controllers	191
B	Implementation of Supervisory Control	194

LIST OF FIGURES

Figure 2.1	Integrated management and montrol system	36
Figure 3.1	Architecture of the advanced operations support system	43
Figure 3.2	Multipurpose batch pilot plant	45
Figure 3.3	Schematic of the ACCOS information management and control system	47
Figure 3.4	A schematic of Control Link	48
Figure 3.5	STN model without consideration of material transfer	50
Figure 3.6	Modelling material transfers: (a) not modelled; (b) modelled explicitly; (c) modelled as "hidden" task	51
Figure 3.7	STN input file in gBSS for modelling material transfers	51
Figure 3.8	STN representation of the example process with aggregated tasks	53
Figure 3.9	Scheduling results for the example process with task aggregation	53
Figure 3.10	Scheduling results for the example process without task aggregation	54
Figure 3.11	Control sequence structure	56
Figure 3.12	Model hierarchy	58
Figure 4.1	Unit level batch pilot plant description	61
Figure 4.2	Common resource utilization during the transfers/recirculations between the units	63
Figure 4.3	Hierarchical equipment models: (a) aggregated unit TANK-3; (b) sparge system submodule; (c) TANK-3 model at the item level	64
Figure 4.4.a	STN representation of the Dextrins process	67
Figure 4.4.b	STN representation of the Cyclodextrins process	67
Figure 4.4.c	STN representation of the cleaning-in-place (CIP) process	67
Figure 4.5	USTN representation for unit TANK-1	69
Figure 4.6	An elementary task for opening a control valve in the simulation model	73
Figure 4.7	Jacket recirculating flow characteristics	78
Figure 5.1	Partitioning of nonlinear systems	90
Figure 5.2	A cascade control structure for nonlinear process control	92
Figure 5.3	Partitioning space design procedure	94
Figure 5.4	Two noninteracting tanks in series	96
Figure 5.5	GLC control with perfect model	97
Figure 5.6	GLC control with mismatch on a model parameter	97
Figure 5.7	Level responses with a cascaded GLC and PI controller	99
Figure 5.8	Reference temperature profile for the starch degradation process by α -amylase	104
Figure 5.9.a	Cascaded GMC/GLC and PI controller structure	107

Figure 5.9.b	Cascaded GMC/GLC controller structure	107
Figure 5.10	Servo control response of the PI controller when heating the reactor from room temperature to 65 °C (experiment 1)	114
Figure 5.11	Servo control response of the GMC controller when heating the reactor from room temperature to 55 °C (experiment 2)	114
Figure 5.12	Servo control response of the GMC controller when heating the reactor from room temperature to 85 °C (experiment 3)	115
Figure 5.13	Estimated heat of reaction profile in starch gelatinization and degradation phases	116
Figure 5.14	Reactor temperature responses under PID control - water jacket control to 65 °C (experimental results)	117
Figure 5.15	Reactor temperature responses under PID control - Sparge heating to 105 °C, then water jacket control to 85 °C (experiment)	118
Figure 5.16	Cascaded GMC/PI controller -Water jacket control to 65 °C (experiment)	119
Figure 5.17	Cascaded GMC and PI controllers - Sparge heating to 105 °C, then water jacket control to 85 °C (experiment)	120
Figure 5.18	Cascaded GMC and PI controllers - Sparge heating to 105 °C, then water jacket control to 65 °C (experiment with water)	121
Figure 5.19	Effect of constraint on the controllers performance (simulation)	122
Figure 5.20	Cascaded GLC/PI controller with and without error on the measurement of reactor weight - initial sparge heating stage (experiment)	123
Figure 5.21	Cascade GLC and PI control with and without error on the measurement of reactor weight - entire operation (experiment)	124
Figure 5.22	Cascaded GMC controller - sparge heating to 105 °C, followed by water jacket control to 55 °C (experiment and simulation)	125
Figure 5.23	Cascaded GMC controller - sparge heating to 105 °C, followed by water jacket control to 85 °C (experiment and simulation)	126
Figure 6.1	T1 cleaning simulation based on fixed-time policy. A - initial CIP service; B - pre-rinse; C - CIP service; D - detergent clean; E - final rinse	135
Figure 6.2	Detergent temperature response during the CIP service	136
Figure 6.3	T1 cleaning simulation based on volume policy. A - initial CIP service; B - pre-rinse; C - detergent clean; D - final rinse	138
Figure 6.4	Cleaning circuits for T1 and T2 in the pilot plant	139
Figure 6.5	Simulation of parallel cleaning of T1 and T2	140
Figure 6.6	Parallel cleaning of T1 and T2 - experiment 1. A - initial CIP service; B - pre-rinse; C - detergent clean; D - final rinse	141
Figure 6.7	Tree structure of the control sequences in the temperature control of the	

		10
	reactor through heat exchanger HE2	143
Figure 6.8	Reactor and jacket temperature responses with current emergency control procedure - experimental result	144
Figure 6.9	Override control scheme for emergency recovery	145
Figure 6.10	Reactor temperature control with model based emergency recovery procedure - experimental result	148
Figure 7.1.a	Continuous hold-up measurements over the entire production	151
Figure 7.1.b	Temperature responses of the reactor and jacket over the entire production	151
Figure 7.2	Case Ia - optimal production schedule for producing a single product with final CIP	155
Figure 7.3	Case Ib - optimal production schedule for producing a single product with final CIP	156
Figure 7.4	Case II - Optimal production schedule for producing two products with intermediate and final CIP	156
Figure 7.5	Utility profile for the Case II	157
Figure 7.6	Detailed supervisory control system configuration linked with simulator	161
Figure 7.7	Case Ia - process schedule at time 12:00	163
Figure 7.8	Case Ib - process schedule at 12:27	163
Figure 7.9	Case Ia - continuous holdup measurements over the entire operation	164
Figure 7.10	Case Ia - temperatures of the reactor and jacket over the entire operation	164
Figure 7.11	Case Ib - SUPERBATCH schedule at time 12:00	168
Figure 7.12	Case Ib - SUPERBATCH schedule at time 17:04, after vessel T1 failure	168
Figure 7.13	Case Ib - SUPERBATCH schedule at time 17:17 after reallocation of same tasks	169
Figure 7.14	Case II - two products, SUPERBATCH schedule of at time 12:00	169
Figure 7.15	Case II - two products, SUPERBATCH schedule at start time 13:08	170
Figure 7.16	Decomposition of the reaction task into two subtasks	171
Figure 7.17	Case Ic - off-line optimal schedule with the new operating procedure	172
Figure 7.18	Utility profile with the new operating procedure	172
Figure 7.19	Case Ic - SUPERBATCH schedule with the new operating procedure at time 12:00	173
Figure 7.20	Case Ic - SUPERBATCH schedule with the new operating procedure at time 12:30	174
Figure 7.21	Case Ic - continuous holdup measurements over the entire production with the new operating procedures	174
Figure 7.22	Case III - SUPERBATCH schedule at time 12:01 with new product recipe	175

LIST OF TABLES

Table 2.1	Summary of applications of nonlinear control techniques to batch processes	34
Table 3.1	Aggregated tasks and processing time	52
Table 3.2	Computational details for the solution of the example process	55
Table 4.1	Summary of equipment and suitability	68
Table 4.2	Summary of STN tasks utility information	68
Table 4.3	Master procedure for the dextrins process	71
Table 4.4	Real-Time control model for the dextrins process	72
Table 4.5	T3 sparge heating control sequence in PARACODE	74
Table 4.6	Simulated sparge heating control sequence in gPROMS	75
Table 4.7	Mapping of STN tasks/control phases/control sequences for the dextrin process	82
Table 5.1	Reactor temperature control range under each control scheme.	103
Table 5.2	Control algorithms under the sparge heating control	105
Table 5.3	Control algorithms under water jacket control	108
Table 5.4	Control system structures for the starch processes	111
Table 5.5	Controller settings in experiments	113
Table 5.6	Quantitative performance of controllers under each of control configuration of Table 5.4	127
Table 6.1	Functional specification of the control sequence for cleaning Tank 1	134
Table 7.1	Summary of operations for starch degradation process with CIP - production of one batch under the operator control	152
Table 7.2	MILP computation statistics	158
Table 7.3	SUPERBATCH production plan transformed from the MILP schedule from Case Ia	159
Table 7.4	SUPERBATCH production plan transformed from the MILP schedule in Case Ia	159
Table 7.5	Batch summary corresponding to Figure 7.7	165
Table 7.6	Batch summary corresponding to Figure 7.8	166

NOMENCLATURE

A	Effective heat transfer area between the reactor and the jacket		(m^2)
A_{10}	Total heat transfer area of the reactor cone part	0.4450	(m^2)
A_{20}	Maximum heat transfer area of the reactor cylinder part	0.4634	(m^2)
A_p	Heat transfer area of product vessels	0.6437	(m^2)
A_t	Heat transfer area of raw material tanks	0.6437	(m^2)
C_p	Heat capacity of the reactant	4.20	(KJ/Kg $^{\circ}$ C)
C_{pj}	Heat capacity of water in the jacket	1.0	(KJ/Kg $^{\circ}$ C)
C_{ps}	Heat capacity of HE2 inlet steam (3 Bar)	2.221	(KJ/Kg $^{\circ}$ C)
C_p^i	Heat capacity of inlet steam i	1.0	(KJ/Kg $^{\circ}$ C)
C_{pt}	Heat capacity of material in vessels	4.20	(KJ/Kg $^{\circ}$ C)
C_{in}^i	Concentration of inlet steam i		(%)
C_d	Concentration of detergent in CIP tank		(%)
C_s	Concentration of starch slurry		(Kg/l)
F_{cw}	Volumetric flow rate of cooling water		(m^3/s)
F_j	Volumetric flow rate of recirculating water in the jacket		(m^3/s)
F_{in}^i	Volumetric flow rate of inlet stream i		(m^3/s)
F_{out}	Volumetric flow rate of outlet stream		(m^3/s)
F_{rc}	Jacket recirculating flow constant	1.1266E-3	(m^3/s)
H_{sp}	Enthalpy of sparge steam (5 bar)	2748.6	(KJ/Kg)
K_{he}	HE2 constant	284.72	(s/ m^3)
K_m	Michaelis-Menten constant	0.08081	(Kg/l)
k_1^{cw}	Cooling water valve constant	1.32539E-3	(m^3/s %)
k_2^{cw}	Cooling water valve constant	2.04235E-5	(m^3/s %)
k_1^{sp}	Sparge steam valve constant	4.6668E-4	(Kg/s %)
k_2^{sp}	Sparge steam valve constant	1.01173E-3	(Kg/s %)
k_3^{sp}	Sparge steam valve constant	2.01099E-3	(Kg/s %)
k^{he}	HE2 steam valve constant	0.09799	(Kg/s)
k_j	Jacket recirculating flow constant	0.29565E-6	(m^3/s %)
k	Total number of inlet streams to product vessels	2	
m	Total number of inlet streams to raw material tanks	2	
n	Total number of inlet streams to CIP tank	2	
P_{dex}	Concentration of dextrans		(Kg/l)
S_d	Weight of dry starch		(Kg)

S_w	Weight of process water		(Kg)
T_{cin}	HE2 cold stream inlet temperature		($^{\circ}C$)
T_{cw}	Cooling water temperature		($^{\circ}C$)
T_{hin}	HE2 hot stream inlet temperature		($^{\circ}C$)
T_{in}^i	Temperature of inlet stream i		($^{\circ}C$)
T_j	Jacket temperature		($^{\circ}C$)
T_{jin}	Jacket inlet temperature		($^{\circ}C$)
T_p	Temperature in product vessels		($^{\circ}C$)
T_r	Reactor temperature		($^{\circ}C$)
T_{rm}	Room temperature		($^{\circ}C$)
T_t	Temperature in raw material tanks		($^{\circ}C$)
Q	Heat rate of the reaction		(KJ/s)
Q_{rp}	Heat rate of the reaction in product vessels		(KJ/s)
U_{10}	Heat transfer coefficient of the reactor cone part	6.165E-2	(KJ/m ² s $^{\circ}C$)
U_p	Heat transfer coefficient of product vessels	0.0009	(KJ/m ² s $^{\circ}C$)
U_{sp}	Heat transfer coefficient during the sparge heating	9.696E-7	(KJ/m ² s $^{\circ}C$)
U_t	Heat transfer coefficient of raw material tanks	0.0012	(KJ/m ² s $^{\circ}C$)
U_w	Heat transfer coefficient between the reactor and the jacket	9.952E-1	(KJ/m ² s $^{\circ}C$)
u_c	Output of secondary controller		(%)
V_{jf}	Jacket working volume		(m ³)
V_d	Working volume of CIP tank		(m ³)
V_{max}	Maximum velocity of the reaction	4.574E-2	(Kg/l s)
V_p	Working volume of product vessels		(m ³)
V_r	Reactor working volume		(m ³)
V_t	Working volume of raw material tanks		(m ³)
W_s	HE2 steam mass flow rate		(Kg/s)
W_{ss}	Sparge steam mass flow rate		(Kg/s)
x^{cw}	Cooling water valve position		(%)
x^s	Steam valve (CV1-4) position		(%)
x^{sp}	Sparge steam valve position		(%)
α	HE2 steam valve constant	77.9	
ρ	Reactant density	1000.0	(Kg/m ³)
ρ_j	Water density in the jacket	1000.0	(Kg/m ³)
ρ_t	Density of mixture in raw material tanks	1000.0	(Kg/m ³)
ρ^i	Density of inlet stream i	1000.0	(Kg/m ³)

τ_{jf}	<i>Time constant of exponential filter for the jacket</i>	60.0	(s)
τ_{qf}	<i>Time constant of exponential filter for the heat of reaction</i>	60.0	(s)
τ_s	<i>Sampling frequency of exponential filters</i>	10.0	(s)
τ_{rf}	<i>Time constant of exponential filter for the reactor</i>	60.0	(s)
τ_{sp}	<i>Jacket time constant during sparge heating</i>	230.0	(s)
Φ_{he}	<i>Heat exchanger factor</i>		

Chapter 1

Introduction

1.1 Introduction

Batch processing is widely used in the chemicals, food and beverages, and pharmaceuticals industries, which operate typically in highly competitive markets (Sawyer, 1993). The interest of batch processing is reflected by a table in Sawyer's (1993) book which shows that 80% of pharmaceutical plants, 65% food and beverage plants and 45% chemical plants are operated in batch mode.

The key driving forces for this mode of manufacture, in addition to inherent technical reasons (*e.g.* ability to handle solid and viscous materials), are the increasing competition on product markets (increased variety and price competition), shortening product life cycles and the competitive advantages offered by increased responsiveness to irregular customer demands. It has been observed that where production requirements are uncertain, batch plants are typically more efficient and flexible (Rippin, 1991). Due to the inherent flexibility in operation, batch plants are well suited to deal with such uncertain market conditions and to meet specific customer requirements. They are typically easier to reconfigure or adapt in order to allow for production changeover. However, such flexibility also results in an increased complexity of operation, from planning, scheduling and control perspectives. In particular:

- several products may be made in a single plant and a given product might follow different routes through the plant at different times. There are many degrees of freedom in operation due to this flexibility;
- the complexity of control systems to support flexible plant operations is often considerably higher than for simpler and single product plants;

- tighter control of unit operations over a wide range of operating conditions is required to deal with frequent process variations (*e.g.* batch sizes, raw materials, process conditions, *etc.*);
- various control activities (*e.g.* supervisory, sequential, continuous and discrete control) are needed to operate a whole manufacturing plant. These control activities must be carried out in a coordinated manner.

In recent years, significant progress has been made in control systems hardware with improved reliability and flexibility, and in advanced sensors to enhance plant and process monitoring. In parallel with advances in hardware, considerable efforts have also been made in developing methods and generic computer-aided tools to address various issues relating to batch operation, such as process planning and scheduling (Egli and Rippin, 1986; Shah, 1992; Reklaitis, 1992), supervisory control and scheduling (Cott and Macchietto, 1989), dynamic simulation (Czulek, 1988; Clark and Kuriyan, 1989; Barton and Pantelides, 1991), and in advanced control and estimation techniques for nonlinear processes (Kravaris and Chung, 1987; Lee and Sullivan, 1988). However, it is generally acknowledged that the potential for flexibility, increased production, consistency in product and reduced operating costs will only be fully realised by considering a manufacturing process as a whole, and by applying these individual techniques in a proper integrated and coordinated manner (Macchietto, 1993; Sawyer, 1993). A number of integration schemes have been proposed at various levels of detail and scope of integration (Ranky, 1986; Williams, 1987; Cott and Macchietto, 1989; Venkatasubramanian and Stanley, 1993). However, it is fair to say that a generally applicable solution is still lacking.

Furthermore, the testing, validation and implementation of integrated operations and control systems are intensive and expensive tasks due to the complexity of batch plant operations. A suitable development environment is clearly desirable to support the validation and testing of all manufacturing control procedures, to provide feedback at an early stage in the development cycle of new plant, process, product and control designs, and ultimately, to achieve a safe and effective on-line implementation of all production control systems.

1.2 Objective

This work deals with the design, implementation and operation of an advanced manufacturing system for multipurpose batch plants to perform various on-line and off-line control and operations support functions in an integrated platform. Two main goals are to develop:

- a *production environment* capable of supporting various advanced production control functions in an integrated way (from process planning and scheduling, to supervisory control and advanced control techniques), aimed at achieving high operations efficiency, product quality, responsiveness to change, *etc.*
- a *development environment* aimed at increasing the efficiency of an engineering team by enabling the use of advanced tools and techniques during the development and implementation of the manufacturing control procedures.

In order to provide a generic solution to a general problem, such a system must be driven by models and utilise, as far as possible, generic solution tools.

Ultimately, the goal of this work is to demonstrate the practical implementation of the design proposed on a multipurpose batch pilot plant at Imperial College.

The benefits from this work are summarised below:

- on-line scheduling and supervisory control strategies can be developed and tested on a simulator before implementation on the real plant. Any potential problems associated with implementation can be identified at the simulation stage;
- development and validation of control sequences can be carried out on the simulator rather than during plant commissioning. A number of control, safety assessment and process improvement studies can be carried out which lead to improved plant operation efficiency;
- unit operations can be significantly improved by applying model based control techniques. This results in an improved tracking control performance, safer operation and improved product quality;
- Plant resources can be more effectively utilised by applying appropriate production planning and scheduling technologies;
- New product recipes and reformulations can be developed while assessing their impact on manufacturability;
- Operator training can be carried out on the simulator to reduce the risk and interruption to the plant operation.

Although the study is based on a specific batch pilot plant, the unified application framework proposed is suitable for a large class of batch plants.

1.3 Thesis Outline

A brief literature survey of the characteristics of batch processes, their operation and control is given in *Chapter 2*.

The design of an advanced manufacturing control system for a multipurpose batch pilot plant is described in *Chapter 3*. Detailed design considerations, the integration of various batch process engineering functions and a model development method to achieve proper data aggregation/disaggregation at each control level are discussed.

Chapter 4 describes in detail the models of the batch pilot plant and of its operations and model validation.

Chapter 5 presents a novel partitioning space approach for the control of nonlinear processes, after discussing the limitations of available control techniques. The new approach is applied to the temperature control of a multipurpose batch reactor. A number of control configurations are investigated using various control techniques. The controller performance under each control configuration is compared using both simulation and experimentation.

The sequential control aspects of the batch plant are discussed in *Chapter 6*. The dynamic models developed in Chapter 4 are used to improve a number of control sequences. In particular, a procedure for parallel cleaning of two vessels is developed and implemented on the plant. An emergency recovery procedure is developed for the temperature control of the reactor using a model based override control. Simulation and experimental results are presented.

In *Chapter 7*, production scheduling and supervisory control of the batch pilot plant are discussed. An off-line scheduling model for the optimal production of two starch based processes using a mathematical programming approach is presented. The optimal schedules are then transformed into more detailed supervisory control procedures which are suitable for closed loop, on-line production control. The practical execution of such closed loop production control is demonstrated on the pilot plant.

Conclusions and future work are given in *Chapter 8*.

Chapter 2

Literature Review

Recent research and development of the various control activities required for batch operation, including production planning, scheduling and control, has proceeded largely independently. However, there is a growing awareness of the need to integrate such control and operations support activities in order to gain the full benefit of advanced techniques. This chapter reviews briefly the state of the art in these production control activities. Section 2.1 begins with a general discussion of some of the relevant characteristics of batch processes. A review of batch process operations, including production planning and scheduling, supervisory control, sequential control, and model based nonlinear control techniques, is given in section 2.2. Section 2.3 briefly reviews the integration of these control activities. The need for an operations support system is discussed in section 2.4. Finally, in section 2.5, some conclusions are drawn for the field of batch process operations, and areas for further research are outlined.

2.1 Characteristics of Batch Processes

Batch processing is widely used for producing low volume, wide variety and high value-added products in the food, chemical, fine chemical, and pharmaceutical industries. The great flexibility offered by the batch mode of operation is one of the main contributing factors to its widespread use. Characteristics of batch processes have been discussed in a number of papers (Cott, 1989; Rippin, 1989; Shah, 1992; Sawyer, 1993) and are summarised as follows:

- batch processing environments are characterised by enormous *diversity* which is determined by a number of factors, *e.g.* variability of product demand, frequency of product changes, scale of production, value of raw materials, *etc.*;
- the batch mode of operation offers great *flexibility*. Batch plants are typically easier to adapt in order to accommodate new product demand or to meet specific customer orders;
- strong interaction between continuous and discrete aspects are present. Thus batch plants are intrinsically "*hybrid*";
- process conditions (*e.g.* batch sizes, raw materials, desired setpoints) may be changed frequently and operating conditions often vary over a wide range;
- process variables are almost entirely in a transient state, *i.e.* the processes are intrinsically dynamic;
- there is a strong requirement for consistent and high quality products;
- the need for operator intervention in the operation of batch plants is much higher than in the operation of continuous plants.

Many batch plants are designed or retrofitted to be *multi-purpose*. In such plants, production of multiple products may take place simultaneously following different paths. Individual items of equipment may be used for more than one operation and the same operation may be performed in different items of equipment. Cleaning-in-place is often required when a unit is used for carrying out different processing tasks, or even between repeated executions of the same tasks on successive batches. Resources (*e.g.* cooling water, manpower, steam, electricity, auxiliary equipment, *etc.*) are often shared by various tasks, and limitations in such resources may add additional constraints to the operation.

2.2 Batch Process Operations

Considering the characteristics of batch processes outlined above, it is not surprising that the operation of such processes is intrinsically complex from a planning and control perspective (Rippin, 1989; Cott and Macchietto, 1989; Bassett *et al.*, 1994). The diversity and complexity of batch processes means that various techniques and support tools are needed to tackle a wide variety of problems by considering operations planning, scheduling, and various operation control functions, defined as follows:

Production planning and scheduling. In a multipurpose plant, several products may be produced at once, and a given product might follow different routes through the plant at

different times. Products in required quantities may need to be produced utilising limited resources by given due dates. Production planning and scheduling decide how all resources (equipment, manpower, raw materials, *etc.*) are to be allocated over time so as to meet all of these constraints at maximum profit.

Supervisory control and management. Fulfilling a production plan and schedule requires the execution of tasks in a desired order. These tasks must be carried out in a coordinated manner, in time as well as quantity (Egli and Rippin, 1986). Supervisory control is one of the main control activities in batch process operation. In addition to initiating any tasks according to the production schedule, the supervisory control system is also responsible for monitoring the execution of the schedule and for ensuring the feasibility and safety of the operations in the face of any disturbances. For example, the supervisory control system detects the availability of processing units, raw material and utilities, equipment breakage, *etc.*, and is able to adapt the current operation schedules in response to such changes. Some "disturbances" may well be desired changes in production demands, new orders or changed priorities for existing orders, *etc.*

Sequence control. Sequence control, along with elementary control (continuous and discrete), is a basic control function in batch processes. It directs elementary control actions to take place in an ordered sequence to perform process-oriented tasks. Sequence control plays a key role in batch process automation in ensuring efficient and safe plant operation, and consistent product quality. This function is typically implemented with programmable logic controllers (PLCs) or sequence controllers in distributed control systems.

Elementary control (continuous and discrete). Ultimately, all control functions utilise individual continuous control loops (*e.g.* for temperature control) and basic discrete control actions (*e.g.* open/close a valve). The need for good control of individual batch operations to ensure good and consistent product quality, low operating cost, and safe operation, is well understood in the literature (Juba and Hamer, 1986). Major challenges for the control of individual batch operations are posed by the intrinsic characteristics of batch processes (Rippin, 1989). Operation in the transient regime is the norm, thus it is typically not possible to operate at "steady-state". This demands good dynamic response from the controllers over the entire operating range of the controlled variables. For highly non-linear systems, conventional PID type controllers perform poorly and have to be retuned frequently to accommodate various changes. In addition, changes in processing conditions may not always be predictable (*e.g.* batch size and batch orders are typically determined by the scheduling and supervisory control levels), and also the retuning of controllers may be very difficult. Furthermore, batch processes

are subject to frequent changes in product formulations (short product life cycle), requiring altogether new control configurations (continuous and sequential).

This section briefly reviews the current state-of-the-art in each of the areas of batch process operation outlined above. The review does not address the interaction between the design and operation of batch plants. Recent reviews of batch plant design are given by Shah (1992) and Barbosa-Povoa (1994). Furthermore, an extensive survey of all possible techniques and methods for each function is beyond the intended scope of this work, and the focus is simply on introducing the techniques which will be utilised later on.

2.2.1 Production Planning and Scheduling

Production planning and scheduling involve three key decisions: the *assignment* of resources, the *sequencing* of activities and the determination of the *timing* of the utilisation of resources by those activities (Reklaitis, 1992). Reviews of academic research in batch process planning and scheduling can be found in Reklaitis (1992), Shah (1992), Rippin (1993), and Pantelides (1993).

Solving scheduling problems involve selecting both discrete variables (*e.g.* equipment/task allocations) and continuous variables (*e.g.* batch sizes) so as to satisfy a set of constraints (*e.g.* mass balances, resources availability, logical conditions, *etc.*) and optimize a given performance criterion. The general scheduling problem is defined by Shah (1992) as:

Given:

- product recipes, in terms of networks of material states and transformation tasks;
- details of available equipment (capacities and suitability);
- details of intermediate storage (locations and capacities);
- utility utilisation profiles during the tasks;
- utility availability profiles over the time horizon of interest;
- value of products and cost of raw materials and utilities;
- production requirements;
- a time horizon of interest;

Determine:

- the allocation of tasks to equipment;
- the timing of tasks;

- the batch sizes for each tasks;
- the storage profiles for the different states of material;

so as

to optimize a given objective function.

A State Task Network (STN, Kondili, 1988; Kondili *et al.*, 1993) representation is used to describe general batch processing networks. In Kondili's approach, the optimization of discrete variables (*i.e.* sequencing and timing of tasks) and continuous variables (*i.e.* batch sizes, utility profiles, storage profiles) are performed simultaneously, in a formulation based on the discretization of time. The problem is solved as a mixed integer linear programming (MILP) problem. In general, it is assumed that there is complete freedom of transfer of material through the plant, and all the transfers are assumed to be instantaneous. A general batch scheduling software system (gBSS) developed by Shah (1992) based on Kondili's approach is used in this work.

The STN only describes process recipes. Processing equipment is simply characterized by its suitability for performing various STN tasks. Additional information regarding the equipment and their connections, and interactions between the process and equipment must also be included. An extension of the basic STN representation defined as the maximal State Task Network (mSTN) was proposed by Crooks (1992) which combines information about the process recipe with plant structure. The mSTN is generated automatically from the process STNs, the plant flowsheet and the mapping between production tasks and equipment defined by the equipment suitabilities. Crooks further defined the Unit State Task Network (USTN) to represent the state of a processing unit (*e.g.* clean or dirty). By appropriate definition of the USTN, any desired sequence of operations may be forced on a unit. Such extensions of the STN permit a batch plant/process to be modelled in a more realistic way, at the expense however of a larger MILP problem. The approach is utilised and further extended by Barbosa-Povoa (1994) to address detailed design and retrofit of multipurpose batch plants.

The basic formulation proposed by Kondili *et al.* (1993) and its extensions are based on the uniform discretization of time. In order to accurately model a process, the greatest common divisor of all task times is used as the width of the discretization interval. Rounding of the problem data must invariably be done in order to reduce the size of the formulation. Zentner and Reklaitis (1992) describe a resource constrained batch process scheduling based on constrained satisfaction techniques and heuristic. The approach firstly involves a preliminary

time interval analysis based on availability of resources. The scheduling problem is then formulated and solved as a mathematical programming problem based on a non-uniform discretization of time derived from the real process events. It has been shown that the proposed approach can reduce the size of some of the scheduling problems considerably. However, it is assumed that batch sizes are pre-fixed, and the user must also specify the number of batches to be produced in order to describe the overall production requirements. Furthermore, a pre-analysis of the time interval for each processing step is required and additional sequencing variables and constraints are introduced in their formulation.

A more general MINLP formulation based on non-uniform discretization was also presented by Zhang and Sargent (1994). Again, it appears that, at present, computational difficulties limit the applicability of this model.

Xia and Macchietto (1994) used a minimax algebra model with a simulated annealing algorithm to find the optimal routing, scheduling and product mix for discrete event dynamic systems. It has been found that in many cases this approach leads to faster solution than a mathematical programming approach for large scale problems. However, the selection of an initial sequence of "routes" and of the algorithm parameters is not always straightforward. Furthermore, the approach cannot presently consider utility resources associated with the process and optimality is not guaranteed. The problem size can become large if routing, scheduling and product mix optimization are considered together.

Despite the considerable progress made in improving solution methods and problem formulations, a major obstacle is still posed by the computational complexity of the MILP models, in particular for realistically large and detailed models, as pointed out by Pantelides (1993). Therefore, the optimal production planning and scheduling is considered only as an off-line operation support tool in this study. However, it should be pointed out that a growing number of industrial scheduling problems are within the scope of this modelling approach. Continuing advances in computer hardware and mathematical programming algorithms are expected to further enhance its applicability.

2.2.2 Supervisory Control

The primary objective of supervisory control is to coordinate the activities of the upper level production planning and scheduling systems, and the low level process control system. This function must also supervise the execution of tasks carried out by the lower level controllers,

and accommodate any variations, while enforcing feasible and safe operation. Cott (1989) developed a supervisory batch management system, called SUPERBATCH, which performs both off-line scheduling and on-line supervisory control functions in a very general way. The system consists of:

- a problem definition language for modelling plant equipment, recipes, and control networks and constraints;
- a continuous time formulation algorithm for determining the start/completion time of each individual activity as well as for the detailed timing of a production schedule subject to all operational constraints;
- a batch management system that initiates and monitors the execution of a production schedule according to the current schedule, while satisfying all the resource and operational constraints;
- a real-time data handling system that provides an interface with control systems or with other application programs (e.g. a simulator), and passes data to and gathers process information from the control system;
- an on-line scheduling modification algorithm which attempts to minimize any delay in the expected schedule completion time while ensuring feasible operation in the face of various disturbances.

In order to operate in a real time environment and deal with large scale problems, the SUPERBATCH system uses a split of decisions between the dynamic scheduler and a higher decision level. It is assumed that decisions regarding batch sizes, the sequence of batches, and the allocation of batches to processing units are made (possibly in an optimal way) at the higher level. The dynamic scheduler takes this information and determines start and finishing times of all operations. It ensures detailed feasibility of the implemented schedule but does not guarantee economic optimality. Other key features of SUPERBATCH are summarised as follows:

- the *autonomy* of the supervisory control activities is ensured (i.e. it can operate the plant independently without an upper level planning and scheduling system). Various production planning and scheduling technologies may be used, if available, at the higher level (e.g. mathematical programming, expert systems, interactive, etc.);
- the *feasibility* of any production schedule received from the upper level planning and scheduling systems is checked before it is used for on-line operation;
- *batch identity* is preserved by the use of "master procedures". Merging/splitting from/into several batches is allowed at the master procedure boundaries;

- *on-line modification* of schedules is performed on a minute-by-minute basis to respond to uncertain variations (*e.g.* processing time, batch size, equipment break downs).

Only few alternative techniques and methods have been published dealing with supervisory control of batch plants in a generic way. Djavdan (1992) has proposed an on-line scheduling technique for a combined batch/continuous plant using dynamic simulation. The simulation model is used to determine the minimum necessary volume of the buffer vessels for smooth operation of the continuous units, and to test on-line scheduling strategies. The simulation is also used to design, test and improve the schedule. The on-line scheduler is directly interfaced to a sequence control system. No detailed on-line scheduling algorithms or techniques are given in the paper. A so called process-variability-based on-line scheduling system (PVSS) has been proposed by Ishii and Muraki (1993) which modifies a schedule based on the current process state and possible future processing time variations at short intervals, and takes into account the maximum waiting time for each task. The performance of the PVSS was tested on a simple multiproduct process model by using discrete-event simulation. The schedule modification algorithm presented can only deal with processing time variations by simply shifting the start/completion times of each task to be executed, and is incapable of dealing with other variations, such as demands, resource/unit availability, resource reassignments, *etc.*

The design of a model-based reactive scheduler was proposed by Terpstra *et al.* (1994a) for mixed batch and continuous plants. The scheduler consists of three modules: a planner, an integer scheduler and a non-integer scheduler. The planner first generates a plan of how to manufacture the product(s) which contains all the degrees of freedom (integer and non-integer variables). It was not clearly described in the paper. Secondly, the integer choices within the degrees of freedom are determined by the integer scheduler (IS). Each integer choice basically represents one path for each batch through the plant, and the sequencing between these batches. The non-integer scheduler (NIS) calculates quantitative values of the non-integer variables (*e.g.* start/stop times of phases and magnitudes of flows through the interconnecting units) for a given integer schedule by transforming the schedule into and solving a non-linear programming (NLP) problem. The feasibility and optimality of each integer schedule made by the IS has to be evaluated by the NIS. This introduces additional complexity and difficulty for obtaining a useful schedule which can be used on-line. The scheduling algorithms used are not clearly described. The robustness of the scheduler to handle some disturbances (*e.g.* variation of processing times) was also discussed (Terpstra *et al.*, 1994b). A prototype of the scheduler has only been partially implemented (*e.g.* the IS and NIS modules are not yet running in reactive mode, and the planner has not yet been implemented) in the real-time expert system shell G2.

Knowledge-based scheduling has also been proposed to provide a solution for the production scheduling problem. A dynamic scheduling package (DSP, Moore, 1993) is developed and implemented in the G2 real time control system using knowledge and expert system scheduling rules. A prototype interactive scheduling system (GanttKit, Halasz *et al.*, 1992) was also developed as part of the BatchKit (Hofmeister *et al.*, 1989) knowledge based system for the planning and operation of batch processes. Scheduling is formulated as a constraint satisfaction problem. A heuristic optimization procedure was used to generate feasible schedules. It is essentially left to users to determine a "best" schedule that matches a given set of product demands.

The main advantages of the knowledge-based approaches are that new information derived from manufacturing experience can be easily incorporated into rules which may be used for determining the "best" schedule. However, such approaches can be successful only if large amounts of specific knowledge are available and lack any guarantee on the performance of the resulting schedules. The effectiveness of decision-making depends typically on the validity of the supplied rules.

In conclusion, at the supervisory control level, the operation feasibility and computational efficiency become very important and must be ensured at all times for on-line operation. Meanwhile, the supervisory controller must also be able to adapt the schedule according to the status of the plant and control in response to any disturbances. The SUPERBATCH system (1994) provides a very generic solution to the supervisory control of batch processes and will be used in this study.

2.2.3 Sequence Control

Sequence control is one of the fundamental control elements in batch process automation. However, the development of control sequences is a highly labour intensive and error-prone task, and requires personnel with mixed expertise and experience (Love, 1991). Several major problems are:

- an incomplete knowledge and understanding of the process may lead to ill-defined, incomplete and inconsistent functional specifications;
- the development of detailed control code from the functional specification often leads to errors. Various engineering and control departments may be involved in this task;

- too many degrees of freedoms are typically left to the control engineers as a result of incomplete functional specifications. This may cause inconsistency of the code with respect to the specifications and also errors.

A knowledge-based approach was proposed by Love *et al.* (1992) to the process of specifying and generating control sequences. The key idea was to provide an environment to formalize the development of functional specification so as to enforce consistency and completeness of the specifications and reduce the degrees of freedom left to the control engineers in the phase of software development. A formal specification is clearly a key step toward an automatic generation of application software. A more formal method was developed by Crooks (1992) to generate sequence specifications based on item-level equipment models. Most of the instructions associated with material transfer operations can be generated automatically. However, the remaining primitive control instructions may have to be supplied by the user for the synthesis of each sequence specification.

Formal methods based on Discrete Event Systems (DESs) theory have also been proposed by many researchers for synthesis and verification of procedural controllers. Recent reviews of the DESs modelling, synthesis and control are given by Sanchez (1994) and Yamalidou *et al.* (1990). Temporal Logic and the Model Checking Verification method (Clarke *et al.*, 1986) were used by Moon *et al.* (1992) for automatic verification of sequential control systems. Moon and Macchietto (1994) applied them to check logical inconsistencies in batch processing control procedures. Synthesis and design of procedural controllers based on the DESs theory was also investigated by Sanchez (1994). Given a model of the primitive plant items and a logical formulation of the desired behavioural specifications, Sanchez's algorithm generates a supervisory controller as a labelled finite state machine which is guaranteed to restrict the process within controllable limits.

The use of DES modelling tools permits the use of different methods of analysis. However, many problems, both theoretical and practical, still remain, such as that of state "explosion". As the number of plant items increases, the number of variables in the process model increases. This results in an exponential increase of the total number of states. Another issue is the handling of continuous variables. Discretization of continuous variables (such as level and temperature measurements) not only leads to an increase in the total number of variables, but may also result in poor or simplified control procedures. *Ad hoc* manners of dealing with time-delay (*e.g.* grace time of on/off control valves and pumps) may also result in application difficulties. In particular, the resulting procedural controllers are only guaranteed to be correct subject to the given specifications and the input models. Incorrect

models or incomplete specifications could lead to incorrect procedures. Furthermore, quantitative effects of the procedures on the behaviour of the system may be difficult to assess.

Therefore, checking of control sequences (developed either manually or automatically) prior to commissioning is typically essential. Dynamic simulation techniques can be used for this purpose. A number of modelling and simulation packages (*e.g.* gPROMS) are available which support the modelling of not only the physico-chemical behaviour of the processes, but also the discrete and logical control actions (*e.g.* in terms of control sequences). Dynamic performance and response of the system under the given control sequences can then be tested and the potential for further improvements may be identified by simulation. Violations of operating constraints or variable bounds could be checked to ensure that the procedures safely perform the intended operations. Dynamic simulation is therefore considered here as an engineer-driven method for developing and validating control sequences.

2.2.4 Model Based Control of Batch Processes

Recent surveys on the status of process control indicate that PID control still predominates and is sufficient for most control needs (Yamamoto and Hashimoto, 1991; Schuler *et al.*, 1991; Richalet, 1993). Eighty to ninety percent of control problems are currently solved by PID controllers. However, it is also well known that PID controllers perform poorly or must be detuned for systems which are subject to constraints and severe nonlinearity, and for systems that are operated over a wide range of operating conditions. In particular, the control of batch processes poses challenging problems due to the intrinsic characteristics of the processes (*e.g.* wide range of operating conditions) and the uncertain production environments (*e.g.* variations of batch sizes, processing conditions, high and consistent quality requirements, *etc.*). The increasing interest from academic and industrial researchers in model based control techniques is well reflected in Richalet's paper (1994) which indicates that the market for Model Predictive Control (MPC) alone was \$80 million in 1993, and is expected to increase fourfold by the year 2000. A renewed emphasis on product quality improvement (in particular, product consistency), operation safety, and cost reduction were cited as contributing factors.

This section reviews two typical classes of model based control techniques: *Model Predictive Control* (MPC) and *Geometric* nonlinear control methods, with emphasis on the applications to nonlinear processes, and in particular, batch processes. MPC techniques are established on solid theoretical foundations and are increasingly found in industrial

applications. Geometric methods offer an effective tool for the analysis and design of nonlinear systems and have yielded a fundamental insight into the nature of nonlinear systems. However, some application difficulties still remain. Many other control techniques can be viewed as a special case or extension of these two techniques, as will be discussed later in detail. A review of nonlinear control of chemical processes is given by Bequette (1991).

2.2.4.1 Model Predictive Control

Model Predictive Control (MPC), also known as *receding horizon control*, has achieved a significant level of acceptability and industrial success in practical process control applications in the last 10 years (Fisher, 1991; Richalet, 1993). Several consulting and large engineering companies now offer MPC software packages and application assistance, such as IDCOM-HIECON (Richalet, 1993), Dynamic Matrix Control (DMC, Cutler and Ramaker, 1979), and Setpoint (Grosdidier, 1987). A comprehensive survey of the history, theory and practice of MPC is given by Garcia *et al.* (1989). Ricker (1991) provides a good technical overview of the techniques. A recent survey and tutorial of MPC techniques for both linear and nonlinear systems is provided by Rawlings *et al.* (1994).

In MPC, algorithms and linear or nonlinear dynamic models are used to predict the effect of future control actions of the manipulated variables on the output. The future moves of the manipulated variables are determined by optimization with the objective of minimizing the predicted error subject to operating constraints. The optimization is repeated at each sampling time based on updated measurements from the plant. A key feature of the MPC is that process constraints can be incorporated directly into the on-line optimization performed at each step (Oliveira and Morari, 1994). This feature contributes heavily to the success of MPC techniques and lead to a number of applications. Richalet (1993) and Grosdidier (1987) summarise some of the applications, mostly in the petrochemical industry. A number of design techniques, such as DMC, Model Algorithmic Control (Rouhani and Mehra, 1972), and Internal Model Control (IMC, Garcia and Morari, 1982) are considered to be emanated from the MPC method (Garcia *et al.*, 1989).

Although MPC can be used for control of both linear and nonlinear systems and the approaches above have been applied to a number of industrial plants, most of the reported applications are on continuous processes for which models can be linearized locally. Very few industrial applications have been reported for batch processes, since MPC algorithms which use nonlinear ODE models for prediction of future outputs lead to a non-convex optimal

control problem. Thus, a reliable implementation of the controller, which requires the on-line solution of the nonlinear program, is very difficult (Oliveira and Morari, 1994; Rawlings *et al.*, 1994).

2.2.4.2 Differential Geometric Approaches

In many plants, especially batch plants, processes are operated under a wide range of operating conditions. For high nonlinear systems, it is well recognised that conventional PID controllers or controllers based on locally linearized models perform poorly and must be detuned to ensure system stability. The differential geometric approaches offer some promise for such systems based on a model which captures the structural nonlinear characteristics of the process. Unlike Jacobian linearization, which linearizes the model only at the normal operating point, these techniques employ nonlinear state transformations and nonlinear feedback transformations that transform a nonlinear model into a linear input/output map (input-output linearization). Then, linear controller design techniques can be employed for the transformed linear system. Intrinsic characteristics of the nonlinear system, such as relative degree of the system and zero dynamics, can be determined using this method. An overview of the methods is given by Kantor (1986). A detailed mathematical perspective of differential geometry and applications to controller design are given by Isidori (1989), and Slotine and Li (1991).

Several nonlinear controller design approaches based on the differential geometric theory have been reported. Globally Linearizing Control (GLC) was proposed by Kravaris and Chung (1987) for nonlinear systems in *control linear* or *control affine* form. The approach is based on the use of static state feedback in an inner loop to transform a nonlinear system into an equivalent linear one, then on the use of an external linear controller with integral action for control of the transformed linear system. An experimental application of the GLC method to a batch polymerisation reactor was studied by Soroush and Kravaris (1992). Their results showed that the GLC controller is very easy to derive and tune, superior in performance to a PID controller, and computationally efficient. Nakamoto and Watanabe (1991) used GLC to control level, temperature and pH in a tank.

Henson and Seborg (1991) extended the GLC method to *general* nonlinear processes and presented a simulation study of the approach to a continuous stirred tank reactor system. Application of geometric nonlinear control to a CO₂ absorption/desorption pilot plant was also carried out by Dore (1993). He further extended the basic geometric control technique to

include algebraic equations. Kravaris and Palanki (1988) analyzed the robustness of the GLC strategy using a Lyapunov-based approach. Given upper bounds on the modelling error, they designed a nonlinear state feedback law that guarantees stability and performance for all perturbations with the given bounds.

Several other controller design approaches were also proposed for nonlinear processes. Generic Model Control (GMC, Lee and Sullivan, 1988) and Reference System Synthesis (RSS, Bartusiak *et al.*, 1989) are two similar approaches which may be viewed as a simplified GLC approach for processes with relative order 1 (Henson and Seborg, 1991; Barolo, 1994). No experimental confirmation of their equivalence is known to have been reported. The fundamental idea of these two approaches is that the desired behaviour of the closed-loop system is represented by a reference trajectory which is known to represent "good" operation. The controller is then designed in such a way that closed-loop response of the system is driven as close to the reference conditions as possible. The design procedure is straightforward and the nonlinear process model is used directly in the controller design. A number of applications have been reported. Balchen *et al.* (1988) applied RSS to a nonisothermic CSTR with first order exothermic reaction. Uncertainties in the heat capacity and activation energy are considered. A complete case study of GMC of an evaporator system was presented by Lee *et al.* (1989a). GMC is compared to DMC control and shows excellent dynamic behaviour. Cott and Macchietto (1989) used GMC for temperature control of a simulated exothermic batch reactor. An exponential filter is used to estimate the heat of the reaction. An experimental study of the GMC controller for a partially simulated exothermic (PARSEX) reactor was presented by Kershenbaum and Kittisupakorn (1994). An extended Kalman filter (EKF) is used for estimation of the heat of reaction, but the authors conclude that the EKF is rather more sensitive to plan/model mismatch than would have been predicted from simulation alone. GMC control of a strong acid-base neutralisation system was reported by Costello (1992). A GMC controller was also used for the automatic start-up of a binary distillation column (Barolo *et al.*, 1992).

Lee *et al.* (1989b) presented an extension of the GMC method to include process/model mismatch compensation. The model/process mismatch f_{mis} is included in the GMC controller formulation. To implement the control law, f_{mis} must be estimated from the measured output and the approximate model at each sampling point. The capability of the approach is demonstrated by simulations of both linear and nonlinear systems with parameter mismatch and measurement noise. Further extension of the GMC approach to handle constrained nonlinear systems was proposed by Brown *et al.* (1990). Reference trajectories of both the constraint variables and the control variables are specified. Slack variables defining the

trajectory departure from the chosen specification curves are added to the GMC control law for both the control variables and the constraint variables. The solution of the problem becomes a nonlinear constrained optimisation, which minimizes a function of the slack variables. The approach was applied to two simulated systems without process/model mismatch.

In spite of the significant progress made in developing control techniques for nonlinear processes, there are many open problems yet to be solved which restrict their application. In general, most nonlinear control techniques require a good model of the entire system, and can only be applied to a class of nonlinear systems which satisfy certain conditions (*e.g.* to systems of low relative order, minimum phase, satisfying disturbance matching conditions, *etc.*).

2.2.4.3 Applications to Batch Processes

A review of unit operations that have been controlled using nonlinear control techniques was presented by Bequette (1991). A summary of CSTR applications of the exact linearization technique was given by Henson and Seborg (1991). A recent review of applications of the geometric nonlinear control to a number of industrial processes was provided by Dore (1993). Most of the applications reported are to continuous processes. Only a small proportion of the applications address batch processes. A summary of some applications to batch processes is given in Table 2.1.

In general, most of the applications have shown that model based controllers perform better than the conventional PID controllers for control of nonlinear batch processes as they often utilise more information in the resulting control laws. The geometric nonlinear control techniques have shown potential for batch processes which are difficult to be linearized locally. However, a good process model of the entire process is typically required which is often difficult to obtain. In particular, there is a need for a general approach to controller design for systems which are only partially known. Furthermore, there also remain a need for detailed experimental studies to evaluate the applicability of these techniques to batch processes.

Reference	Method	Application
Bhat, <i>et al.</i> (1990)	RSS	batch reactor,
Cott and Macchietto (1989)	GMC	batch reactor, simulation, exponential filter
Feyo De Azevedo, <i>et al.</i> (1989)	GPC	batch evaporative crystallizer, simulation
Foss, <i>et al.</i> (1994)	MPC	batch reactor, local model, simulation
Garcia (1984)	DMC	batch reactor
Kershenbaum and Kittisupakorn (1994)	GMC	batch reactor, simulation and experiment, EKF
Kravaris, <i>et al.</i> (1989)	GLC	batch reactor, open loop observer, simulation.
Le Lann, <i>et al.</i> (1992)	Adaptive MPCMR	pilot and industrial scale batch reactors, experiments
Marroquin and Luyben (1972)	Variable Gain	batch reactor, experiment
Peterson, <i>et al.</i> (1992)	DMC	semi-batch reactor
Rawlings, <i>et al.</i> (1989)	MPC	semi-batch reactor, end-point control
Soroush and Kravaris (1992)	GLC	batch reactor, experiment

Table 2.1 Summary of applications of nonlinear control techniques to batch processes

2.3 Integrated Operation of Batch Processes

As already discussed in section 2.1, the flexibility and diversity of batch processes also leads to increased complexity of operation. Despite the significant progress made in the solution of problems in batch process operation (*e.g.* planning and scheduling, control, *etc.*), each of these aspects are typically considered in isolation. It is however generally recognized that a direct coupling exists between various level of control activities. For example, poor low-level control might result in the loss of a batch which can have a significant impact on the supervisory control by calling for the revision of previous schedules. Thus, an integration of various control activities is typically required for the proper operation of batch processes.

Integration of the various functions for operation of batch processes has been discussed by a number of authors (*e.g.* Rosenof and Gosh, 1987; Mather and Garner, 1989; Pekny *et al.*, 1991; Macchietto *et al.*, 1992). Meanwhile, industrial batch process standards, such as these proposed by the NAMUR and ISA SP88 committees, are in the process of development. The

standards provide some of the essential specifications needed for integration, in particular with respect to the lower level production control functions. The SUPERBATCH system (Cott, 1989) provides a general framework to integrate various production planning systems and a general control interface to link with low level control systems for on-line plant control and supervision. The system conforms broadly with the functions defined in the proposed ISA SP88 standard (1994).

A computer aided procedure synthesis (CAPS) system was developed by Crooks (1992) which provides as a byproduct a rather generic way of integrating various batch processing functions. Given a description of the plant and processes, operating goals and constraints, the CAPS system identifies the control phases, their organisation into networks to form "master procedures", and the detailed specifications for each control phase required to achieve one or more operating goal(s). The solution approach follows three steps. First, a discrete-time, mixed-integer linear program (MILP) approach is used to address a detailed tasks scheduling problem, using the STN, mSTN and USTN representations. A modified version of the gBSS software of Shah (1992) is used for this purpose. Secondly, control phases and master procedures are identified from the detailed tasks schedule (subgoal output). Finally, the third step of CAPS utilises much more detailed information about the plant network (P&ID) and a set of rules to generate rather more detailed control sequence specifications. The output produced by CAPS (control phase definitions, master procedure definition, and optimal schedule to achieve the required goals) is then converted into a format that can be used directly to define the input models for the SUPERBATCH system. This provides the possibility of a downward path from the generation of an off-line optimal production schedule to the on-line execution of the corresponding production schedule.

Chua (1995) extended this idea to include some re-scheduling functions at the production planning level to handle major production variations (*e.g.* new production requirements, breakdown of equipment, *etc.*). Given a plant and process model, and a production goal, the CAPS system is used to generate an initial optimal schedule and all required supervisory procedures/phases. A one-to-one mapping between the tasks in the optimal schedule and control phases in the supervisory control level is established (except for instantaneous transfers) so that the status of the plant/control phases can be mapped back to the optimal scheduling level, and provide new initial conditions for updating the schedule. Again, the STN representation and a MILP formulation were used to generate an initial optimal production schedule, and to perform the re-scheduling function by taking into account the current plan status and any new production requirements. Based on the above ideas, the system resulting

from integration of the optimal planning and the supervisory control functions is broadly summarised in Figure 2.1.

In general, the work discussed above concentrated mainly on integrating the optimal planning and supervisory control activities, assuming that low level controllers (*e.g.* control sequences) are already available. In order to generate meaningful supervisory control phases and to map back the control phases to the tasks at the optimal scheduling level, the process and plant must be modelled in sufficient detail at the optimal scheduling level. This typically leads to rather large problem sizes and solution times at the schedule optimization level.

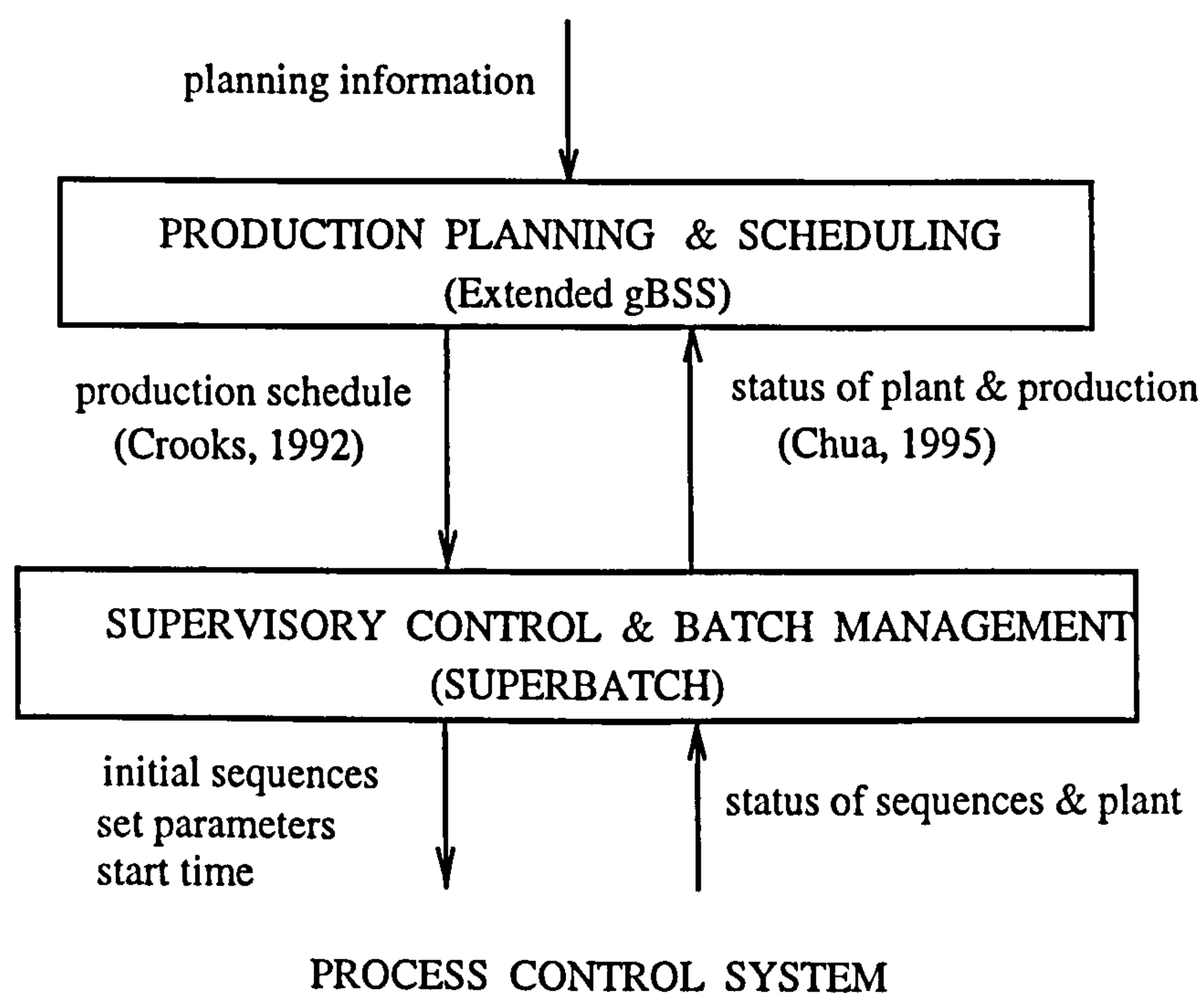


Figure 2.1 Integrated management and control system

A model-based integration of process operations is also discussed by Bassett *et al.* (1994). A decomposition strategy is used to solve the planning and scheduling problems. The planning problem is posed as a MILP, based on the aggregation of the scheduling constraints over time, thus resulting in a more concise, but less detailed description of the problem. The planning problem solution serves to define the production targets and horizons for a series of scheduling problems. However, due to the constraints aggregation in the planning problem, scheduling problems may be found to be infeasible under the decomposition. A multi-layer representation of the scheduling process (graphical representation, scheduling language, mathematical programming language, and object oriented computer language) is considered within the RCSP++ scheduling system (Zentner *et al.*, 1994). A large scale model predictive control is

also incorporated into the integrated system hierarchy for calculating optimal values of key process variables. However, the resulting problem size is typically too large for programming algorithms to solve. Results have been presented only in a conceptual way on academic problems.

The requirements for an integrated batch system environment and integration of design, operations scheduling and control are discussed in more details by Macchietto *et al.* (1992) and Macchietto (1993). They concluded that different representation are clearly best for addressing different problems due to the complexity and diversity of batch processing activities, and that the consistency between representations and data aggregation and disaggregation needs to be given particular attention so as to achieve a proper integration.

2.4 Operations Support Systems

The need for further research towards an "intelligent" operation support system, which would ease present concerns including quality assurance, safety, fault detection, and personnel reduction, was highlighted by Yamamoto and Hashimoto (1991). As it is typically difficult to account for all operation aspects due to the diverse nature of the processes, operations support tools are often needed to provide feedback in the early stages of development. Moreover, validation and testing of each level of control in an integrated system on an actual plant is not only expensive, but also unsafe. A system that supports the development of manufacturing control procedures and control algorithms, and enables various testing and validations of control activities to be carried out in an off-line fashion is obviously beneficial. Such a system may also provide continuous operations support throughout the plant lifecycle, with the aim of enabling continuous improvement in the efficiency and safety of operations.

Few reports have been presented in the literature in this area of research. An operation support system using dynamic simulation techniques for a combined batch/continuous plant is presented by Deguchi *et al.* (1994). The plant consists of several parallel batch reactors, intermediate storage tanks, and continuous separation units. The support system is composed of three blocks, including a *plant model* block which models the sequence of batch reactor operation and material balance; a *decision making* block that is used to determine the outlet flow rates of the intermediate tanks to ensure a smooth operation of the continuous units and a *man-machine interface* block. An Operation policy is investigated using the support system

and applied to the actual plant. Their results show that the support system based on dynamic simulation is an effective tool to improve plant operation under complex constraints.

Singh and Mahler (1994) described the integration of a decision support system with *legacy* systems (they refer to existing proprietary systems, interfaces and databases as legacy systems). The support system is used for production scheduling. It essentially uses a heuristic approach based on the classification of the knowledge, and the algorithms used for decision making are not given. Modelling and simulation of recipe driven multipurpose batch plants using an object oriented approach are also presented by Engell and Wollhaf (1994). A simulation technique is used to improve the support of the operators in their tactical decisions on batch sizes, assignment of equipment to batches, sequencing, starting times, *etc.*

The BATCHES simulator (Clark and Joglekar, 1992) has been described as providing a simulation based decision support environment. By modelling the equipment and recipe networks and using dynamic models to represent corresponding processing steps, the BATCHES system allows the user to specify interactively the assignment of equipment and resources to operations, and to evaluate the dynamic response of the plant through the simulation. The scheduling and planning decisions are essentially performed manually.

Simulation is often employed for various decision support environments. However, most operations support systems consider one level of the control hierarchy. There is a clear need for a support system for validating various scheduling, supervisory and low level control functions, either individually or in an integrated manner. Also the system should not only address off-line simulation but also be targeted towards on-line implementation.

2.5 Summary and Conclusions

This chapter has presented a review of recent advances in batch process operations. In section 2.1, the characteristics of batch processes were discussed. The flexibility of batch plants implies that batch operation is often very complicated. This calls for the development of various operation support tools/techniques and requires the integration of various control functions in a hierarchical structure. Section 2.2 presented a brief review of research activities in four areas of batch operations, namely, production planning and scheduling, supervisory control, sequential control and advanced model based control. There is a clear need for integration of these various control activities to achieve effective plant operation. A review of

integrated batch process operations was presented in section 2.3. Section 2.4 discussed the development of an operations support environment for improving batch process operations.

From the literature review, it is apparent that considerable attention has been paid to the development of methods and computer-aided tools which address various individual functions (*e.g.* planning, scheduling, control, simulation, *etc.*). However, much of the work on integration has concentrated mainly on the planning and scheduling levels, assuming that the low level controls (*e.g.* sequential and elementary control) are readily available. Due to the strong coupling between these control functions, the development of flexible and effective control sequences, as well as of robust controllers that are capable of operating over a wide range of conditions must also be considered in an integrated way. Furthermore, differences in the modelling techniques and decomposition of control activities may cause inconsistencies when various levels of control are integrated.

The inherent variability and complexity of batch processes increases the difficulty of operation, as well as the difficulty of developing, testing and implementing the operations support tools and techniques. An operations support system is required to provide off-line validation and testing functions for the integrated system, and to provide feedback at an early stage in the development cycle, aimed at achieving safe and easy on-line implementation. Reported work in this area is very scarce.

Finally, very few practical applications of generic systems that support both on-line control and off-line development are reported. Demonstration of such systems on actual plants (as opposed to simulators) is essential to assess the applicability of the techniques, and to evaluate the relevant benefits of using the systems.

In conclusion, there is a clear need for further research in many areas associated with batch control integration. There is also a need for a tool to support batch integration. It is obviously desirable to develop a system which could provide both on-line and off-line control and various operations support functions in an integrated way. The following production controls and operations support functions should be included:

- a high level optimal production planning and scheduling function that can handle batch process scheduling problems in a generic way;
- an intermediate level supervisory control system which coordinates the off-line optimal scheduling with a lower level control system, and supervises the production on-line in a closed loop;

- a control system that can perform and support detailed lower level control functions (*e.g.* sequential control, continuous and discrete controls, *etc.*). These low level controllers must be flexible and robust to ensure responsiveness, operational efficiency and safety;
- an off-line operations support environment to assist in the development and testing of various control functions and new manufacturing control procedures both individually and in an integrated way.

In order to address the above requirements in a generic manner, problems at all levels must be described in terms of suitable models (model-based approach). A strategy is needed to coordinate controls and to map the production control models used at the different levels, ensuring that information and local decisions can be transferred/shared among various application programs and/or control systems.

The aims of this thesis are therefore: i) to design and develop an integrated system that provides the above production control and operations support functions, and ii) to demonstrate such a system on an actual plant. The following chapters will show how these aims can be achieved.

Chapter 3

Advanced Processes Manufacturing System - Design

In this chapter, the design of an advanced processes manufacturing system is presented. Some design considerations and the overall structure of a proposed operations support system are given in section 3.1. Descriptions of a multipurpose batch pilot plant, an industrial control system and a software package for communication support are also given in section 3.1. Section 3.2 discusses some of the modelling issues, with particular emphasis on aggregation/disaggregation, coordination, and design of control sequences for flexible operation. Conclusions are given in section 3.3.

3.1 Advanced Operations Support System - Design and Implementation

Batch process engineering spans a wide range of activities (including plant design, production planning and scheduling, process control and automation, operator training, *etc.*). The diversity of these activities (in terms of time scale, as well as the nature of problems involved) and the complexity of batch plant operations mandates the use of a *hierarchical* design of the operations support system. A hierarchical structure is also favoured due to safety, maintenance, validation and flexibility considerations. Moreover, Williams (1987) lists the following reasons why a hierarchical approach has been preferred for operation of batch processes:

- it promotes the principle of *autonomy*, *i.e.* responsibility can be delegated as low in the hierarchy as possible;
- it promotes the principle of *locality*, *i.e.* distributed control can be readily applied since plant units are usually widely distributed but are relatively self-contained;

- the distribution of control functions reduces the span of control responsibility of each control computer, thus reducing its work load.

A key requirement in the development of an operations support system of general use is the ability to abstract and represent process operation problems in such a way that information can be structured, translated and validated in the form of models, so as to allow specific decisions to be made in a systematic way, through the use of *generic* support tools and techniques. Therefore, a *model based* approach is considered in the design of the support systems.

Given these general design considerations, an advanced operations support system was developed that performs:

- *on-line control implementation* by utilising actual control hardware/software for running a plant. This provides a *production environment* capable of supporting various production control functions (from process planning and scheduling to supervisory control and advanced control techniques), aimed at achieving high operations efficiency, product quality, *etc.*
- *operations support* functions by utilising a simulation of plant and/or controls. This provides a *development environment* capable of enabling the use of advanced tools and techniques for the development and testing of manufacturing control procedures aimed at achieving safe and efficient development and implementations of new control strategies.

A schematic representation of the system is shown in Figure 3.1. The box on the left includes a pilot plant (described in section 3.1.1) and its industrial control system (described in section 3.1.2). The box on the right utilises the gPROMS simulator (Barton and Pantelides, 1991), which is capable of modelling and simulating batch plants and operations. The supervisory batch control functions are carried out using the SUPERBATCH software (Cott, 1989), while optimal production planning and scheduling are performed using the gBSS software (Shah, 1992). The model based control function utilises nonlinear control techniques described in Chapter 5. The production planning and scheduling, supervisory control and model based control functions are shared by the on-line and the simulated portions of the system. The link between all applications and functions is provided by Control Link, a software package described in section 3.1.3.

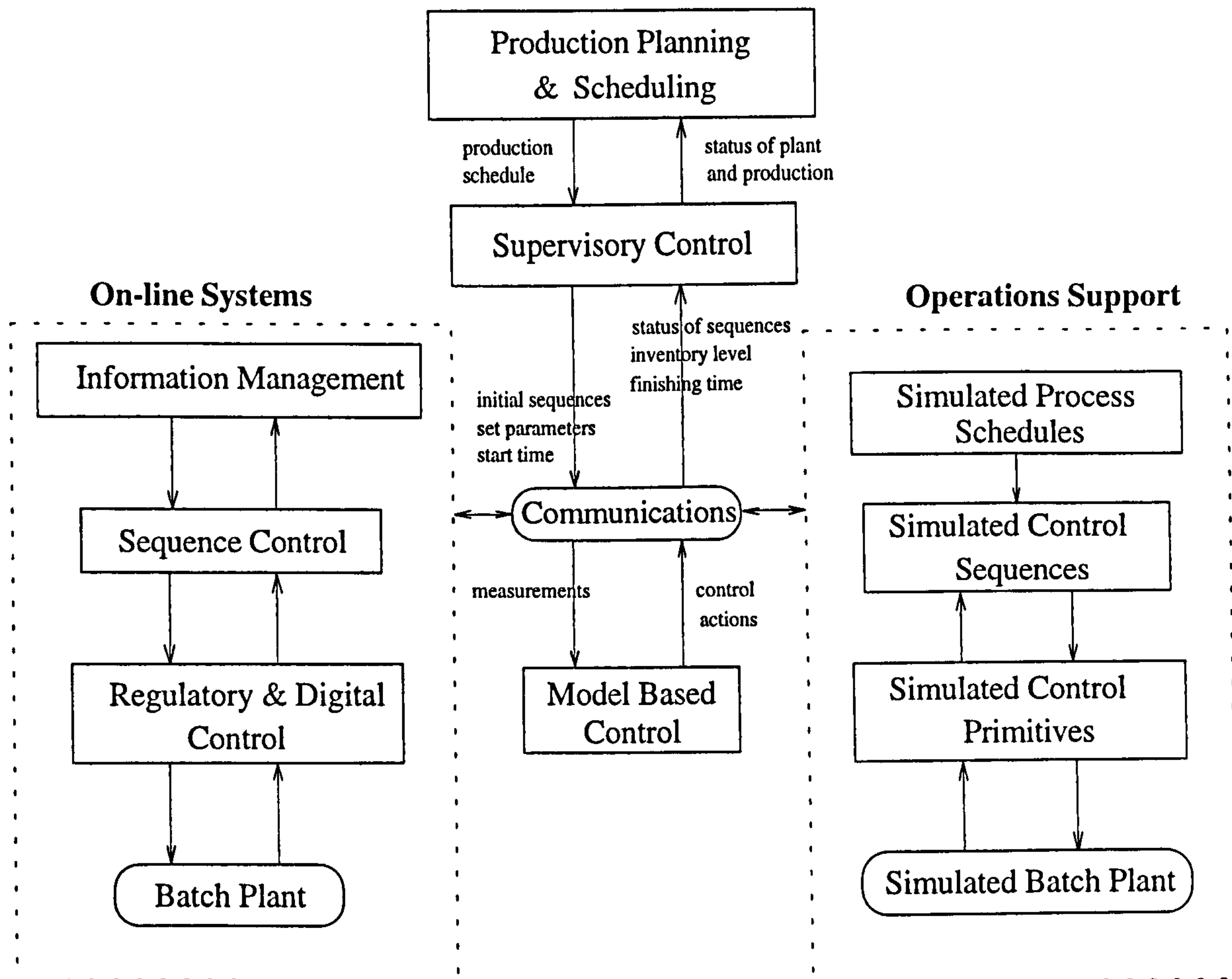


Figure 3.1 Architecture of the advanced operations support system

Within this design, each application can also be operated/tested individually or in combination with others. For example, an off-line simulation of a process can be performed individually. In this case, the order of the tasks to be executed is fixed a priori (in an open-loop manner) and programmed as a specific "schedule" in the gPROMS language. This mode of operation is useful for evaluating a control sequence, testing a model based control algorithm, debugging a dynamic model, *etc.* Then, the simulator may be connected to and driven by the supervisory controller. The control sequences (simulated) will then be executed according to the supervisory control algorithm, which will modify the schedule in a closed-loop manner. This provides a useful test bed for the supervisory level control. In this mode, it is possible to evaluate the effect of any variations on operations (built into the simulation model) and to test the effectiveness of the supervisory decisions. The simulator may be used to perform just a control task, taking data from the plant and sending the calculated setpoints back to the regulatory control portion of the hardware. This may be used to implement advanced model based control functions. Finally, the system can also be used for a variety of operator trainings. This is particularly important for batch processes due to the need for frequent operator intervention.

3.1.1 The Batch Pilot Plant

The batch pilot plant (Figure 3.2) was designed to hygienic standards as required for food production. It consists of two raw material storage vessels T1 and T2, which are used to prepare raw materials and to feed a multipurpose batch reactor T3. Both batch size and flow rates can be controlled. The reactor is installed on load cells, with which the weight of the reactor contents is measured. The reactor temperature can be controlled in several ways depending on the type of reaction (exothermic or endothermic) and control requirements. Additives (enzymes, acid or alkali) can be added to the reactor with peristaltic pumps to produce the required reaction conditions. An on-line HPLC system with automatic sampling is being installed to perform analytical functions. A viscometer is mounted on the reactor vessel to monitor the viscosity change throughout a reaction. Two secondary vessels, T4 and T5, are currently used for product storage. T5 can also be used as a decanter to separate liquid phases. All vessels have 100 L capacity. The plant is also equipped with three heat exchangers to carry out temperature control tasks, and a Cleaning-In-Place (CIP) unit for supplying cleaning fluid (a caustic solution) to five cleaning circuits.

Highly flexible plant connections between the five vessels is achieved through a pipe network and double-seat valves which permit diverting flows between the vessels. Some of the pumps feature variable speed control and bi-directional flow. The industrial plant control system (ACCOS 30, by APV Baker) manipulates 45 on/off actuators and 7 analogue control loops through 98 operating sequences. All the automatic valves and pumps have a feed back position sensor which allows these individual items to be continuously monitored through the plant hardware and control sequences (when a sequence is active). The plant may be used to carry out a variety of processes, two of which are described in section 4.2.1.

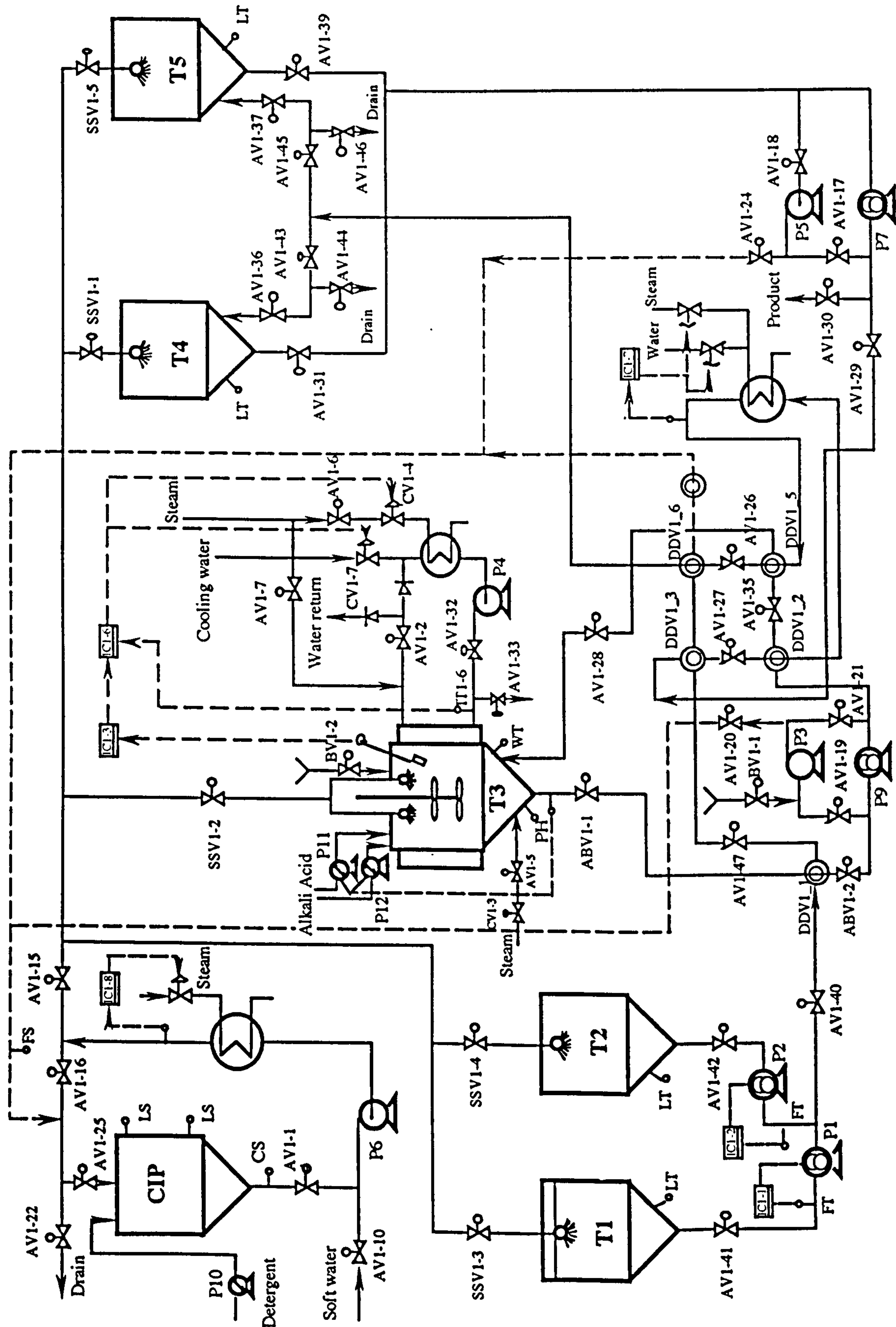


Figure 3.2 Multipurpose batch pilot plant

3.1.2 Control Hardware Configuration

An industrial control system, consisting of the ACCOS 300, ACCOS 30 and ACCOS 2S (supplied by APV Baker), is used to perform on-line control and management functions. A detailed schematic of the control system is shown in Figure 3.3. The ACCOS 300 is a management information system. It collects plant information (measurements, equipment status, *etc.*) from the ACCOS 30 control system through the proprietary *Paranet* link and stores the data in a plant "mirror". A colour graphics terminal displays active routes in the plant, values of main process measurements (*e.g.* flow rates, temperatures, viscosities, *etc.*). Two PCs (PC-1 and PC-2) are used as additional operator terminals running a Graphics Menu, a maintenance program and other applications. ACCOS 30 is a general process control system based on the Motorola 68010 microprocessor and VME bus. It operates as a stand alone system or can be networked with other systems through a *Network controller* and the *Paranet*. Control sequences are written in PARACODE, a proprietary control language, and run on sequence controllers. Individual sequences may be initiated by entering an Operator Entry Code (OEC) on the engineering console, through the graphics menu on PC-1, or directly by the supervisory control system in which process scheduling and on-line monitoring functions are performed.

A number of application programs can run on a network of Sun Workstations and communicate with the ACCOS system through the Control Link software (also running in a Sun Workstation) and through an Ethernet connection. Applications include in particular optimal production planning/scheduling (gBSS), off-line scheduling and on-line supervisory control (SUPERBATCH), dynamic modelling and simulation (gPROMS) and nonlinear control.

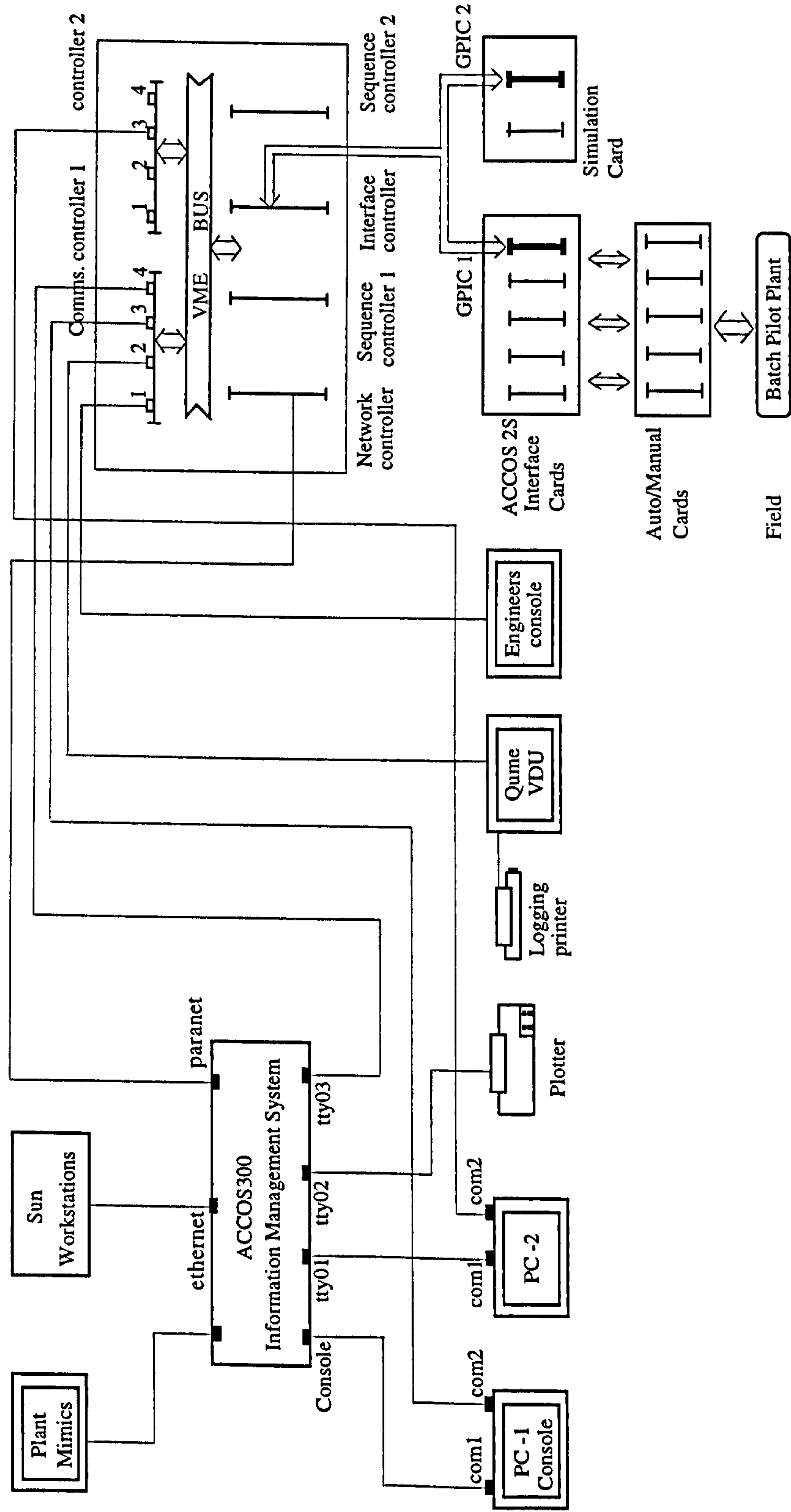


Figure 3.3 Schematic of the ACCOS information management and control system

3.1.3 Communication Design

Communication between applications (clients) and servers is provided by Control Link, a UNIX based software package developed at Imperial College. A schematic of Control Link is shown in Figure 3.4.

A generic portion provides host names and network information for the client programs (*e.g.* SUPERBATCH, gPROMS) to connect to the appropriate server. It also provides a control system tag database to map application tag names to the corresponding control system configurations (*e.g.* item numbers, item types, and channel numbers). A system specific portion of Control Link supports communication protocols to specific control system servers (an interface to the ACCOS 30 system is used here). This ensures information (data, control commands, messages, alarms, *etc.*) can be transferred asynchronously among various application programs and/or control system(s) in a multiuser/multinode environment.

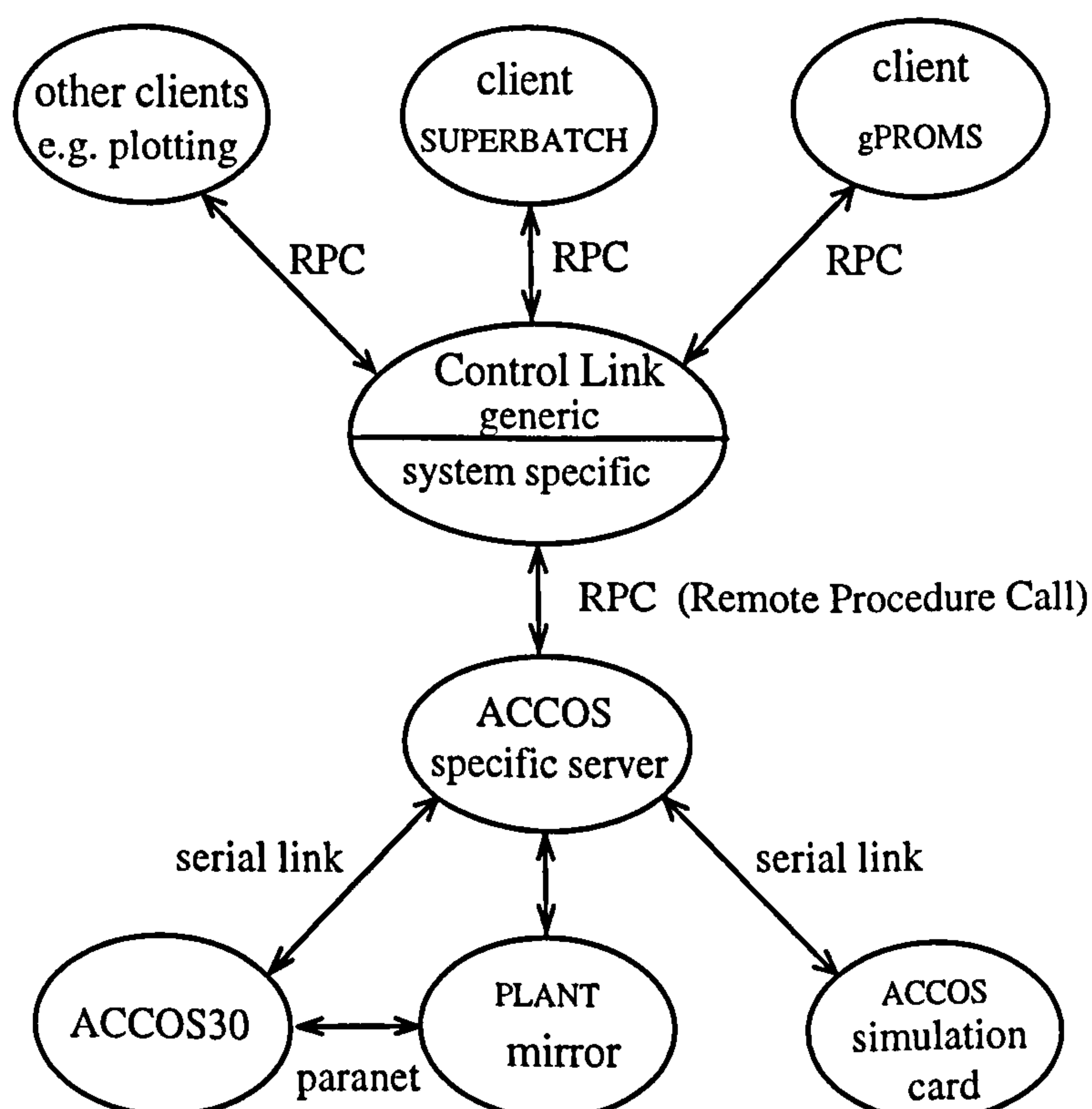


Figure 3.4 A schematic of Control Link

The connection/configuration to the simulator or the ACCOS control system is done through two files:

- *ControlLink.dat* - This provides host names and network information for the client programs (*e.g.* SUPERBATCH and gPROMS) to connect to the appropriate server;
- *CL_tag_map.dat* - This provides control system tag database which maps application tag names to the ACCOS 30 control system configuration.

To connect to the simulator, SUPERBATCH and the simulator open network connections to a Control Link server (configured for data storage) running on a Sun Workstation. Data is then passed between the two applications upon a request (*e.g.* using GET/SEND command) to the server. To connect to the plant, SUPERBATCH opens a connection to the Control Link server which is itself connected to ACCOS 300 and the plant.

3.2 Modelling of Batch Process Operations

One of the key issue is the modelling of the manufacturing system. The use of mathematical models enables the information to be represented in a formal way, and to be translated into knowledge in a form in which decisions can be made in an algorithmic manner. Due to the complexity and diversity of the problems dealt with in batch processing, a variety of model representations must be considered. However, a number of modelling issues must be addressed to ensure that decisions can be transferred between different applications and appropriately utilised.

3.2.1 Model Aggregation

The need for appropriate aggregation of process models (*e.g.* equipment, tasks, demands, *etc.*) has been addressed by many authors (*e.g.* Macchietto, 1993; Bassett *et al.*, 1994). The level of aggregation required within each application largely depends on the types of decisions to be made, the local objectives and the ability of the algorithms to handle complexity and problem size. A proper aggregation of process models is important both for reducing the complexity of problems as well as for ensuring feasible and accurate production schedules, control, *etc.* The following two examples are used to illustrate how a proper aggregation of processing tasks (a modelling choice) can be used to ensure that decisions made at different levels of aggregation are consistent and the resulting models are simple.

3.2.1.1 Transfer Tasks

For optimal planning and scheduling, we utilise models based on the STN formulation (Kondili *et al.* 1993). For simplicity, instantaneous transfers of material and complete freedom in the transfers of material throughout the plant are typically assumed. However, in most processes, transfer operations take a large proportion of the total production cycle. Neglecting transfer times may result in a schedule which is either inaccurate or infeasible for control execution.

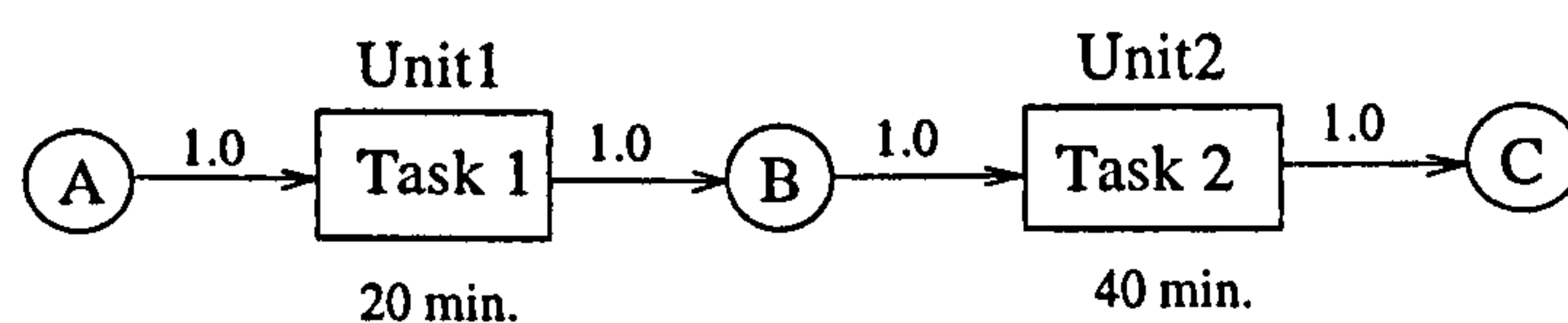


Figure 3.5 STN model without consideration of material transfer

Consider for example a process to produce product C through two processing stages (Task 1 of 20 minutes and Task 2 of 40 minutes) allocated respectively to Unit 1 and Unit 2. The STN model of the process without modelling the transfer of material between the two units is shown in Figure 3.5.

Clearly, Unit 1 will be free immediately after Task 1 is finished, and the intermediate material (state B) is instantaneously available for Task 2 in Unit 2. A feasible allocation of tasks with this model is shown in Figure 3.6(a). Of course, in actual operation, the transfer operation must be considered during which both Unit 1 (source) and Unit 2 (sink) are allocated to the transfer.

Crooks (1992) has discussed how such problems may be overcome by introducing extra parameters and constraints in the MILP formulation. "HOLD" tasks are introduced to deal with material transfer problems and constrained to be allocated to the units immediately preceding or following tasks which interact with explicit material transfers in the MILP formulation. With this MILP formulation, the resulting schedule is as shown in Figure 3.6(b). However, this introduces extra complexity into the MILP formulation and increases the number of tasks and states in the models, which in turn, results in a more difficult scheduling problem.

Alternatively, and more efficiently, the time (TTime) for the transfer task (10 minutes in the example) may be implicitly aggregated/embedded in both the source task (*e.g.* Task 1) and sink task (*e.g.* Task 2). This is achieved by a relatively simple modification of the basic MILP

formulation of Kondili (1988) and Kondili *et al.* (1993). The basic material balance constraint in the original formulation was modified to accommodate feed times of input states to a processing unit, thus enabling the proper synchronisation of the tasks in the two units. The resulting more accurate schedule is shown in Figure 3.6(c). The STN model in gBSS input language for synchronising the material transfer above is given in Figure 3.7.

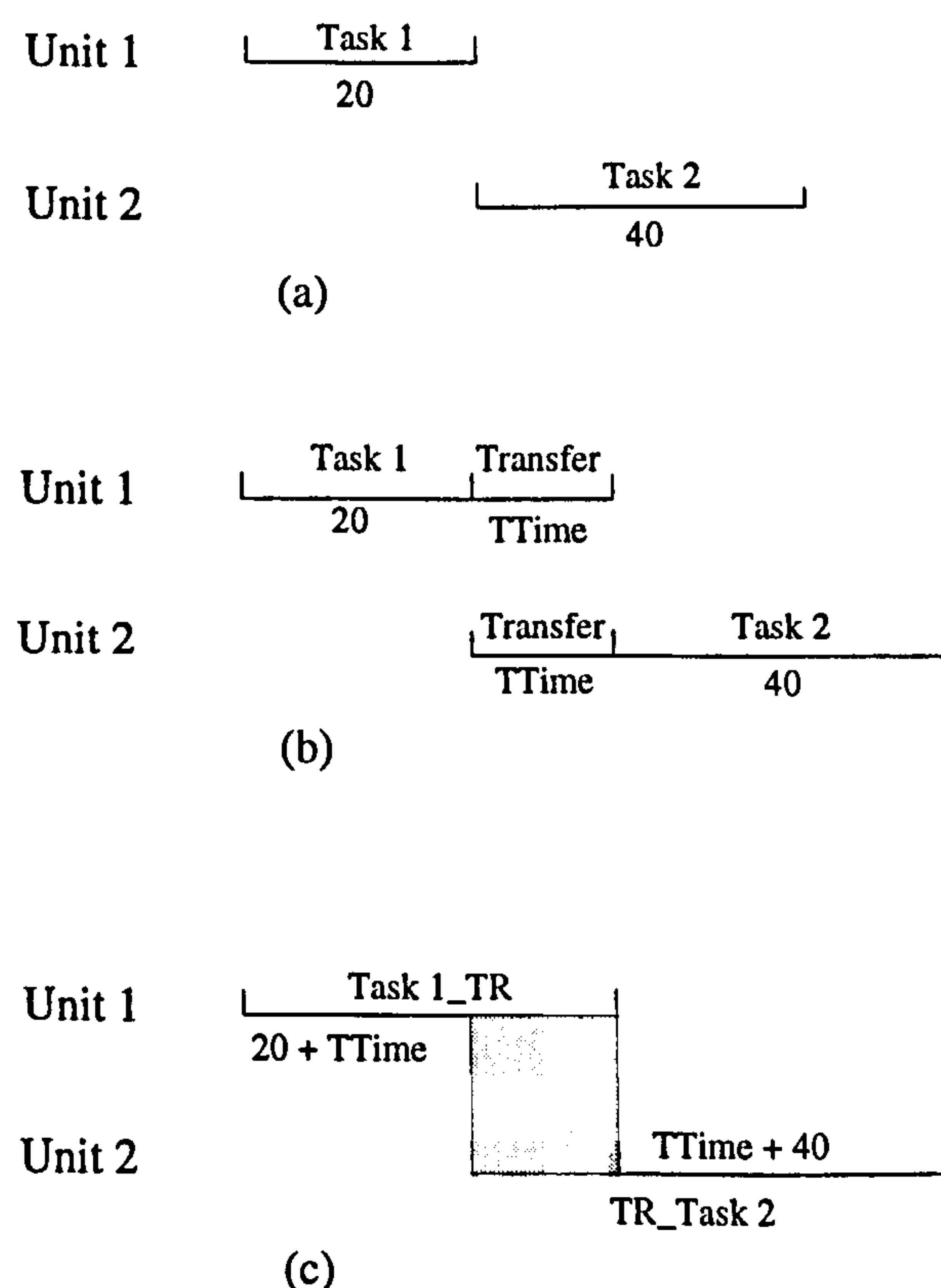


Figure 3.6 Modelling material transfers: (a) not modelled; (b) modelled explicitly; (c) modelled as "hidden" task.

```

STATE A
STATE B
STATE C

TASK Task1_TR
  INSTATE A FRACTION 1.0
  OUTSTATE B FRACTION 1.0 PROC_TIME 30

TASK TR_Task2
  INSTATE B FRACTION 1.0 FEED_TIME 10
  OUTSTATE C FRACTION 1.0 PROC_TIME 50

```

Figure 3.7 STN input file in gBSS for modelling material transfers

3.2.1.2 Sequential Tasks

When sequential tasks are to be performed in the same unit without change of resources, they may be aggregated into a single task for the purpose of production planning and scheduling. This aggregation not only simplifies the scheduling problem (without compromising optimality), but may also be necessary for computational efficiency.

For example, a recipe of a process is summarised in Table 3.1. The second column gives the required processing tasks in some detail, while the first column represents the same tasks in a more aggregated form. Due to operation and equipment constraints, Task1_A and Task1_B can only be performed in the same unit (T1 or T2) in sequential order. The transfer task between the unit T1/T2 and T3 must be considered and is synchronised by the tasks Tr1_To and Tr1_From in the detailed model. Task1_A, Task1_B and Tr1_To are therefore aggregated into *Task1*. Similarly, Task2_A, Task2_B, and Task2_C must be carried out in sequence and in T3 only. Together with the transfer tasks Tr1_From and Tr2_To, they may be aggregated into a single *Task2* task. An STN model of the process with aggregated tasks is shown in Figure 3.8.

Aggregated task	Operational details / disaggregated task	Unit / capacity
Task1	1) Task1_A 5 min. 2) Task1_B 5 min. 3) Tr1_To 10 min. Total time 20 min.	T1/ 100 L or T2/ 100 L
<i>Transfer</i> (hidden)	Synchronised by <i>Tr1_To</i> and <i>Tr1_From</i>	
Task2	1) Tr1_From 10 min. 2) Task2_A 5 min. 3) Task2_B 15 min. 4) Task2_C 75 min. 5) Tr2_To 10 min. Total time: 115 min.	T3/ 100 L
<i>Transfer</i> (hidden)	Synchronised by <i>Tr2_To</i> and <i>Tr2_From</i>	
Task3	1) Tr2_From 10 min. 2) Task3_A 10 min. Total time: 20 min.	T4/ 100 L or T5/ 100 L

Table 3.1 Aggregated tasks and processing time

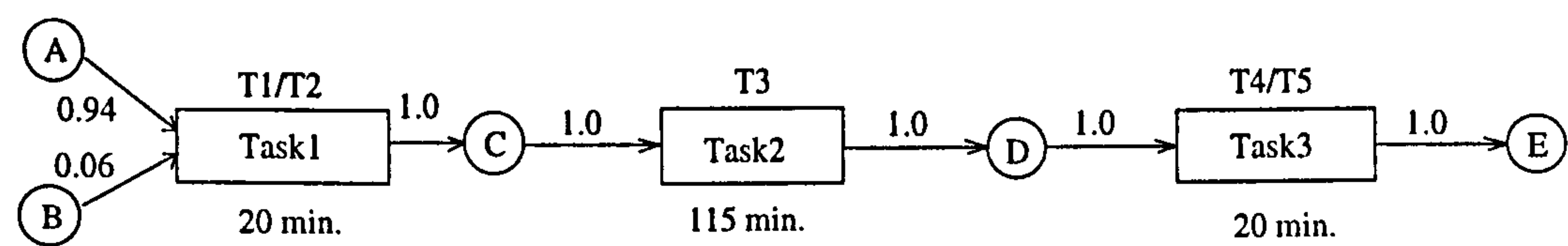


Figure 3.8 STN representation of the example process with aggregated tasks

Off-line optimal production schedules for the process with and without task aggregation were generated using the gBSS package. The production goal is to deliver 600 Kg of product E within a 12 hours time horizon. The scheduling objective is to deliver the product with the required quantity as soon as possible. Schedules are shown in Figures 3.9 and 3.10 with and without tasks aggregation, respectively.

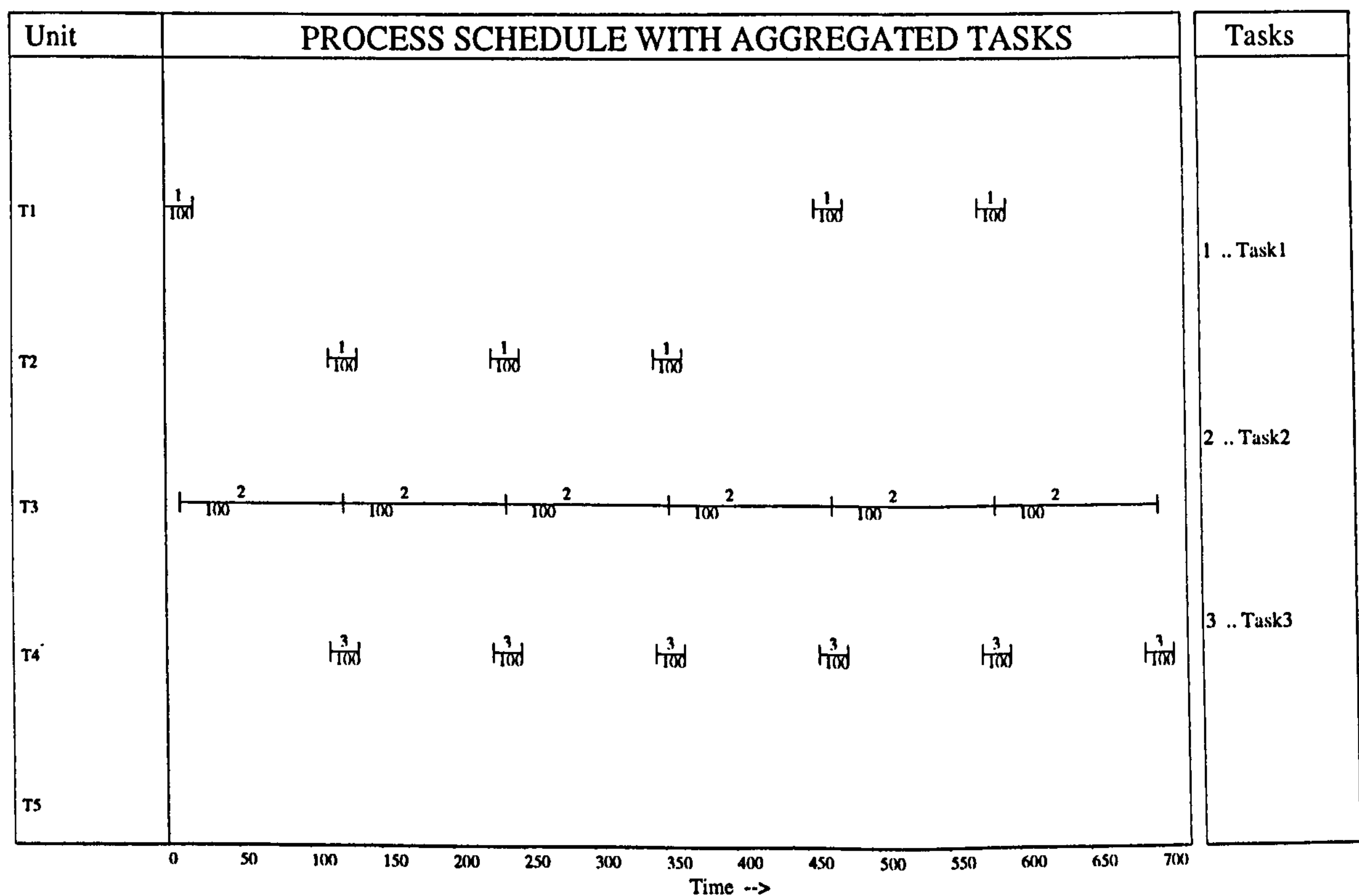


Figure 3.9 Scheduling results for the example process with task aggregation

gBSS vs. 1.1A (05 Sept 1994)

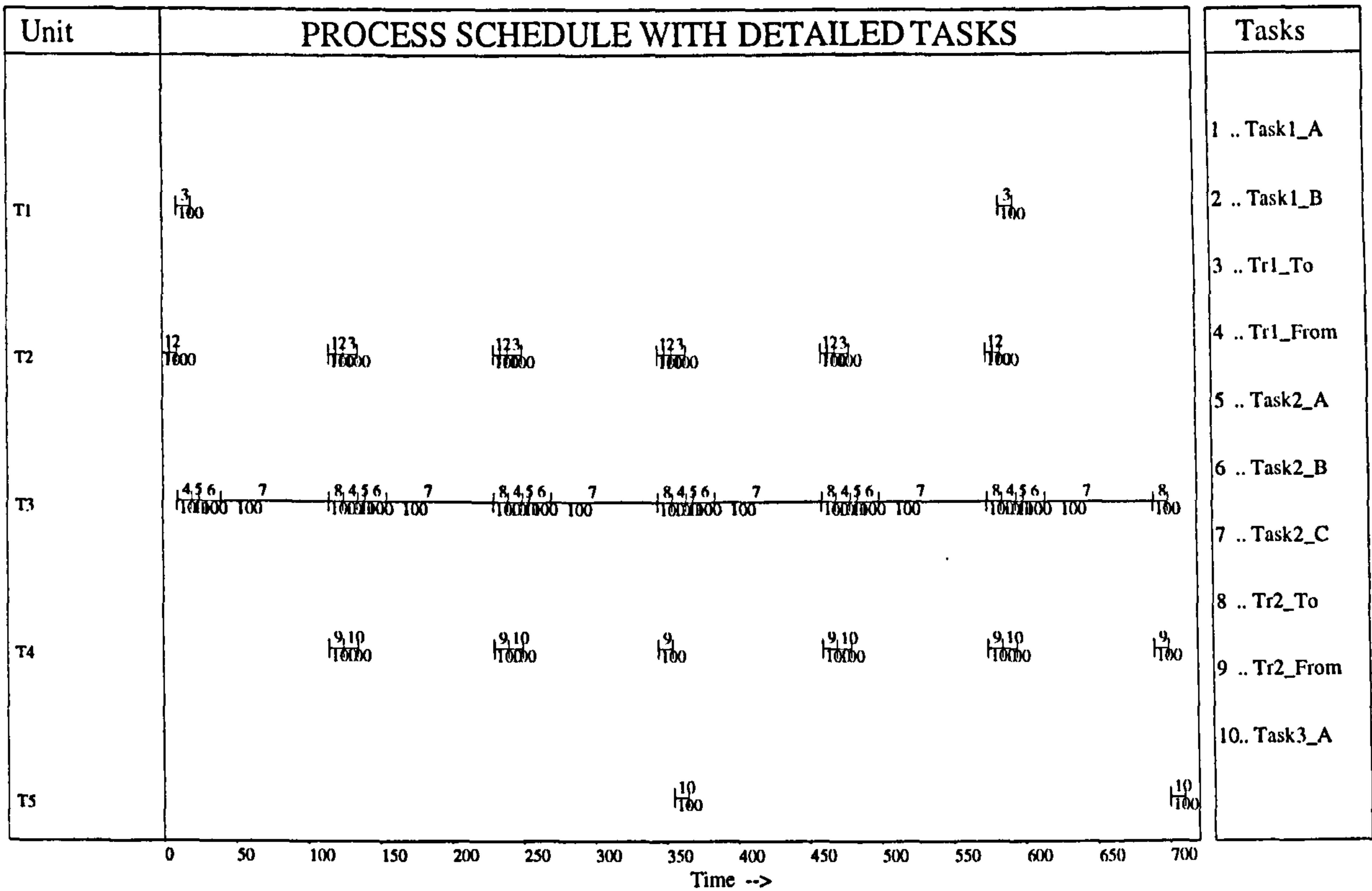


Figure 3.10 Scheduling results for the example process without task aggregation

The finishing time for the two cases is identical (710 min.). However, the allocation of tasks to the processing/storage units are different. With the aggregated tasks (Fig. 3.9), the first, fifth and sixth *Task1* tasks are allocated to T1, while the remaining *Task1* tasks are allocated to T2. Each *Task3* task is allocated to T4. Of the detailed tasks (Fig. 3.10), the first and sixth "equivalent to *Task1*" tasks (Task1_A, Task1_B and Tr1_To) are allocated to T1 (Tr1_To) and T2 (Task1_A and Task1_B). This is obviously infeasible in terms of operation as T1 is in fact empty. "Task3_A" tasks are allocated to T4 and T5 of which two (third and sixth) are infeasible. This example clearly shows that a proper aggregation of sequential processing tasks is necessary for obtaining feasible process schedules. If sequential processing tasks must be performed in a processing unit and other identical units exist, modelling the process without aggregation may lead to a solution which is infeasible at the supervisory control level, or may result in undesirable operations.

The computational details for the two cases are summarised in Table 3.2. For this simple case, the computation time for the aggregated model is less than 40% of the CPU time without the aggregation. The results show that task aggregation is also essential for optimal scheduling.

Computational Details	With Aggregation	No Aggregation
No. of Integer Variables	686	1556
Total No. of Variables	2097	4272
No. of Constraints	2278	4506
No. of Non-zero Elements	9680	15907
No. of Simplex Iterations	910	1152
CPU Time (s) Sun Sparc10	63	159

Table 3.2 Computational details for the solution of the example process

3.2.2 Design of Control Sequences for Flexibility

Sequence control plays a key role in batch process automation in ensuring efficient, safe plant operation and consistent product quality. This function is typically implemented by means of programmable logic controllers (PLCs) or sequence controllers in distributed control systems. In a multipurpose plant, a process unit may be used to perform processing tasks at different times which require different control procedures/sequences. Modification of existing control sequences is not only expensive as it requires recommissioning, but also it may easily lead to errors. Therefore, the design of control sequences to ensure flexible plant operations must be considered at the early stage of functional specification, as well as during the development of sequence control code.

Control sequences may be broken down into sub-sequences to carry out control functions which are broadly independent or where limited interactions are best handled at a higher level. As for supervisory tasks, a careful analysis is required to identify such interactions, which may be due to shared use of resources (interlocks, *etc.*). Thus sequences will be naturally organised in a hierarchical fashion, with initiation commands, parameters, termination commands, alarms, *etc.*, suitably communicated between all levels. A typical control sequence structure proposed in this study is shown in Figure 3.11.

The "top sequence" is often designed to carry out and control a main processing task (*e.g.* a temperature control sequence for a chemical reaction), while the sub-sequences are typically used to perform some auxiliary control functions (*e.g.* a flow monitoring sub-sequence which can only be active when the transfer control sequence is active). A top sequence (*e.g.* seq-A)

may also be called by another top sequence (*e.g.* seq-B). In this case, seq-A becomes a sub-sequence of the seq-B. For example, a pH control sequence (top sequence) can be executed individually to control the pH value of a reactant to a desired level. However, if this sequence is called by a reaction control sequence, it is then transformed into a sub-sequence of that reaction control sequence.

Different sets of sub-sequences may be called by a top sequence depending on particular process requirements. In such cases, "conditional switches" may be used in the top sequence to represent the different control requirements and to trigger a different set of sub-sequences. The different control requirements are "parameterised" by means of variables, the values of which may be specified in a control phase of the supervisory control model and passed to a control sequence.

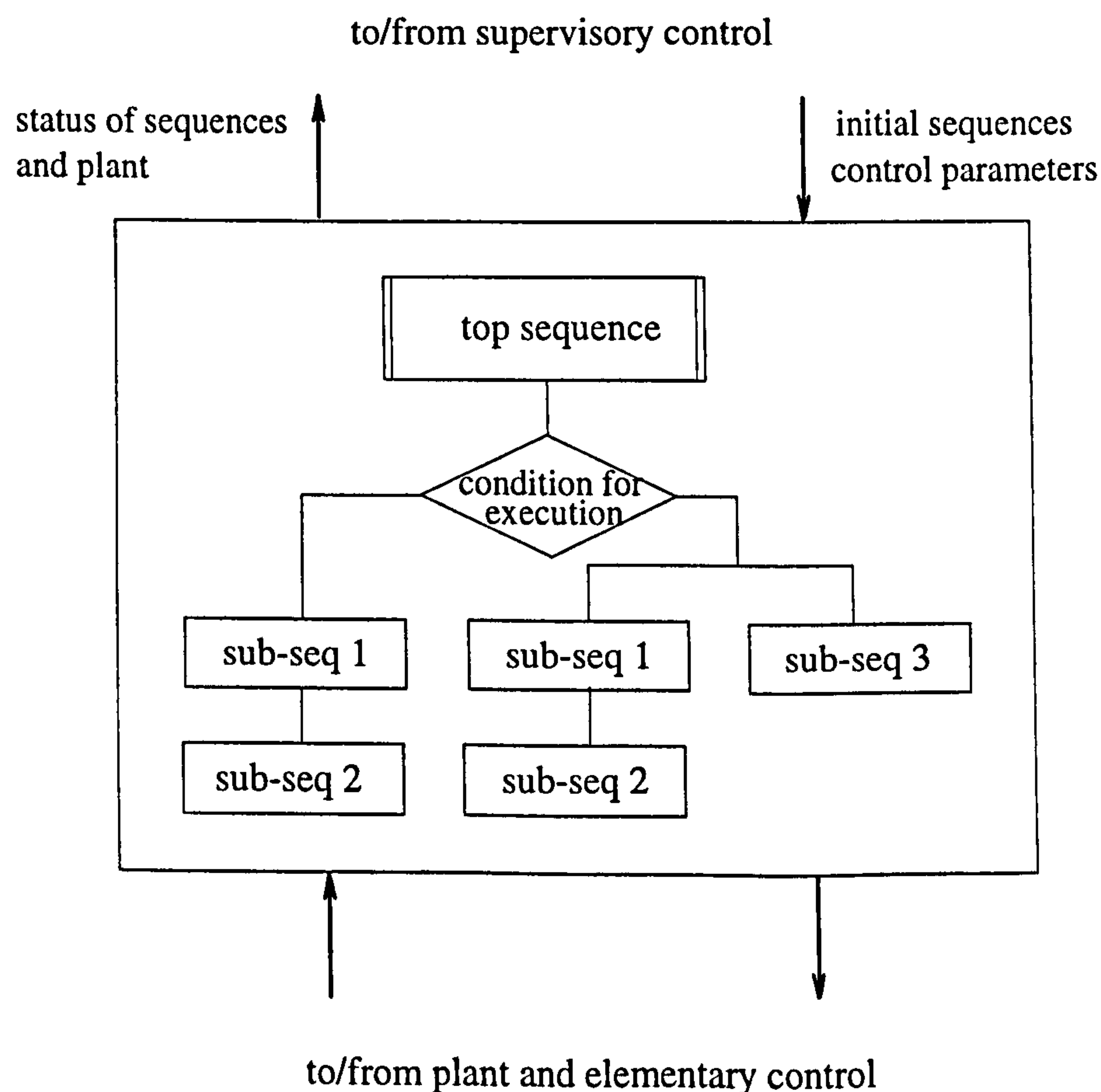


Figure 3.11 Control sequence structure

In our experience, maximum flexibility of the sequence control code is obtained by disaggregating control functions into sub-sequences as small as possible, so that they can be recombined into different control structures without further modification.

3.2.3 Models Hierarchy and Coordination

In order to enable an automatic translation of optimal solutions (based on the MILP method) to control phases and master procedures for the supervisory control, Crooks (1992) and Chua (1995) modelled processes in sufficient detail such that there is a one-to-one correspondence between an STN task and a control phase (with the only exception being an instantaneous transfer, for which the control phase is not mapped to a scheduled task in optimal scheduling level). This type of mapping is also useful to provide suitable initial conditions for performing re-scheduling at the optimal production planning and scheduling level. However, this increases the complexity and computational difficulty of the scheduling problem. For example, it took nearly 10 hours (CPU time) to obtain an initial feasible (also optimal) solution for a very simple process as reported by Chua (1995). The models for optimal scheduling must be by necessity rather aggregated.

When an aggregated process model is used for process planning and scheduling, the aggregated tasks must be decomposed into more detailed control entities (phases and procedures) at the supervisory control level to ensure feasible execution. For example, the aggregated Reaction_A task of section 4.2.2 must be decomposed into its individual phases (pH control, gelatinization and degradation). Furthermore, these control phases must be mapped to even more detailed equipment control sequences. Proper coordination between these entities must be established to enable them to be correctly linked.

In this study, each control phase is only mapped to a "top sequence" at the highest level of the control sequence hierarchy. For example, a *Sparge Heating Control* sequence (top sequence, which calls a level monitor, an emergency control and an analogue control loop sub-sequences) is used to carry out the temperature control of the starch gelatinization and is mapped to the *Gelatinization* phase at the supervisory control level. A schematic representation of the model hierarchy and coordination is shown in Figure 3.12.

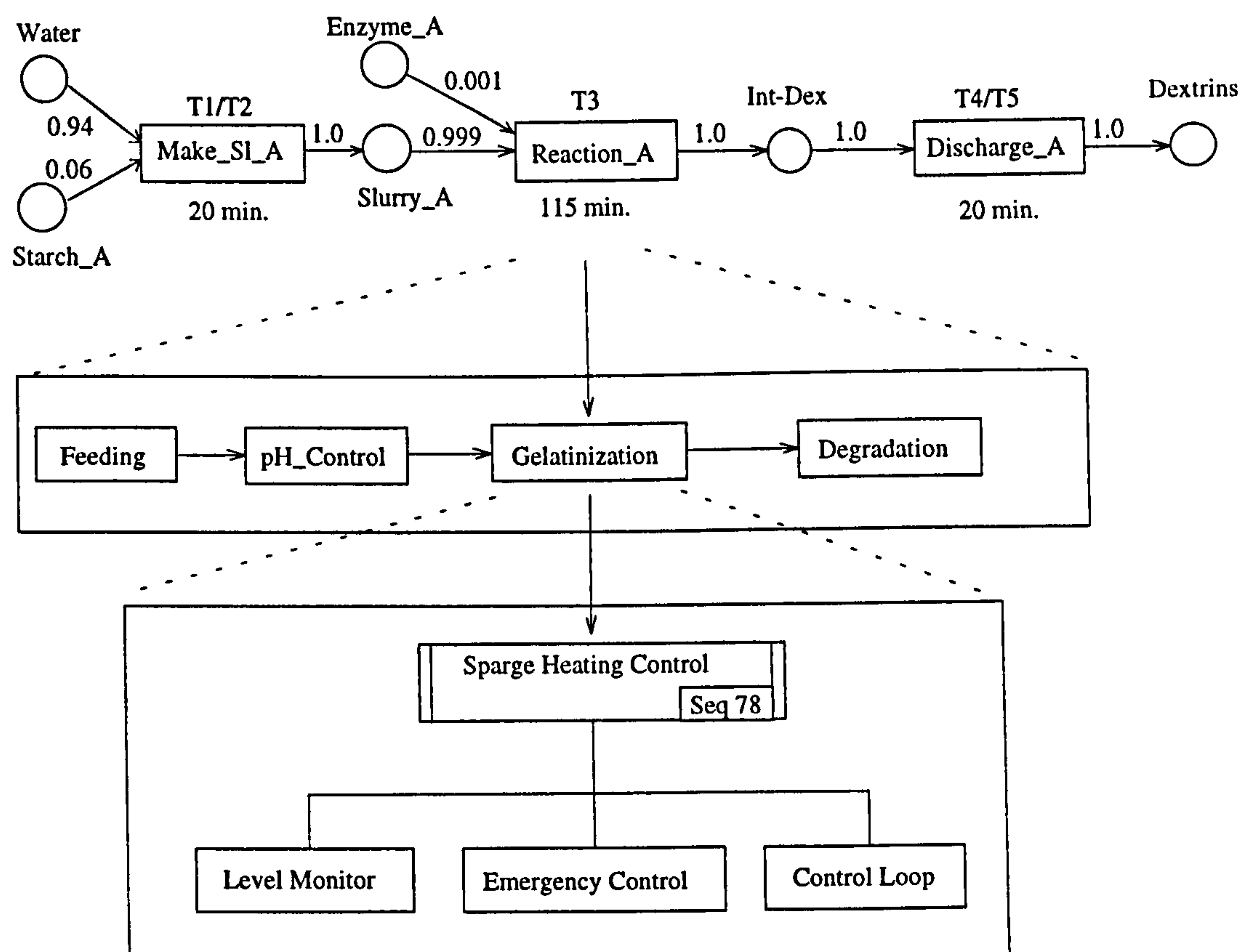


Figure 3.12 Model hierarchy

3.3 Summary and Conclusions

A hierarchical and model-based operations support system was presented in this Chapter. The system supports optimal production planning, supervisory control, sequence control and advanced model based control functions which can be tested or performed either individually or in an integrated manner. To ensure the feasible execution of scheduled activities and of an entire production plan, and to ensure local control decisions are properly propagated, a proper coordination of models at various levels of resolution must be established. To this purpose, the development of control sequences and supervisory control models were therefore not treated as distinct engineering activities, but were carried out simultaneously so as to ensure that this coordination is built-in by design. The need for better control of unit operations becomes more important in the integrated operation environment to suppress disturbances locally and to provide a robust performance in the face of process variations.

As observed, the key to proper coordination between all production control levels is a careful analysis and definition of model aggregations to be used at each level. In order to improve computational efficiency and to ensure a feasible process schedule, a more aggregated model was required for production planning and scheduling. A simple and efficient way to model material transfers was presented that utilises the existing MILP formulation.

To fulfil a production plan and process schedule, aggregated tasks must be decomposed into more detailed control phases and control procedures at the supervisory control level. Additional information regarding the detailed control parameters, logical precedence between the processing phases, resource utilisation, equipment connections and control system information (*e.g.* tags) are specified at this level. This information is required to ensure the feasibility of process schedules and to link with a control system for on-line supervisory control.

To support flexible batch plant operation, control sequences must also be designed in such a way that they can be easily combined to satisfy different process requirements, without the need for modifications to the constituent smaller control sequences. Hierarchical control sequence structure and "parameterisation" of the control requirements were used for the design of flexible control sequences.

Although decisions about model aggregations for use at various levels were done manually for the pilot plant example discussed, there is considerable scope for algorithmic techniques to support this important design activity. Some initial results, described in Crooks *et al.* (1992), Kuriyan *et al.* (1994) and Chua (1995), indicate that identifying the model aggregation to be used for supervisory control, control procedures and main control phases (top sequences) can be to some extent automated. At a much more detailed level, the synthesis of individual control sequences and subsequences is also a promising candidate for automation (Rotstein *et al.*, 1995).

Chapter 4

Modelling a Multipurpose Batch Pilot Plant

In this chapter, a model is presented of a multipurpose batch pilot plant in the Department of Chemical Engineering, Imperial College. Two starch based processes are considered, with a cleaning-in-place (CIP) operation. To achieve a properly integrated operations support for the batch plant, operation models must cover a number of aspects:

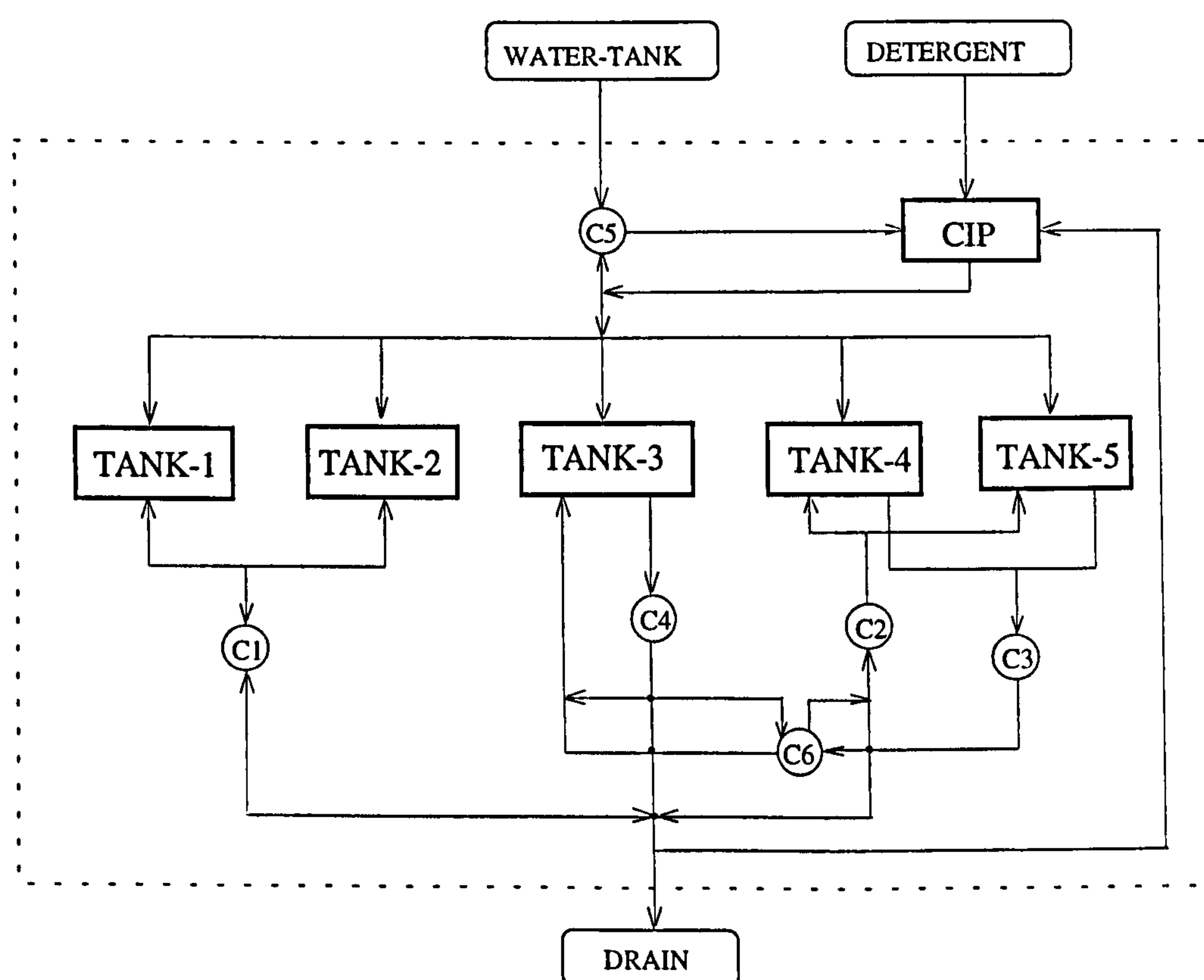
- A model of the plant (equipment and its connections);
- A model of the process (recipes);
- A model of plant control and operations (control recipes, control sequences, *etc.*);
- A model of the plant dynamics.

The equipment models are discussed in section 4.1. Section 4.2 describes the models of the processes in the form of STNs. Operations control models for the plant are detailed in section 4.3 and 4.4. Dynamic models of the plant and model validation are discussed in section 4.5. Finally, a summary of Chapter 4 is given in section 4.6.

4.1 Model of Equipment

For production planning, scheduling and supervisory control, an aggregated equipment model is typically considered to reduce the size and complexity of those problems. However, this aggregated model must be decomposed into a more detailed item level model for the purpose of generating control sequence specifications and for control sequence validation (Crooks, 1992). A hierarchical approach is therefore used in this work for modelling the plant equipment.

In order to deal with the existence of parallel processing units and their restrictive inter-unit connections, Cott (1989) introduces two levels of equipment description at the aggregated model level: the *unit* and *resource*. A unit consists of the physical equipment (typically, processing or storage vessels), whereas a resource defines a group of functionally equivalent units, which can perform the same set of operations. A plant flowsheet can then be defined as a directed network of units or resources.



(C1) - AVI-40; (C2) - Rtn-route; (C3) - P7; (C4) - P9; (C5) - HE3; (C6) - HE1

Figure 4.1 Unit level batch pilot plant description

An aggregated unit level equipment configuration of the batch pilot plant is shown in Figure 4.1. Here the resource and unit levels of equipment description are identical since no parallel equivalent units exist. The plant flowsheet (Figure 3.2) is decomposed into six interconnected equipment units (TANK-1, TANK-2, TANK-3, TANK-4, TANK-5 and CIP), each of which represents an aggregated plant unit composed by a main vessel and associated auxiliary instrumentation. The Water-Tank, Det-Tank and Drain units (Figure 4.1) are not

physically pieces of equipment included in the plant P&ID, however, they are modelled as storage units to represent the connections between the plant and the external world.

Further analysis of the plant operations suggests that additional constraints associated with limited unit connectivity must also be considered to prevent conflicting operations (*e.g.* recirculation of material in TANK-4 and TANK-5 cannot occur simultaneously as both recirculations share a common transfer route). A number of "critical elements" in the transfer/recirculation routes are then identified and modelled as "common resources" with an availability of one. These elements are shown in Figure 4.1 as small circles. Each circle corresponds to a plant item in the P&ID (*e.g.* C1 represents the valve AV1-40; C2 represents a pipe segment between the DDV1-6 and AV1-43/45, *etc.*).

Figure 4.2 shows schematically the utilization of these common resources during material transfers and recirculations between units. In the figure, units shown on the left hand side are source units (*i.e.* units from which material is discharged). Units on the top of the figure are sink units (*i.e.* units to which material is delivered). Nodes on the crossed dotted lines indicate a feasible transfer/recirculation (*i.e.* units are connected). For example, no transfers can be performed from TANK-1 to TANK-1 (recirculation), or to TANK-2, while transfers from TANK-1 to CIP, TANK-3, TANK-4, and TANK-5 are possible. However, each transfer requires resource C1 (which is indicated on the dotted line). Transfers from TANK-1 to TANK-4 and to TANK-5 also requires resource C2. Similarly, all transfers from the CIP unit require resource C5. If one transfer (*e.g.* transfer from CIP to TANK-1) is taking place, transfers of material between the CIP unit and any other unit are infeasible as only one resource C5 available. Thus only one cleaning operation can take place at any time. All possible transfers and recirculations are represented in Figure 4.2 in a similar fashion and in a compact way. The only exception is resource C6, which may or may not be required for the recirculation of TANK-3, TANK-4 or TANK-5 since there are alternative feasible paths.

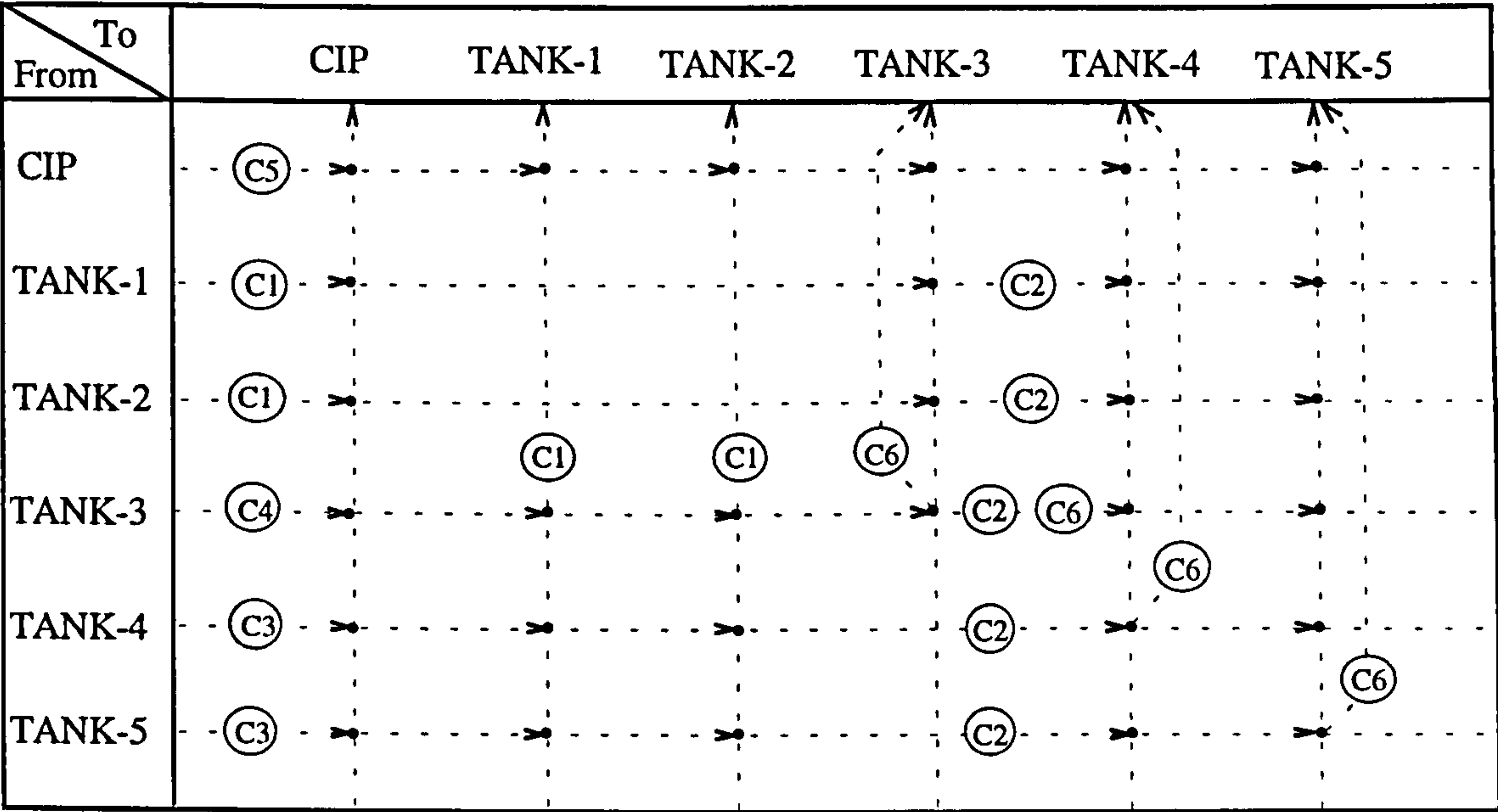


Figure 4.2 Common resource utilization during the transfers/recirculations between the units

For the purpose of control sequence validation and simulation, the aggregated plant units (modules) shown in Figure 4.1 are decomposed to the item level (*e.g.* valves, pumps, controllers, *etc.*) as shown in Figure 4.3. At the item level, all equipment found on the plant P&ID is included. The decomposition is *via* submodules comprised of simple elementary hierarchical models. For example, the aggregated unit TANK-3 is further decomposed into a *Reactor* (a primitive item which is represented by a rounded square), a *Jacket-system* submodule (represented by a square that can be further decomposed), and a *Sparge-system* submodule. These submodules and primitive items are interconnected by either flow streams (solid lines with arrows representing the flow direction) or control and measurement signals (dotted lines). After this initial decomposition, the development of models for each of these submodules can be considered in isolation.

The *Sparge-system* performs one of the reactor temperature control functions. It consists of an analogue control loop (including controller, temperature probe, and a control valve) and an on/off valve. The reactor temperature measurement (Reactor.Tr) serves as an input signal to the *Sparge-system* submodule. The output of the submodule is the steam flow (Input_sparge) which is controlled by a feedback control loop. This flow acts as an external input to the Reactor. Similarly, the *Jacket-system* submodule (not detailed in Figure 4.3 for brevity) provides an alternative means of control of reactor temperature.

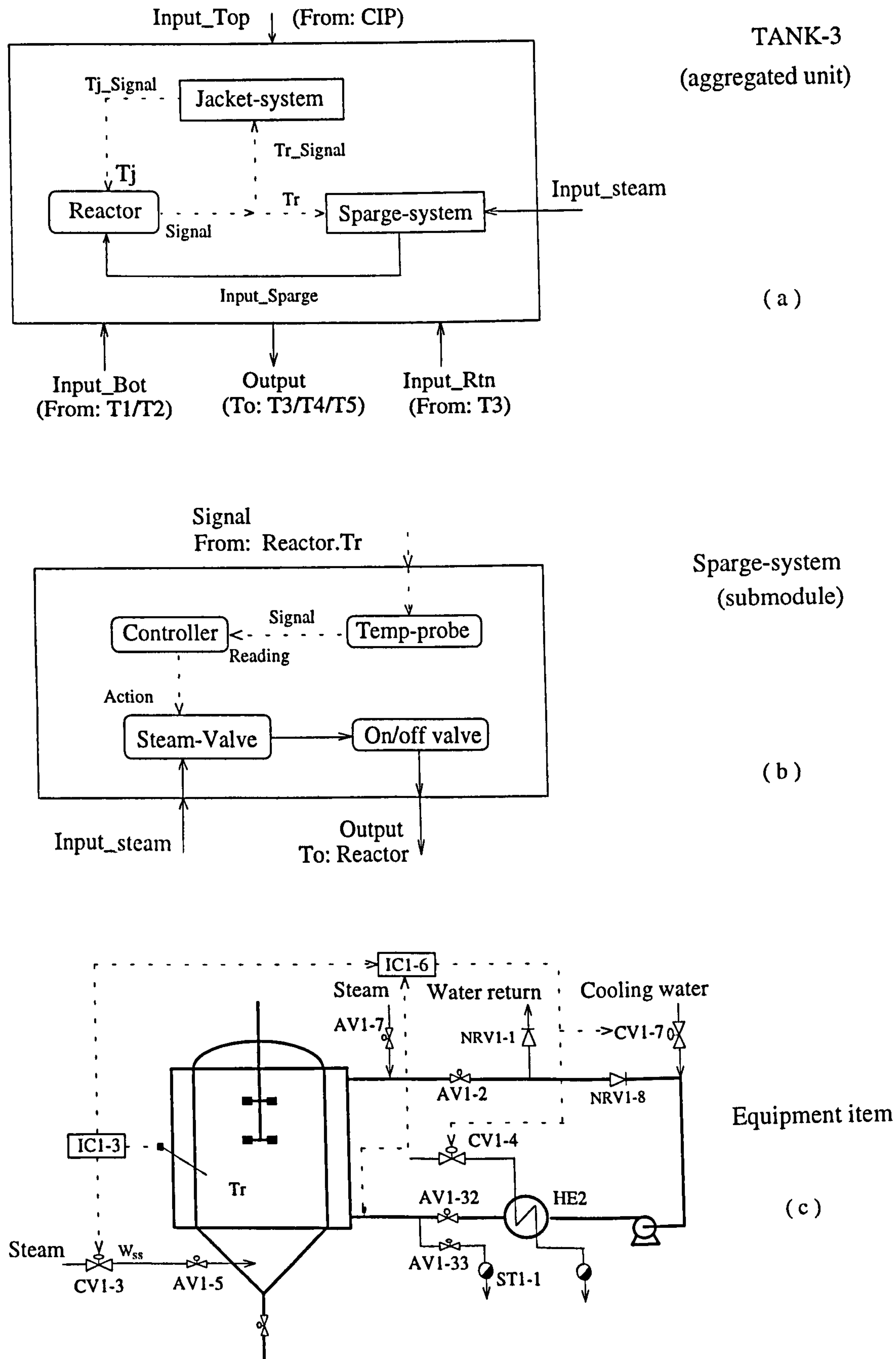


Figure 4.3 Hierarchical equipment models: (a) aggregated unit TANK-3; (b) sparge system submodule; (c) TANK-3 model at the item level

The item level equipment model for the aggregated unit TANK-3 is shown in Figure 4.1.c. An equipment model for the entire plant has been developed in this way, *via* a number of intermediate hierarchical levels comprised of interconnected primitive models.

4.2 Model of Processes - Recipe Models

In this section, process models for two enzymatic starch degradation processes and a cleaning-in-place (CIP) operation are developed in the form of STNs. These are directed networks of processing tasks that transform the raw materials into products, often *via* intermediate materials. The CIP operation does not transform process materials into useful products. However, it alters the state of process equipment (*e.g.* from dirty to clean) and this "process" is modelled in a similar way.

4.2.1 Process Descriptions

As one of the most widely distributed raw material, starch is widely used in the food and non-food industries. By applying physical, chemical or biochemical techniques, the properties of starch can be adapted in many ways. Two starch based processes are considered here, using different type of enzymes, operating procedures and process conditions.

As in most food production plants, cleaning operations are important in this pilot plant to ensure plant hygiene and to prevent product contamination. Intermediate cleaning is typically needed if a unit is used to perform different tasks and at the end of productions.

Starch liquefaction using α -amylase. Wheat starch (Starch_A) is mixed with process water to make slurry (Slurry_A). This can be done in either TANK-1 or TANK-2. The proportions are 6 mass units of dry starch to 94 mass units of process water. The slurry is agitated for 5 minutes, then transferred to the reactor vessel TANK-3. The pH of the slurry is adjusted to a setpoint (6.2). The mixture is heated to 105 °C by direct steam injection (sparge heating) and maintained at the setpoint for 15 minutes to produce a gel (starch gelatinization phase). The starch gel is then cooled to a temperature setpoint of 85 °C. A small amount (3.8 ml) of α -amylase (Termamyl 60L, by NOVO) is added to the starch gel to initiate the degradation phase. The reaction is continued for 75 minutes to produce the final product (a mixture of several

dextrins). The final product is then discharged from the reactor to a product vessel (TANK-4 or 5), and from there to a further destination (here modelled as DRAIN).

Cyclodextrins process using Cyclodextrins Glucocyltransferases (CGTase). A soluble corn starch (Starch_B) is mixed with process water to make a starch solution (in TANK-1 or TANK-2) in a mass ratio of 6:94. The solution is agitated for 5 minutes, then transferred to either the reactor TANK-3 or to TANK-4/5 at a flow rate of 8.0 l/min. A different enzyme, *CGTase*, is added to the reaction vessel which is then heated to and maintained at a temperature of 65 °C. The temperature control is *via* the water jacket (if the reaction is performed in TANK-3) or by recirculation of the material through a heat exchanger (if the reaction is carried out in TANK-4/5). The reaction takes 70 minutes. The final product (a mixture of cyclodextrins) is discharged from the reaction vessels.

Cleaning-In-Place (CIP). The batch pilot plant has five cleaning routes, one for each main processing vessel. Only one route can be cleaned at any one time. Each clean cycle consists of three cleaning phases, including *pre-rinse* with soft water for 5 minutes, *detergent clean* for 10 minutes and *final-rinse* with soft water for 5 minutes. A route is said to be cleaned only if all three cleaning phases are complete. The CIP service operation, which can be carried out independently, is required to prepare the detergent solution with desired properties (batch size, temperature and conductivity). The detergent is a solution of NaOH in water at the conductivity of 300.0 µS.

4.2.2 State-Task-Network Models

State Task Network models for the above processes are developed according to the following rules:

- sequential tasks which are performed in the same unit are aggregated into a single task, as illustrated in Chapter 3;
- the time for all transfer operations is aggregated into source and sink tasks using the modelling method discussed in Chapter 3;

The State Task Network models for the two starch conversion processes and the CIP are shown in Figure 4.4.a/b/c, with time in minutes and proportions of ingredients by mass. The equipment required and their suitabilities for performing these STN tasks are summarised in

Table 4.1. The details of the STN tasks and utilities used within each process are given in Table 4.2.

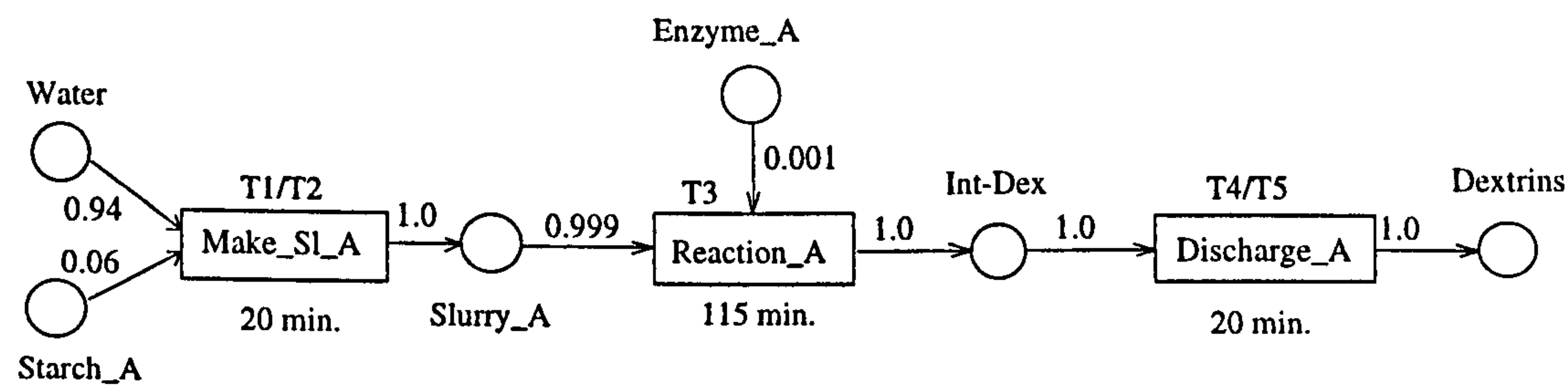


Figure 4.4.a STN representation of the *Dextrins* process

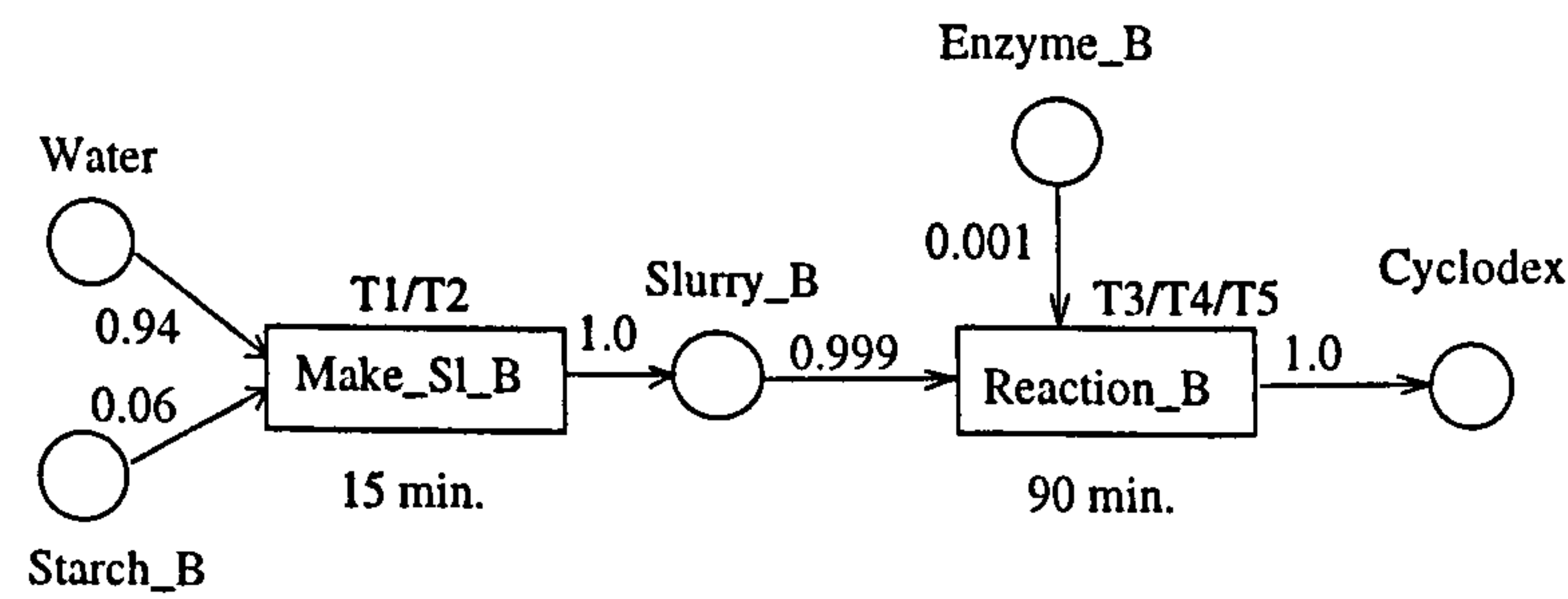


Figure 4.4.b STN representation of the *Cyclodextrins* process

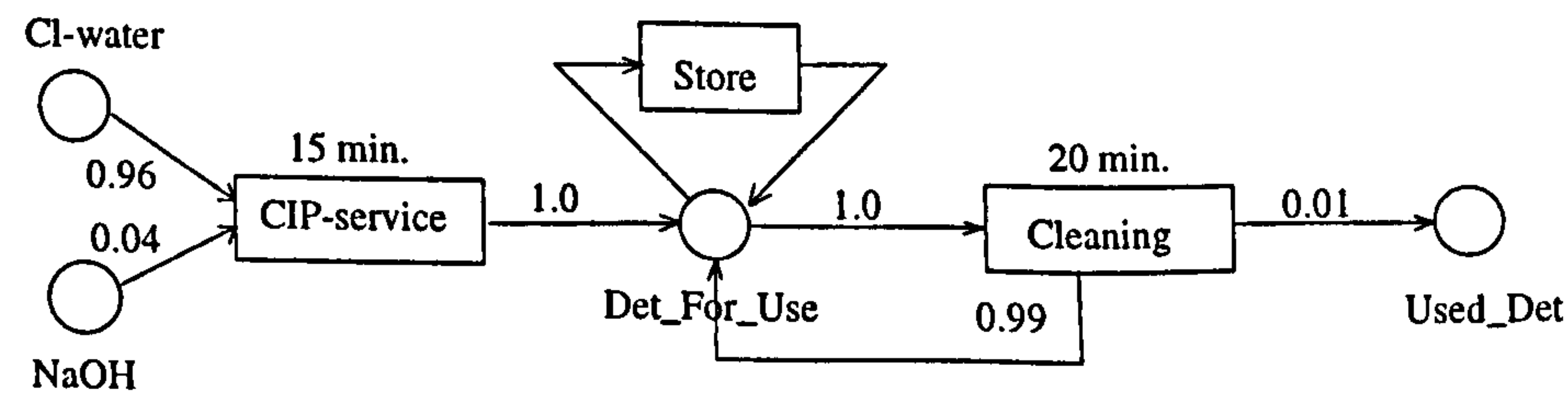


Figure 4.4.c STN representation of the cleaning-in-place (CIP) process

Aggregated Unit	Capacity (l)	Suitability to STN Tasks
TANK-1 (T1)	100	Make_Sl_A, Make_Sl_B, Cleaning
TANK-2 (T2)	100	Make_Sl_A, Make_Sl_B, Cleaning
TANK-3 (T3)	100	Reaction_A, Reaction_B, Cleaning
TANK-4 (T4)	100	Reaction_B, Discharge_A, Cleaning
TANK-5 (T5)	100	Reaction_B, Discharge_A, Cleaning
CIP	100	CIP_Service, Store Det_For_Use

Table 4.1 Summary of equipment and suitability

STN Task	Duration (min.)	Input State	Output State	Utility Usage	From -To Time (min.)
Make_Sl_A	20	Starch_A	Slurry_A	HE3	0 - 5
		Water		AV1_40	10 - 20
Make_Sl_B	15	Starch_B	Slurry_B	HE3	0 - 5
		Water		AV1_40	5 - 15
Reaction_A	115	Slurry_A	Int_Dex	P9	105 - 115
		Enzyme_A		Rtn_route	105 - 115
				HE1	105 - 115
Reaction_B	90	Slurry_B	Cyclodex	P7	0 - 90
		Enzyme-B		Rtn_route	0 - 80
				HE1	0 - 80
Discharge_A	20	Int_Dex	Dextrins	P7	10 - 20
Cleaning	20	Det_For_Use	Det_For_Use	HE3	0 - 20
			Used_Det		
CIP_Service	15	Cl_Water	Det_For_Use	HE3	0 - 15
		NaOH			

Table 4.2 Summary of STN tasks utility information

The STN model for each process is rather simple, as required for optimal scheduling purposes. However, the complexity of operations is increased considerably when production of both starch processes is considered, with an intermediate cleaning operation. In such cases, additional operational constraints must be considered to represent the state of the processing unit (*e.g.* clean, dirty with material A, *etc.*), and to represent how the various processing tasks modify the state of the processing unit.

Crooks (1992) introduced the notion of unit state (*UState*) to represent the state of a processing unit. A Unit State Task Network (USTN) was defined to represent how processing tasks modify the state of a processing unit. For example, a suitable USTN for the unit TANK-1 is shown in Figure 4.5.

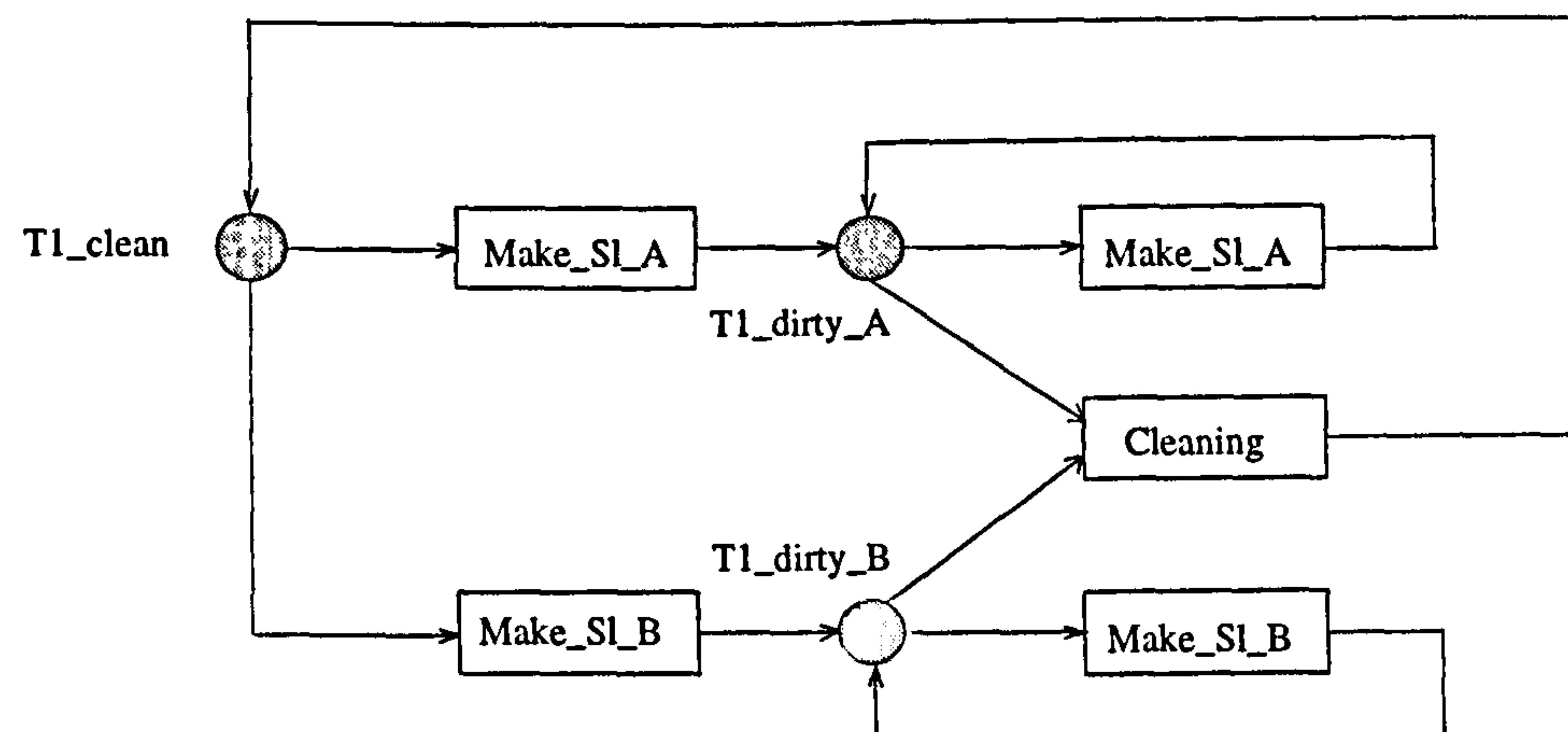


Figure 4.5 USTN representation for unit TANK-1

Process specifications require that TANK-1 must be cleaned before it is used to perform different types of tasks (*i.e.* switching from *Make_Sl_A* to *Make_Sl_B*, or vice-versa). No intermediate cleaning between batches is necessary if the unit is used to perform the same processing task. The USTN model in Figure 4.5 guarantees these requirements are satisfied. The figure shows that at any given time, the unit TANK-1 can exist in one of the three states: *T1_clean*, *T1_dirty_A*, and *T1_dirty_B*. The *Make_Sl_A* task changes the unit TANK-1 from state *T1_clean* to state *T1_dirty_A*. From this state, the task *Make_Sl_B* cannot be performed unless the *cleaning* task is carried out, which will change the unit from *T1_dirty_A* to *T1_clean*. By appropriate modelling of the *UStates* and transitions, any desired sequence of operations may be enforced on a unit.

4.3 Supervisory Control Procedures - Control Recipe Models

To operate a batch plant, information regarding its control parameters, control phases and logical phase precedence relationships must be specified. For this purpose, Cott (1989) defines *master procedures* as directed networks of process and transfer phases. In this way, the supervisory control system can delay phase start times until prerequisite phases are complete.

The idea of batch identity is associated to a master procedure which is defined to include all related operations on a well identified "batch" of material. A master procedure thus may include several tasks, and is typically bounded between storage locations where batches may be merged or split (and thus the identity of the batch lost). Thus, for a process with storage units, several master procedures may be required to make intermediate and final products. For the pilot plant, TANK-4/5 and the CIP vessel are modelled as storage units. However, certain types of processes might be performed in these storage units using the "process in storage" function in SUPERBATCH version 1.06. There are 11 master procedures in total for the processes described in the previous section. An example of a master procedure for the dextrans process is shown in Table 4.3. The weight basis (batch size scaling parameter) is defined at the beginning of the master procedure, followed by nine control phases (the procedure beginning with a START phase and ending with a END phase) for which precedence relationships are specified. In each control phase, a nominal processing time (either as an absolute time or as a rate basis), the usage of common resources, and any control parameters and their desired values (*e.g.* the temperature setpoint in STEP-6, flow rate setpoint in STEP-3, *etc.*) to be passed to a control system are also specified.

In order to initiate, execute and monitor the production on-line, information regarding to the links of control phases with specific units in the plant and the control system must also be provided. This is specified in a real time control model. The real-time model defines the relation between the generic procedures and control phases in the resources model and specific plant units. Furthermore, it defines the mapping between the specific status variables defined in the control recipe model and the corresponding entities in a control system. Status variables include the current time and an availability tag for each plant unit (a unit is out of service if the tag value is not 0). Tag names are specified for passing control parameters (*e.g.* the flow rate setpoint, temperature setpoint, *etc.*) to the control system. Two other parameters must always be given. The first is called a *condition code* which allows the supervisory control procedure to communicate to the control sequence the existence of an error or abnormal condition during the execution of the unit phase (*e.g.* alarm, hold, completed successfully, *etc.*). The second is the *weight parameter*, which passes the batch size to the control sequences. The value of the weight parameter is specified in the control recipe model, and it is sent to the control system together with the command to start the phase. An example of the real time model corresponding to the master procedure of Table 4.3 is given in Table 4.4.

master procedure section		phase name		STEP_5
/		resource phase		SPARGE_H_T3
master procedure	PA_T1T3T4	preceding phase		STEP_4
weight basis	89.0	stability		STABLE
/		processing time		15.0
phase name	START	time type		TIME
/		resource		WATER_TANK
phase name	STEP_1	transfer		-12.0
resource phase	TR_W_T1	material		WATER
preceding phase	START	unit selection		ANY MATERIAL
stability	UNSTABLE	resource		TANK_3
processing time	5.0	transfer		12.0
time type	TIME	parameter		TEMPERATURE
resource	WATER_TANK	value		105.0
transfer	-77.0	common resource		STEAM
material	WATER	usage		20.0
unit selection	ANY MATERIAL	/		
resource	TANK_1	phase name		STEP_6
transfer	77.0	resource phase		WATER_H_C_T3
material	SLURRY	preceding phase		STEP_5
unit selection	ANY MATERIAL	stability		STABLE
common resource	HE3	processing time		75.0
usage	1	time type		TIME
/		parameter		TEMPERATURE
phase name	STEP_2	value		85.0
resource phase	MIXING_T1	common resource		ELECTRICITY
preceding phase	STEP_1	usage		400.0
stability	UNSTABLE	common resource		STEAM
processing time	5.0	usage		300.0
time type	TIME	common resource		COOLING_WATER
/		usage		1000.0
phase name	STEP_3	common resource		HE2
resource phase	TR_T1_T3	usage		1
preceding phase	STEP_2	/		
stability	UNSTABLE	phase name		STEP_7
processing time	10.0	resource phase		TR_T3_T4
time type	TIME	preceding phase		STEP_6
resource	TANK_1	stability		STABLE
transfer	-77.0	processing time		8.9
material	SLURRY	time type		RATE
unit selection	ANY MATERIAL	resource		TANK_3
resource	TANK_3	transfer		-89.0
transfer	77.0	resource		TANK_4
parameter	BATCH_SIZE	transfer		89.0
value	76.0	material		DEXTRINS
parameter	FLOW_RATE	unit selection		ANY MATERIAL
value	0.16	common resource		HE1
common resource	AV1_40	usage		1
usage	1	common resource		PUMP_9
/		usage		1
phase name	STEP_4	common resource		Rtn_Route
resource phase	PH_CONTROL	usage		1
preceding phase	STEP_3	parameter		TEMPERATURE
stability	STABLE	value		80.0
processing time	5.0	/		
time type	TIME	phase name		END
parameter	PH_SetP	preceding phase		STEP_7
value	6.20	stability		STABLE

Table 4.3 Master procedure for the dextrins process

real-time plant status section			
/		/	
current time	SYS\$TIME	unit phase	T1_TO_T3
/		condition	RC_T1_T3
process unit	T_1	weight parameter	T1_T3_10
availability tag	T1_Tag	resource phase	TR_T1_T3
inventory tag	IT1_1_T1	process resource	TANK_1
process unit	T_2	process unit	T_1
availability tag	T2_Tag	process resource	TANK_3
inventory tag	IT1_2_T2	process unit	T_3
process unit	T_3	parameter	BATCH_SIZE
availability tag	T3_Tag	tag name	T1_BS
inventory tag	IT1_5_T3	parameter	FLOW_RATE
/		tag name	T1_F_RATE
storage unit	T_4	/	
availability tag	T5_Tag	unit phase	T3_PH
inventory tag	IT1_14_T5	condition	RC_T3_PH
storage unit	T_7	weight parameter	T3_WP_40
availability tag	T7_Tag	resource phase	PH_CONTROL
inventory tag	IS1_1_T7	process resource	TANK_3
unit phase section		process unit	T_3
/		parameter	PH_SetP
unit phase	W_T1	tag name	T3_PHset
condition	RC_W_T1	/	
weight parameter	W_T1_10	unit phase	T3_SPARGE
resource phase	TR_W_T1	condition	RC_T3_SP
process resource	WATER_TANK	weight parameter	T3_WP_50
process unit	W_TANK	resource phase	SPARGE_H_T3
process resource	TANK_1	process resource	WATER_TANK
process unit	T_1	process unit	W_TANK
/		process resource	TANK_3
unit phase	T1_MIX	process unit	T_3
condition	RC_MIX_T1	parameter	TEMPERATURE
weight parameter	T1_WP_150	tag name	T3_SP_Tsp
resource phase	MIXING_T1	/	
process resource	TANK_1	unit phase	T3_WJ
process unit	T_1	condition	RC_T3_WJ
/		weight parameter	T3_WP_70
unit phase	T3_TO_T4	resource phase	WATER_H_C_T3
condition	RC_T3_T4	process resource	TANK_3
weight parameter	T3_T4_10	process unit	T_3
resource phase	TR_T3_T4	parameter	TEMPERATURE
process resource	TANK_3	tag name	T3_WJ_Tsp
process unit	T_3		
process resource	TANK_4		
process unit	T_4		
parameter	TEMPERATURE		
tag name	T3_T4_Tsp		

Table 4.4 Real-Time control model for the dextrans process

4.4 Control Sequences and Modelling

To operate a plant, control at the individual item level (*e.g.* open a valve, enable an analogue control loop, *etc.*) is needed. For batch processes, these individual control actions are typically grouped together in a pre-defined order to form a control sequence. Control sequences are implemented in programmable logic controllers (PLCs) or DCS control systems. In this study, the proprietary ACCOS control system (APV Baker) is used to control the batch pilot plant. The equipment control logic is written in the form of control sequences in the PARACODE control language. A total of 98 control sequences have been developed which cover most of the possible operations of the pilot plant. An example control sequence for the sparge heating of the reactor (corresponding to the control phase T3_Sparge in Table 4.3) is shown in Table 4.5. The *sparge heating sequence* (Seq 78) is a top sequence which can also be started/stopped by the operator through an Operator Entry Code (OEC). It consists of three sub-sequences, including the *T3 level monitoring sequence* (Seq 203), the analogue *Control loop sequence* (Seq 103), and the *emergency control sequence* (Seq 127). The control hierarchy for the sparge heating was shown in Figure 3.12 in Chapter 3.

For off-line simulation, elementary control actions are modelled in the form of tasks in the gPROMS simulation environment, which describe basic control actions imposed on the plant model by its environment. For example, a task for opening a control valve is shown in Figure 4.6 in the gPROMS language. The task *Open_valve* sets a variable (*Control_signal*) value in the model (*On_Off_Valve*) to the value of 1.0 which corresponds to a control action to open the valve.

```

TASK Open_valve
  PARAMETER
    ConValve AS MODEL On_Off_Valve
  SCHEDULE
    RESET ConValve.Control_signal := 1.0; END
END

```

Figure 4.6 An elementary task for opening a control valve in the simulation model

These elementary tasks can then be combined, either in sequence or parallel, to form a composite task (equivalent to a control sequence) which describes a more complex operation, such as the sparge heating of the reactor. A task in the gPROMS language for the sparge heating control is given in Table 4.6.

<pre> .SEQ 78 / / SPARGE HEATING IN T3 / ***** / STARTED BY OEC. / INSQ 1.50 / T1 TO T3 INSQ 1.51 / T2 TO T3 INSQ 1.52 / T1 T2 TO T3 INSQ 1.55 / T3 TO T4 INSQ 1.59 / T3 TO T5 INSQ 1.69 / T3 RC INSQ 1.72 / T3 RC HE1 INSQ 1.75 / T3 RC WITH ADDITION INSQ 1.76 / PH CONTROL INSQ 1.77 / T3 ADDITIVES INSQ 1.80 / JACKET CONTROL INSQ 1.107 / PRESSURE CONTROL INSQ 1.153 / CLEAN T3 / EMFL T1.656 SEFL T1.313 CLFL T1.78 / S78A1, ADCI IT1-6, F1.57 IF F1.57, LT, 145.0, S78A MESS 1.41, 1.1 / T3 Temp too high. VARN 3 SEFL T1.78 S78A, IF F1.53, GT, 80.0, S78D MESS 1.11, 1.1 / Less than 80kG VARN 80 GNAL 1 GOTO S78SP S78D, IFIE BV1-2B, S78G2 MESS 1.81, 1.1 / BV1-2 not closed VARN 2 GNAL 1 GOTO S78SP S78G2, IFOD AV1-11, S78G3 MESS 1.75, 1.1 VARN 11 SEFL T1.78 / S78G3, IFOD AV1-12, S78G5 MESS 1.75, 1.1 VARN 12 SEFL T1.78 S78G5, IFOD AV1-7, S78G6 / Steam to HX MESS 1.75, 1.1 VARN 7 SEFL T1.78 S78G6, IFOD AV1-28, S78G7 / T3 inlet MESS 1.75, 1.1 VARN 28 SEFL T1.78 S78G7, IFOD ABV1-1, S78H / T3 outlet MESS 1.80, 1.1 VARN 1 SEFL T1.78 / S78H, IFFC T1.78, S78I MESS 1.1, 1.1 VARN 78 </pre>	<pre> CLFL T1.78 GNAL 1 ALSQ 78 / Check and restart GOTO S78A1 / S78I, EMRG AV1-5 / Steam sparge stop valve EMSQ 1.127 / Emergency sequence CHDE BV1-2A CHCK BV1-2B SEFL T1.27 / S82 to monitor pressure SEFL P1.128 / NOT COMPLETE SB SEFL T1.655 / RUNNING SB / IFSAC 1.203, S78K / T3 HL monitor STOT 1.203 / S78K, STSQ 1.103 / T3 sparge control / S78CK, ADCI IT1-6, F1.57 IFREG F1.57, GE, R1.13, S78T / Temp. in T3 > SP / ADCI IT1-8, F1.76 / Check T3 pressure IF F1.76, LE, 3.8, S78CKA / Too high MESS 1.46, 1.1 / T3 venting GNAL 1 ALSQ 1.78 / S78CKA, IF F1.53, LT, 130.0, S78CK MESS 1.50, 1.1 / Content in T3 > 130KG VARN 3 GNAL 1 ALSQ 1.78 GOTO S78CK / S78T, MESS 1.47, 1.1 / T3 temp to SP STMR 1.4, R1.14 RLTM 1.4 S78CK1, ADCI IT1-6, F1.57 IF F1.57, GT, 150.0, S78X1 IFTZ 1.4, S78X2 ADCI IT1-8, F1.76 / Check T3 pressure IF F1.76, LE, 3.8, S78CKB / Too high MESS 1.46, 1.1 / T3 venting GNAL 1 ALSQ 1.78 S78CKB, IF F1.53, LT, 130.0, S78CK1 MESS 1.50, 1.1 / Content in T3 > 130kG VARN 3 GNAL 1 ALSQ 1.78 GOTO S78CK1 S78X1, MESS 1.41, 1.1 / T3 temp. too high VARN 3 GNAL 1 / Alarm ALSQ 1.78 GOTO S78CK S78X2, MESS 1.90, 1.1 / Completed successfully VARN 78 CLFL P1.128 / Complete SB / S78Z, SPSQ 1.103 S78SP, STOP </pre>
--	--

Table 4.5 T3 sparge heating control sequence in PARACODE

```

TASK Start_Sparge_Heating

PARAMETER
    Plant AS MODEL Complete_Batch_Plant

SCHEDULE
SEQUENCE
    GET
        Plant.R_T3.T3_Sparge_Cont.Controller.Setpoint := T3_SP_Tsp ;
    END
    WHILE Plant.R_T3.T3_Sparge_Cont.Controller.Setpoint < 40.0 DO
        SEQUENCE
            CONTINUE FOR 10
            MESSAGE " ** Waiting for T3_SP_Tsp to be set (>=40.0) **"
            PAUSE "TIMER VIRTUAL"
            GET
                Plant.R_T3.T3_Sparge_Cont.Controller.Setpoint := T3_SP_Tsp ;
            END
        END # seq
    END #while
    RESET Plant.StatusT3_SP := 1.0 ;END
    RESET Plant.R_T3.T3_Unit.Process_Flag:=2.0; END
    RESET Plant.R_T3.T3_Unit.Control_Flag :=0.0 ; END
    RESET Plant.R_T3.T3_W_Jacket_Cont.T3_J.Control_Flag :=0.0 ; END
    Open_Steam_Stream_Valve (Steam_Valve IS Plant.R_T3.T3_Sparge_Cont.AV1_5);
    Enable_Sparge_GMC_Controller(Controller IS Plant.R_T3.T3_Sparge_Cont.Controller);
    RESET Plant.SP_Time :=900 ;END

REPLACE
    Plant.T3_Counter
WITH
    Plant.T3_Counter :=TIME-OLD(TIME) ;
END

WHILE Plant.T3_Counter < Plant.SP_Time DO
    SEQUENCE
        CONTINUE FOR 10
        SEND
            IT1_5_T3 := Plant.R_T3.T3_Unit.V3;
            IT1_6 := Plant.R_T3.T3_Unit.Tr;
        END
        PAUSE "TIMER VIRTUAL"
    END # Seq
END #While

SEQUENCE
    RESET Plant.R_T3.T3_Unit.Process_Flag:=1.0; END
    Disable_Sparge_GMC_Controller(Controller IS Plant.R_T3.T3_Sparge_Cont.Controller);
    Close_Steam_Stream_Valve (Steam_Valve IS Plant.R_T3.T3_Sparge_Cont.AV1_5);
    END

# Interface with SB
    RESET Plant.PhaseTag_SP := 0.0 ;END
    RESET Plant.StatusT3_SP := 0.0 ;END
    SEND
        T3_SPARGE :=0 ; # Finish task
        RC_T3_SP := 0 ; # Complete successful
        IT1_5_T3 := Plant.R_T3.T3_Unit.V3 ;
    END
END

END

```

Table 4.6 Simulated sparge heating control sequence in gPROMS

The form of the control sequence models in simulation differs to some degree from the actual control sequences implemented in the PARACODE. For example, "ENGE AV1-5" in the PARACODE opens the control valve AV1-5 which is equivalent to the simulated task shown in Figure 4.6. The inheritance and hierarchical model decomposition techniques used in the gPROMS simulation environment are not supported by the ACCOS control system. However, the basic control logic and the actions imposed on the plant equipment items are identical in the simulation models and the actual control code.

4.5 Model of Plant Dynamics

The plant model is developed using the gPROMS simulator (Barton and Pantelides, 1991). The package supports model development in a combined modular and hierarchical manner, starting from the definition of simple elementary models (*e.g.* a control valve). Simple equipment component models are combined to form models of more complex plant units (*e.g.* the aggregated unit of TANK-3 shown in Figure 4.1). The following assumptions are made:

- perfect mixing in each vessel with either agitation or recirculation at a high flow rate;
- there is no change of phase during reactions or material transfers;
- no energy is gained or lost during transfers.

In all the models discussed below, definitions of variables and model parameters are given in the nomenclature.

Model of raw material tanks. The raw material tanks (TANK-1/2) are used to prepare reactants which are then transferred to the reaction vessel. For this reason, a steady state model is sufficient to represent the core properties, such as the initial concentration of a reactant. The tank total hold-up and the energy balance dynamic models are:

$$\frac{dV_t}{dt} = \sum_{i=1}^m F_{in}^i - F_{out} \quad (4.1)$$

$$C_s = \frac{S_d}{S_d + S_w} \rho_t \quad (4.2)$$

$$\frac{d(V_t T_t)}{dt} = \frac{1}{\rho_t C_{pt}} \sum_{i=1}^m (\rho^i C_p^i F_{in}^i T_{in}^i - A_t U_t (T_t - T_{rm})) - F_{out} T_t \quad (4.3a)$$

Model of CIP tank. The CIP tank is used to store the cleaning detergent and to prepare detergent with required properties. The CIP tank model is similar to that of the raw material tanks, with Eq (4.2) replaced with its dynamic counterpart so as to model the detergent concentration in the tank during the detergent dosing control.

$$\frac{d(V_d C_d)}{dt} = \sum_{i=1}^n F_{in}^i C_{in}^i - F_{out} C_d \quad (4.4)$$

Model of reactor and jacket. There are four ways to control the reactor temperature (see section 5.3.1). Depending on the method used for the temperature control and type of reactions performed in the reactor, different models must be used. For the dextrin process, sparge heating is used for fast gelatinization of the starch slurry, and water jacket control is used for the starch degradation phase. The reaction kinetics is expressed by the Michaelis-Menten equation (Fujii *et al.*, 1981).

During sparge heating control:

$$\frac{dV_r}{dt} = 1/\rho W_{ss} \quad (4.5)$$

$$\frac{dP_{dex}}{dt} = V_{max} C_s / (K_m + C_s) \quad (4.6)$$

$$V_r \frac{dT_r}{dt} = -\alpha_1 (T_r - T_j) - \alpha_2 (T_r - T_{rm}) + \alpha_3 (H_{sp} - C_p T_r) W_{ss} + \alpha_3 Q \quad (4.7a)$$

$$\frac{dT_j}{dt} = 1/\tau_{sp} (T_r - T_j) \quad (4.8a)$$

Under water jacket control, Eq (4.7a) and Eq (4.8a) are replaced by (4.7b) and (4.8b), respectively.

$$V_r \frac{dT_r}{dt} = -\alpha_4 (T_r - T_j) - \alpha_2 (T_r - T_{rm}) + \alpha_3 Q \quad (4.7b)$$

$$V_{jr} \frac{dT_j}{dt} = \beta_1 (T_r - T_j) + F_j (T_{jin} - T_j) \quad (4.8b)$$

Where $\alpha_1 = A_{20} U_{sp} / \rho C_p$, $\alpha_2 = A_{10} U_{10} / \rho C_p$, $\alpha_3 = 1 / \rho C_p$, $\alpha_4 = A U_w / \rho C_p$, $\beta_1 = A U_w / \rho_j C_{pj}$

For the cyclodextrins process, only the water jacket control is used for the temperature control of the reactor. Due to the lack of appropriate models, the kinetic model as for the dextrans process is used. This may of course lead to errors in the model predictions. Nevertheless, the reason for considering this process is to evaluate the interactions between various processes at the production planning and supervisory control levels (in simulation mode only).

Jacket recirculation flow: Experimental results reveal that the jacket recirculation flow has a strong influence on the jacket dynamics. To avoid simultaneous heating and cooling, control valves CV1-4/CV1-7 are configured in a split-range, such that a controller output of 0 to 50% switches off CV1-4 and regulates CV1-7 between 100 to 0% open. Similarly, a controller output in the range 50 to 100% switches off CV1-7 and regulates CV1-4 in the range 0 to 100%. Figure 4.7 shows that three possible situations occur in the jacket recirculation route including: i) *full heating*. The jacket recirculation route is a "closed system" in which no water enters or exits. The cooling water control valve (CV1-7) and non-return valve (NRV1-1) are fully closed and only the steam valve CV1-4 is in operation; ii) *mixing*. The water in the recirculation route is bled through NRV1-1 and makeup cooling water enters through CV1-7. iii) *full cooling*. The water from the jacket is fully returned to the coolant supply system through NRV1-1. The jacket flow rate model as a function of the controller output was determined experimentally.

$$F_j = \begin{cases} F_{rc} & \text{Constant,} & 40 < u_c \leq 100 \\ k_j(80 - 2u_c) + F_{rc} & & 12.5 < u_c \leq 40 \\ F_{cw}(u_c) & & 0 \leq u_c \leq 12.5 \end{cases} \quad (4.9)$$

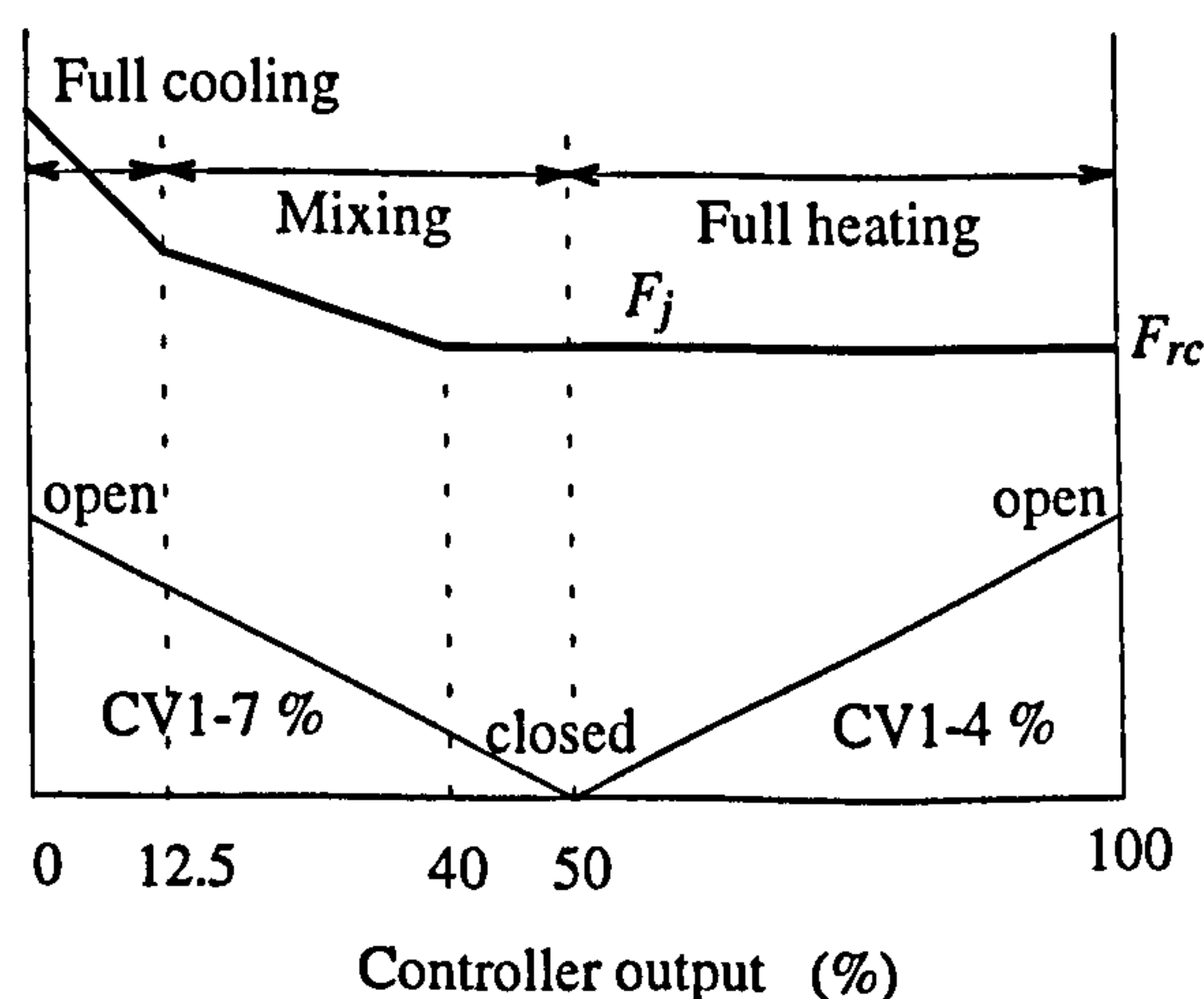


Figure 4.7 Jacket recirculating flow characteristics

Heat exchanger (HE2): HE2 is a steam counterflow plate heat exchanger modelled using the arithmetic mean temperature difference equation.

$$T_{cin} = T_j + (F_{cw}/F_j) \cdot (T_{cw} - T_j) \quad (4.10)$$

$$T_{jin} = T_{cin} + \varphi_{he} (T_{hin} - T_{cin}) \quad (4.11)$$

$$\varphi_{he} = \frac{2C_{ps}W_s}{C_{ps}W_s + 2C_{ps}K_{he}F_jW_s + \rho_j C_{pj}F_j} \quad (4.12)$$

Model of product tanks. The product tanks (TANK-4/5) are used to store the final products. They may also be used to perform certain reactions. Temperature control in these tanks is achieved by recirculating the contents through an external heat exchanger (see section 5.3.1). Again, the same reaction kinetic models as in Eq (4.6) are used. The material balance is the same as for the raw material tanks. The energy balance in Eq (4.3a) is replaced by Eq (4.3b) to take into account the heat of the reaction in the vessels.

$$\frac{d(V_p T_p)}{dt} = \frac{1}{\rho_i C_{pt}} \sum_{i=1}^k (\rho^i C_p^i F_{in}^i T_{in}^i - A_p U_p (T_p - T_{rm}) + Q_{rp}) - F_{out} T_p \quad (4.3b)$$

Control valves: All analogue control valves in the plant are equal percentage. However, experimental data show that the water control valves have linear flow characteristics, while the steam control valves have nonlinear characteristics. The dynamics of the control valves is much faster than that of the reactor and the jacket. Therefore, only steady-state models are considered. Models of CV1-4, CV1-7 and CV1-3 are developed based on experimental data and are given by Eq (4.13), Eq (4.14) and Eq (4.15), respectively.

$$W_s = k^{he} \alpha^{(x^s/100-1)} - k^{he} \alpha^{(-1)} \quad (4.13)$$

$$F_{cw} = \begin{cases} k_1^{cw} x^{cw} & 0 \leq x^{cw} \leq 40 \\ k_2^{cw} (x^{cw} - 40) + 40k_1^{cw} & 40 < x^{cw} \leq 100 \end{cases} \quad (4.14)$$

$$W_{ss} = \begin{cases} k_1^{sp} x^{sp} & 0 \leq x^{sp} \leq 15 \\ k_2^{sp} (x^{sp} - 15) + 15k_1^{sp} & 15 < x^{sp} \leq 50 \\ k_3^{sp} (x^{sp} - 50) + 35k_2^{sp} + 15k_1^{sp} & 50 < x^{sp} \leq 100 \end{cases} \quad (4.15)$$

Model of control elements. Modelling of on/off control valves is necessary since some operations (e.g. CIP) are achieved by operating these valves in a defined sequence. The dynamics of the transmitters (flow, level and temperature) is very fast and so dynamics may be neglected.

4.5.1 Model Validation and Parameter Estimation

Model parameters were fitted to experimental data. In particular, the reactor heat transfer coefficient, heat exchanger (HE2) coefficient and control valve constants were estimated.

Reactor heat transfer coefficient U_{I0} was estimated by linear regression of W_{ss} measured against $(T_r - T_{rm})/(H_{sp} - C_p T_r)$ at different operating points according to Eq (4.7a). The slope of this regression line gives the least square estimation of $A_{I0}U_{I0}$. Similarly, heat transfer coefficient U_w was determined from linear regression of $(T_j - T_r)$ and $(T_r - T_{rm})$. From Eq (4.7b), the slope of the straight line is $A_{I0}U_{I0}/AU_w$.

Reactor time constant during the sparge heating τ_{sp} was obtained from a step change response of the jacket temperature based on Eq (4.8a).

All the control valve constants were estimated using a least square estimation technique based on the measurements of flow rates at different valve positions.

The following experiment was conducted to estimate K_{he} . i) The non-return valve NRV1-8 was blocked so that the recirculating flow will be forced back to the supply system. Therefore, temperature T_{cin} cold stream inlet to HE2 remains constant; ii) A 4-20 mA signal generator is used to drive CV1-7 while CV1-4 is driven by a controller; iii) position of CV1-7 was fixed while varying the position of CV1-4 to obtain a set T_{jin} . This will give a set of corresponding values of ϕ_{he} according to Eq (4.11). From the Eq (4.12), $1/\phi_{he}$ is a linear function of $1/W_s$. The slope of the fitted regression line gives $\rho_j C_{pj} F / 2C_{ps}$ and the intercept of the line on the $1/\phi_{he}$ axis yields $(0.5 + F_j K_{he})$. Thus the estimation of the HE2 heat transfer area and coefficient ($K_{he} = \rho_j C_{pj} / U_{he} A_{he}$) are avoided.

The values of the estimated and physical parameters of the system are given in the nomenclature.

4.6 Summary

In this Chapter, modular and hierarchical models of the batch pilot plant were developed at various levels of detail using appropriate modelling tools. In order to achieve both on-line and off-line control and various operations support functions, the following major modelling components were considered:

- *Equipment models.* The equipment was modelled in a hierarchical way in which aggregated unit models were used for the high level production planning and supervisory control functions, and a detailed item level model is only considered for the lower level controls (*e.g.* for validating/testing control sequences). The mapping between the equipment and production tasks at the planning and scheduling level is determined by the equipment suitability.
- *Process models.* Processes were represented as State Task Networks (Kondili *et al.*, 1993). Unit State Task Networks (Crooks, 1992) were used to represent states of equipment units which permit a desired sequence of operations to be forced on a unit.
- *Control recipe models.* More detailed information regarding the operation of the plant was modelled as master procedures and control phases. Mappings between control phase parameters and external control system tags, devices, *etc.* were also specified in the model that allows the information (data, decision, *etc.*) to be passed between the supervisory system and external controllers.
- *Plant dynamics.* Knowledge regarding the physico-chemical behaviour of the system was modelled in the plant dynamic models. Inclusion of the plant dynamic model allows various operation support actions to be tested by simulations before implemented on the plant. The dynamic model is also required for developing model based controllers.

Model coordination and mapping between various levels of the hierarchy ensures the integrity of control decisions at a local level. The mapping (those related to the dextrin process and CIP operation only) between the STN tasks, control phases and control sequences (top level) is summarised in Table 4.7

STN Model (MILP scheduling Level)	Control Phase (Supervisory Level)	Control Sequence (ACCOS 30)
Make_SI_A	Tr_W_T1/2	Seq 65/66 - Fill T1/2 with water
Reaction_A	Tr_T1/2_T3	Seq 50/51 -Transfer T1/2 to T3
	pH_Control	Seq 76 -pH control
	Sparge_H_T3	Seq 78 - Sparge heating
	Water_H_C_T3	Seq 80 -Jacket water H/C
Discharge_A	Tr_T3_T4/5	Seq 55/56 - Fill T4/5 from T3
	Emp_T4/5_P	Seq 67/68 - Product discharge
CIP_service	Service_CIP	Seq 157 - Service detergent tank
	DET_Dosing	Seq 161 - Detergent dosing
	CIP_RC_HE3	Seq 160 - CIP heating
Cleaning	CL_T1-5	Seq 151 to 155 - Cleaning T1 to T5

Table 4.7 Mapping of STN tasks/control phases/control sequences for the dextrin process

The models developed here will be served as a basis for the study of each level of control and operations support functions detailed in the remaining chapters.

Chapter 5

Partitioning Space Approach for Nonlinear Process Control

The need for tight control to ensure high performance and consistent product quality is well appreciated. However, the existence of process nonlinearities and frequent changes in operating conditions over a wide range present major obstacles for tight control. To ensure stability for a wide range of operating conditions, linear controllers or controllers based on locally linearized models must be significantly detuned or retuned for different operating conditions. However, in flexible multipurpose plants, the processing conditions (batch size, raw materials, products, *etc.*) might be nondeterministic, and thus controller tuning parameters cannot be set precisely in advance. Retuning linear controllers in response to changes in processing conditions is not only expensive but may also be difficult.

During the last decade, significant progress has been made in the control of non-linear systems and processes with a wide range of operating conditions. For highly nonlinear processes, controller design strategies based directly on nonlinear process models can be expected to provide significantly improved performance (Henson and Seborg, 1990). Globally Linearizing Control (GLC, Kravaris and Chung, 1987) is one of such design approaches that are based directly on a nonlinear process model or an exact linearization of a nonlinear model. Unlike Jacobian linearization, which linearizes the model only at the nominal operating point, these methods employ nonlinear transformations that provide exact linearization of the model in a neighbourhood of the nominal point. Generic Model Control (GMC, Lee and Sullivan, 1989) can be viewed as a special case of the GLC approach for systems of relative degree of one (Henson and Seborg, 1991), and their control can be made identical by a proper selection of the reference system parameters and the tuning parameters of the external linear controller (Barolo, 1994). However, no experimental verification has appeared in the literature that confirms this

equivalence. In this study, no further theoretical distinctions will be made between the GMC and GLC methods or approaches based on exact linearization. The GMC method is only considered for the systems with relative order one. The following discussion is concentrated on the GLC approach, but is also valid for most control approaches that based on exact linearization.

5.1 Control Based on Exact Input-output Linearization

A number of applications have been reported that are based on the GLC approach, either implicitly or explicitly (Henson and Seborg, 1991; Dore, 1993), but very few experimental studies have been presented. There are a number of limitations associated with the GLC approach that must be overcome to make the approach more applicable in practice.

1. Robustness. Exact linearization techniques require a perfect model of the process in order to design a control law that exactly cancels all the nonlinearities between the input and output. If the process model is imperfect, these nonlinearities are not cancelled and the resulting input-output map is not linear. The effect of the model mismatch on the controller performance is difficult to assess quantitatively, and the resulting control performance is often case-dependent. The general rule is to avoid using exact linearization when the quality of the model is poor. This restricts application of these methods in a large number of practical cases even when the system is partially well known.
2. Disturbance decoupling. For a nonlinear system with disturbance, the output of the system can be completely decoupled from the disturbance only when the so called disturbance matching condition is satisfied (Isidori, 1989). However, this condition is rarely satisfied in practical systems. Although several approaches have been developed for processes with disturbances that do not satisfy the matching condition, the resulting control law often contains derivatives of the disturbances (Daoutidis and Kravaris, 1989), or high gain feedback is required (Marino *et al.*, 1989). These approaches may not be suitable for control of processes which have high frequency noise or unmodelled dynamics.
3. Constraints on the state variable. Process input, output and state variables are often subject to constraints. Although constraints on the input or input movement can be explicitly specified in the GLC control law (Soroush and Kravaris, 1992), constraints on the state variables are difficult to handle using the GLC approach. The effect of these constraints

when active in the feedback transformation and on the resulting controller performance is not clear.

4. *Higher relative orders.* The complexity of the controller resulting from the exact linearization approaches grows exponentially with the relative order of the system. For a system with high relative order (*e.g.* greater than two), the resulting control law may be very difficult to implement.
5. *Internal dynamics.* When the order of the system is greater than the relative degree of the system, the input-output linearization does not lead to input-state linearization (Slotine and Li, 1991). The GLC approach cannot guarantee the stability of the internal dynamics of the system.

In conclusion, control approaches based on exact input-output linearization have been successfully applied to a limited number of practical control problems. However, the approaches have been restricted only to a class of systems for which models are well defined and satisfy a number of conditions, such as disturbance matching conditions, stable internal dynamics, an low relative order. Obtaining a good model for an entire process system is difficult. On the other hand, portions of the system are often easy to model. In the following section, a partitioning space approach is presented for nonlinear processes that is aimed at making the best use of the well known part of the system model for the nonlinear controller design and resolving some of the problems. A theoretical analysis is described first. The approach is then demonstrated by two examples including a simple level control for two noninteracting tanks, and the temperature control of a pilot scale multipurpose batch reactor.

5.2 Partitioning Space Approach for Nonlinear Process Control

5.2.1 Characteristics of Cascade Control Structure

The motivations for cascade control structure have been well understood in linear control theory (Stephanopoulos, 1984). The following characteristics have led to the extensive use of cascade control in chemical processes:

- *Local rejection of disturbances.* Disturbances arising within the secondary loop are rejected by the secondary controller before they can affect the system output.

- *Suppression of uncertainty/nonlinearities.* Uncertainty and nonlinearities are suppressed in the secondary loop before they can affect the output of the primary control loop. A classical example is flow control in which the secondary loop is used to compensate for flow rate changes. The uncertainty on the valve inlet and outlet pressures, effect of wear on valve, and valve leakage are suppressed by the secondary loop.

However, to effectively decouple the secondary loop from the primary requires the closed loop bandwidth of the secondary to be much greater than that of the primary. In most applications, only conventional PID type controllers are used in a cascaded control structure and no analytical tools are available for the selection of the secondary loop output. The selection procedure is mainly based on individual experiences at the present.

In conclusion, cascade structures offer more robust and better disturbance rejection properties than direct feedback. Therefore, it is of interest to look at how this approach can be used with nonlinear controllers.

5.2.2 Partitioning Space

Consider the following SISO control *affine* nonlinear system:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u + \gamma(x)d \\ y &= h(x)\end{aligned}\tag{5.1}$$

where x is an n -dimensional state vector, u is a scalar input, d is a scalar disturbance, y is a scalar output, $f(x)$, $g(x)$ and $\gamma(x)$ are vector functions, and $h(x)$ is a scalar function. It is assumed that $f(x)$, $g(x)$, $\gamma(x)$ and $h(x)$ are of class C^∞ (*i.e.* have continuous derivatives of all orders). The above system can always be transformed into the following form:

$$\begin{aligned}\dot{x}_I &= f_I(x) + g_I(x)u + \gamma_I(x)d \\ \dot{x}_{II} &= f_{II}(x) + g_{II}(x)u + \gamma_{II}(x)d \\ y &= h(x)\end{aligned}\tag{5.2}$$

where $x=[x_I \ x_{II}]^T$, $f(x)=[f_I(x) \ f_{II}(x)]^T$, $g(x)=[g_I(x) \ g_{II}(x)]^T$, $\gamma(x)=[\gamma_I(x) \ \gamma_{II}(x)]^T$. The partition is chosen such that the following conditions are satisfied:

$$\frac{\partial y}{\partial x_i} = 0 \quad \forall x_i \in x_I, \quad \frac{\partial y}{\partial x_j} \neq 0 \quad \forall x_j \in x_{II} \quad (5.c1)$$

If either x_I or x_{II} are empty, then a partition does not exist.

The SISO system in Eq (5.2) is said (Isidori, 1989) to have relative degree r in a region Ω if, $\forall x \in \Omega$

$$\begin{aligned} L_g L_f^i h(x) &= 0 \quad \forall i = 0, \dots, r-2 \\ L_g L_f^{r-1} h(x) &\neq 0 \end{aligned}$$

From this definition, the following new corollary is obtained.

Corollary 1: The relative degree of the system given by Eq (5.2) is greater than one if

$$g_f(x) = 0, \quad \forall g_f(x) \in g_{II}(x). \quad (5.c2)$$

Proof:

$$L_g h(x) = \frac{\partial h(x)}{\partial x} g(x) = \frac{\partial h(x)}{\partial x_I} g_I(x) + \frac{\partial h(x)}{\partial x_{II}} g_{II}(x)$$

As $\frac{\partial h(x)}{\partial x_I} = 0$, $L_g h(x) = 0$ if $g_f(x) = 0$, $\forall g_f(x) \in g_{II}(x)$.

Condition (5.c2) provides a simple sufficient condition to identify systems which have a relative order greater than one without explicit calculation of each order of Lie derivatives.

In the rest of the work presented, the partitioning space approach is only considered for systems in which $g_{II}(x) = 0$, since the design of controllers for systems of order one is rather simple. The system in Eq (5.2) is, therefore, simplified as follows:

$$\begin{aligned} \dot{x}_I &= f_I(x) + g_I(x)u + \gamma_I(x)d \\ \dot{x}_{II} &= f_{II}(x) + \gamma_{II}(x)d \\ y &= h(x) \end{aligned} \quad (5.3)$$

Assume the system (5.3) has a relative degree q with respect to the disturbance d . A necessary and sufficient condition for the solution of the disturbance decoupling problem is (Isidori, 1989):

- $q > r$ if the disturbance cannot be measured. This is often called the disturbance matching condition (Henson and Seborg, 1991)
- $q \geq r$ if the disturbance can be measured.

From this definition, the following new corollary is obtained:

Corollary 2: The disturbance matching condition is not satisfied for a system described by Eq (5.3) if

$$\frac{\partial h(x)}{\partial x_{II}} \gamma_{II} \neq 0 \quad (5.c3)$$

Condition (5.c3) provides a simple sufficient condition to identify partitioned nonlinear systems which do not satisfy the disturbance matching condition. For a system satisfying condition (5.c3), the disturbance cannot be decoupled using the input-output feedback linearization approach even when the disturbance can be measured. The following remarks can be made for systems described in Eq (5.3):

1. The nonlinear system is not controllable (the relative degree of the system is ∞) if $g_I(x)=0$.

Proof:

$$L_g h(x) = \frac{\partial h(x)}{\partial x} g(x) = \left[\frac{\partial h(x)}{\partial x_I} \quad \frac{\partial h(x)}{\partial x_{II}} \right] [g_I(x) \quad g_{II}(x)]^T = \frac{\partial h(x)}{\partial x_I} g_I(x) + \frac{\partial h(x)}{\partial x_{II}} g_{II}(x) = 0$$

For any integer value of r

$$L_g L_f^r h(x) = \frac{\partial (L_f^{r-1} h(x))}{\partial x} g(x) = 0$$

This proves that the system has a relative degree of ∞ which simply implies that the control input u has no effect on the system output y . Such a system is rare in practice.

2. The nonlinear system is not controllable if $\partial f_j(x)/\partial x_i = 0$, $\forall x_i \in x_I$, $\forall f_j(x) \in f_{II}(x)$.

The second condition can be proved intuitively. As $g_{\Pi}(x)=0$, the control input u has no direct affect on state variables x_{Π} , and so neither on the output y . If $\partial f_j(x)/\partial x_i = 0, \forall x_i \in x_I, \forall f_j(x) \in f_{\Pi}(x)$ is satisfied, all the state variables x_I are independent of variables x_{Π} . Therefore, the control input u will not directly or indirectly affect the system output y . Such a system is not controllable.

The above general remarks and corollaries provide a simple method for screening systems suitable for partitioning.

5.2.2.1 Nonlinear System Partitioning

For the system in Eq (5.3), if $\exists x_c, x_c \in x_I$ such that

$$\frac{\partial f_j(x)}{\partial x_c} \neq 0, \text{ for at least one } j \text{ such that } f_j(x) \in f_{\Pi}(x) \quad (5.c4)$$

$$\frac{\partial h}{\partial x_{\Pi}} \left(\frac{\partial f_{\Pi}(x)}{\partial x_c} + \frac{\partial \gamma_{\Pi}(x)}{\partial x_c} \right) \neq 0 \quad (5.c5)$$

Then, the system can be partitioned into two subsystems SysI and SysII, as in Eq (5.4)

$$\begin{cases} \text{SysI} \begin{cases} \dot{x}_I = f_I(x) + g_I(x)u + \gamma_I(x)d \\ y_c = x_c \end{cases} \\ \text{SysII} \begin{cases} \dot{x}_{\Pi} = f_{\Pi}(x, x_c) + \gamma_{\Pi}(x)d \\ y = h(x) \end{cases} \end{cases} \quad (5.4)$$

The output y is not directly dependent on state variables x_I in SysI. However, y is a function of all state variables x_{Π} in SysII. A state variable x_c in SysI is selected to serve as a new output of that subsystem. The selection of x_c is arbitrary, as far as it satisfies the conditions of Eq (5.c4) and (5.c5). However, in order to implement the control law, x_c should be a physically meaningful and measurable process variable, such as temperature or flow rate. The x_c should also be selected in such a way that the closed loop response of the u - x_c loop is faster than that of the x_c - y loop. The partitioning space gives an extra dimension for control structure selection. A schematic representation of the partitioning of nonlinear systems is shown in Figure 5.1.

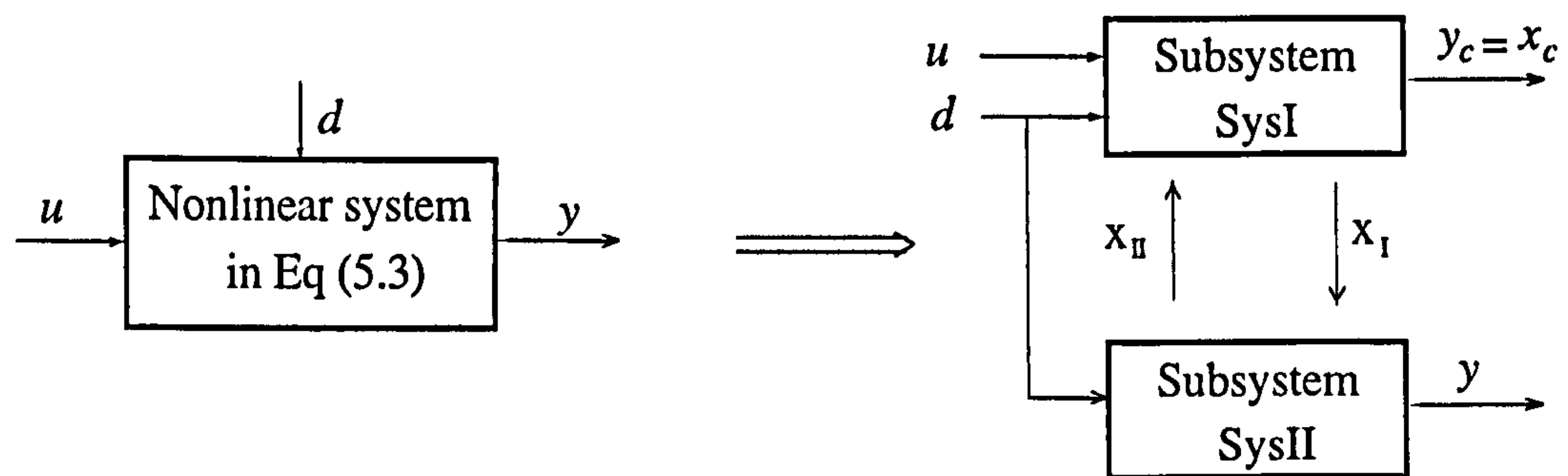


Figure 5.1 Partitioning of nonlinear systems

Corollary 3: For the system described by Eq (5.4), the disturbance d can be decoupled from SysII if it can be measured.

Furthermore, if $\frac{\partial h(x)}{\partial x_{II}} \gamma_{II} = 0$, the disturbance d can always be decoupled from the output in subsystem SysII even if d cannot be measured. By partitioning the nonlinear system into two subsystems, the relative degree of each subsystem will be less than that of the original nonlinear system. It can also be proved that the relative degree of the subsystem SysII, with respect to the intermediate control input x_c , is one. In SysI the relative degree of the system with respect to the intermediate output and control input u is reduced by one. As the growth in controller complexity is exponential with respect to the relative degree of the system, the reduction of the relative degree leads to a simpler controller for each subsystem.

5.2.3 Control System Design Based on Partitioned System

For the partitioned nonlinear system described in Eq (5.4), a cascade control structure can be formed by selecting x_c (an output of SysI) as a new intermediate control input for SysII. The outer loop ($x_c - y$) tracks the output y using the intermediate input x_c and the inner loop ($u - x_c$) achieves servo control of x_c using the original control input u . Unlike the exact input-output linearization approach that linearizes the nonlinear system in the whole space, the partitioning-space approach allows the design of a controller for each subsystem separately. Various design approaches can then be applied to each subsystem, depending on the confidence in the process model and the characteristics of each subsystem.

For example, consider the GLC approach for design of controllers for each subsystem. Assuming that the disturbance d can be measured in the subsystem SysII and SysII can be transformed into the control affine form:

$$\begin{aligned}\dot{x}_{II} &= f'(x) + g'(x)x_c + \gamma_{II}(x)d \\ y &= h(x)\end{aligned}$$

the GLC procedure developed by Soroush and Kravaris can be applied. Under the following state feedback:

$$x_c = x_c^* = \frac{v_2 - h(x) - \beta L_f' h(x)}{\beta L_g' h(x)} - \frac{L_{\gamma_{II}} h(x)}{L_g' h(x)} d$$

the subsystem SysII can be transformed into a linear form:

$$\beta \frac{dy}{dt} + y = v_2$$

The linear controller design techniques can be used for the transformed linear system. This results in a control law which calculates the desired manipulated action x_c^* in order to drive the output y along a defined trajectory. However, if the nonlinear system cannot be transformed into a control affine form, the input-output linearization of general nonlinear processes (Henson and Seborg, 1990) can be considered.

The subsystem SysI is in the control affine form. Therefore, the GLC procedure can be directly applied. The secondary GLC controller performs servo control to drive x_c as close and as fast to the desired trajectory x_c^* as possible. This results in the cascade control structure shown in Figure 5.2 where Ψ_1 and Ψ_2 are state feedback transformations for SysI and SysII, respectively. Figure 5.2 provides a general cascade control structure with either linear or nonlinear controllers. The conventional cascade structure with linear controllers can be viewed as a special case by choosing both state-feedback transformations as unity ($\Psi_1=1$, $\Psi_2=1$). Nonlinear controllers derived from various design techniques may also be used for control of each subsystem.

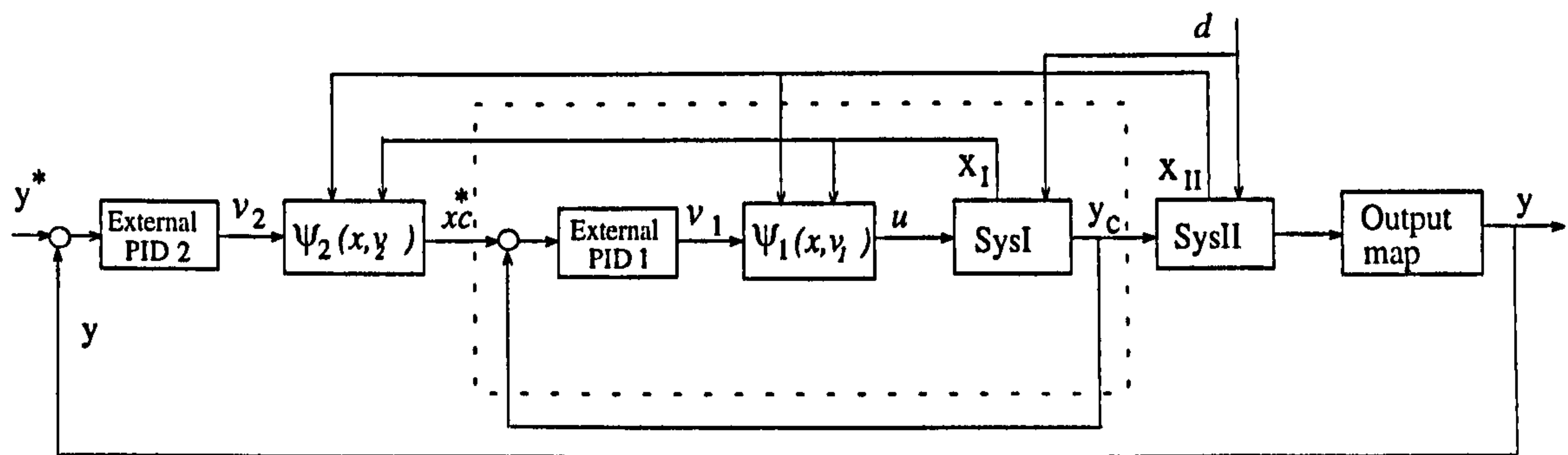


Figure 5.2 A cascade control structure for nonlinear process control

By partitioning the nonlinear control system and using the cascade control structure, a more flexible control system can be obtained in the sense that various control techniques are accommodated for each subsystem. This offers robustness and flexibility over approaches that are based directly on exact input-output linearization.

5.2.4 Handling Constraints

Constraints are often present in processes on controlled, manipulated and state variables due to the physical limitations of the equipment, safety and environmental regulations, economic considerations and product specifications. The ability to handle constraints is essential for any industrial control algorithm implemented on real processes. Garcia and Prett (1987) classified the constraints into two types according to their properties:

- *Hard constraints.* No dynamic violations of the bounds are allowed at any time;
- *Soft constraints.* Violations of the bounds are temporarily permitted in order to satisfy other heavily weighted criteria.

Brown *et al.* (1990) placed constraints into three categories:

- Input constraints

$$u_L \leq u \leq u_U$$
- Input movement constraints

$$\Delta u_L \leq u(t + \Delta t) - u(t) \leq \Delta u_U$$
- Functional constraints

$$C_L \leq C(y, x, u, t) \leq C_U$$

where C , C_L , and C_U are vectors. Each element of C is generally a nonlinear function of y , x , u and t . These types of constraints are usually derived from economic considerations.

As a further extension of the GMC control approach, Brown *et al.* (1990) also proposed a control algorithm that handles the functional constraints involving state and/or output variables. Slack variables which define the deviation from the GMC reference profiles are introduced to the GMC control law for both the control variables and constraint variables. By selecting the weighting factors on these slack variables and defining a control objective function in the controller, the desired compromise between constraint violation and setpoint tracking can be obtained. The solution of the problem is formulated as a single-step, nonlinear constrained optimization. The input and input movement constraints are often handled by specifying the upper and lower bounds explicitly in the control algorithms.

However, for the GLC approach, the key idea is to transform the nonlinear system into a linear input-output map by state feedback transformation. The constraints on the state variables are difficult to incorporate into the GLC controller algorithms. No work is reported in which the GLC approach is used for systems with constraints on the state variables.

In the partitioning space approach, a constraint on a single state variable may be dealt with by selecting the constrained variable as an intermediate control input, provided the conditions in Eq (5.c4) and Eq (5.c5) are satisfied. Therefore, the constraint on the state variable is transformed into an input constraint. The following equation should be specified in the controller algorithm

$$x_c^{\min} \leq x_c \leq x_c^{\max}$$

As a result of this constraint, the output of the master controller is limited to within $x_c^{\min}$ and $x_c^{\max}$. With respect to the unconstrained case, this may lead to a slower inner loop tracking response and some degradation of control performance. Carrying out a general quantitative assessment of the effect of constraints on the state transformation and controller performance is a difficult task and no further theoretical analysis is presented here. The effect of constraints on controller performance is however demonstrated in the temperature control of the batch reactor example in which the jacket inlet temperature is limited between the cooling water temperature T_c and the upper safety limit $T_j^{\max}$.

5.2.5 Design Procedure

The partitioning space approach solves the nonlinear control problem in two steps. Firstly, the original nonlinear system is translated into two subsystems. Secondly, a controller is designed separately for each subsystem using available techniques. This results in a cascade control system structure in which the outer loop achieves the output tracking control using the new intermediate input x_c , and the inner loop achieves the servo control of x_c using the original control input u . A heuristic design procedure is summarised in Figure 5.3.

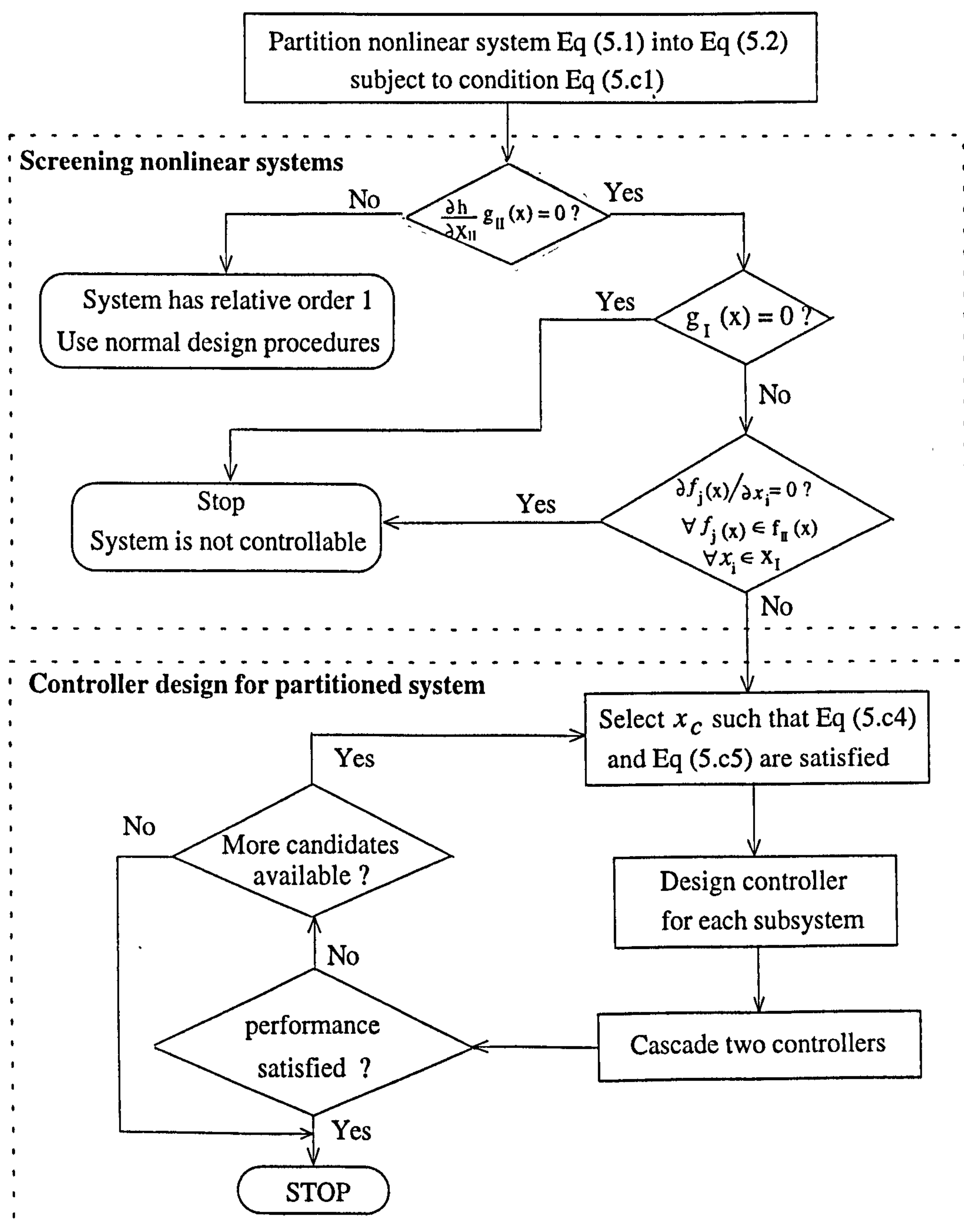


Figure 5.3 Partitioning space design procedure

From Figure 5.3 it may be also observed:

- i) If a variable x_c does not exist in the subsystem x_I which satisfies Eq (5.c4), this means that

$$\frac{\partial f_j(x)}{\partial x_i} = 0 \quad \text{for } \forall f_j(x) \in f_{II}(x) \text{ and } \forall x_i \in x_I$$

This system is not controllable.

- ii) If there are a number of state variables in the subsystem x_I that satisfy Eq (5.c4) and Eq (5.c5), a state variable with fast dynamics should be selected (*i.e.* based on knowledge of the system or simulations) as x_c to guarantee fast inner loop servo response. An exhaustive search procedure might be used for determining the best possible candidate.
- iii) The performance criteria can be set according to the applications and may be for example the speed of response or tracking performance. Model based control techniques are normally considered only if conventional PID controllers fail to achieve a control objective. Thus the performance of PID controllers is a benchmark for comparison.
- iv) If the model of the system is well defined for both subsystems, model based control approaches should be considered in the design of each subsystem controller. If only SysII is well defined and there is uncertainty in the model of SysI, model based approaches should be considered in the design of primary loop controller. However for SysI, uncertainty will cause an error in the computation of the control input u . In this case, there is no guarantee that the model based controllers will perform better than PID controllers.

5.2.6 Illustrative Example

Consider the level control of two noninteracting tanks in series as shown in Figure 5.4. The objective is to control the level (x_2) in tank T2 by manipulating the inlet flow (u) of tank T1. The T1 outlet flow is assumed to be proportional to the level (x_1) in T1, while the T2 outlet flow is assumed to be proportional to the square root of x_2 . The system can be represented as follows:

$$\begin{aligned} \dot{x}_1 &= -\alpha\gamma_1 x_1 + \alpha u \\ \dot{x}_2 &= \beta\gamma_1 x_1 - \beta\gamma_2 \sqrt{x_2} \\ y &= x_2 \end{aligned} \tag{5.5}$$

where y is the system output, and $\alpha, \beta, \gamma_1, \gamma_2$ are constant parameters determined by the cross sections of the tanks and outlet pipes.

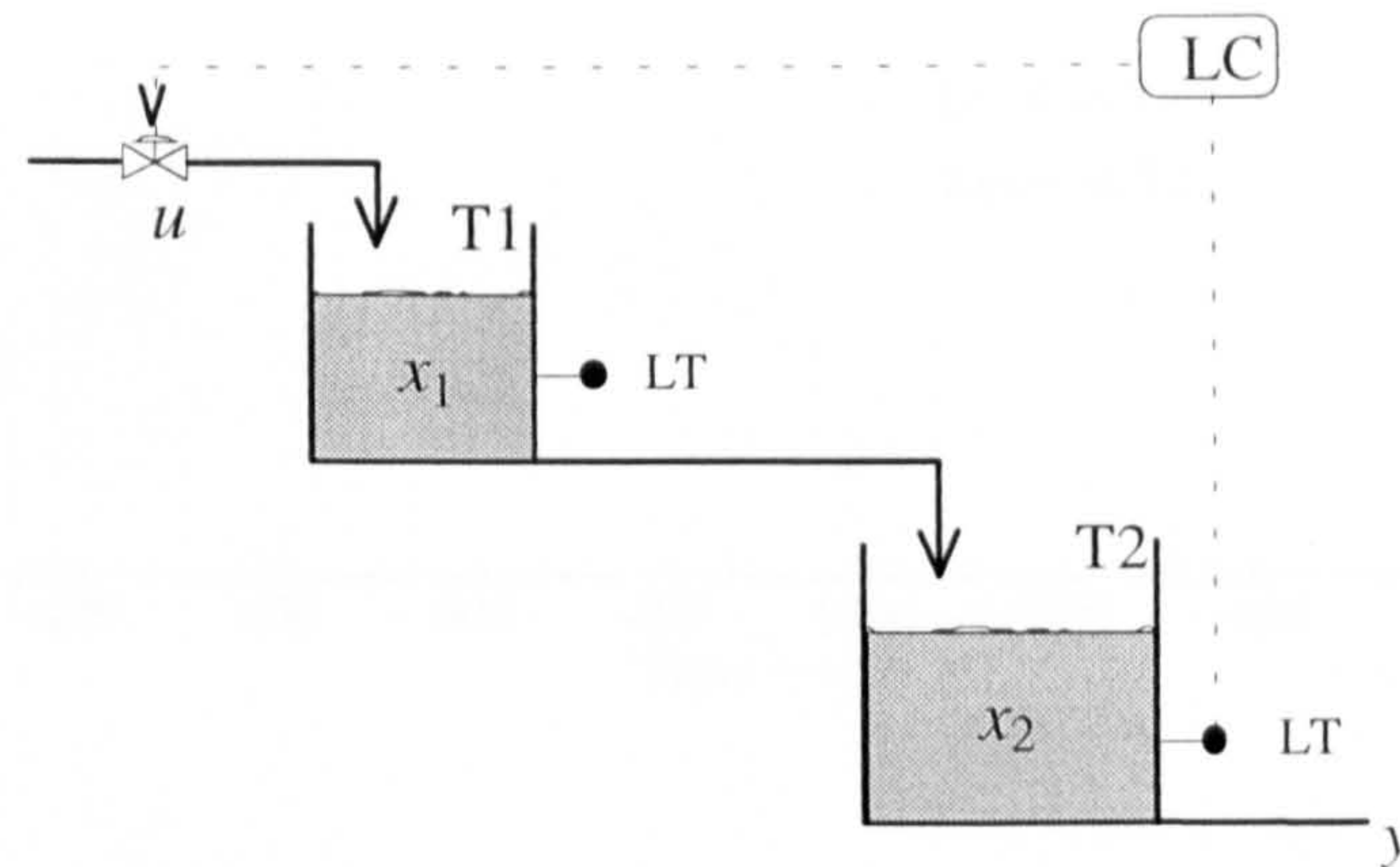


Figure 5.4 Two noninteracting tanks in series

The relative degree of the system is $r=2$ ($L_g h(x) = 0$, $L_g L_f h(x) = \alpha\beta\gamma_1 \neq 0$) in the full state space $\mathcal{R}^n$, $n=2$. With the following input/output linearizing state feedback of the full system

$$\begin{aligned}
 u &= \frac{v - h(x) - \sum_{i=1}^r k_i L_f^i h(x)}{k_r L_g L_f^{r-1} h(x)} \\
 &= \frac{1}{k_2 \alpha \beta \gamma_1} (v - x_2 - \beta \gamma_1 (k_1 - k_2 \alpha \gamma_1) x_1 + \beta x_2 \frac{1}{\sqrt{x_2}} (k_1 x_2 + 0.5 k_2 \beta \gamma_1 x_1) - 0.5 k_2 \beta^2 \gamma_2^2)
 \end{aligned} \tag{5.6}$$

the closed-loop linear system $v - y$ is given by:

$$y + \sum_{i=1}^r k_i \frac{d^i y}{dt^i} = v$$

With a step change in setpoint of T2 level from zero (initial T2 level) to 8 meters, and initial T1 level of 10 m, the level responses of the GLC controller are shown in Figure 5.5, assuming all model parameters are known exactly ($\alpha=0.1$, $\beta=0.05$, $\gamma_1=0.2625$, $\gamma_2=1.05$).

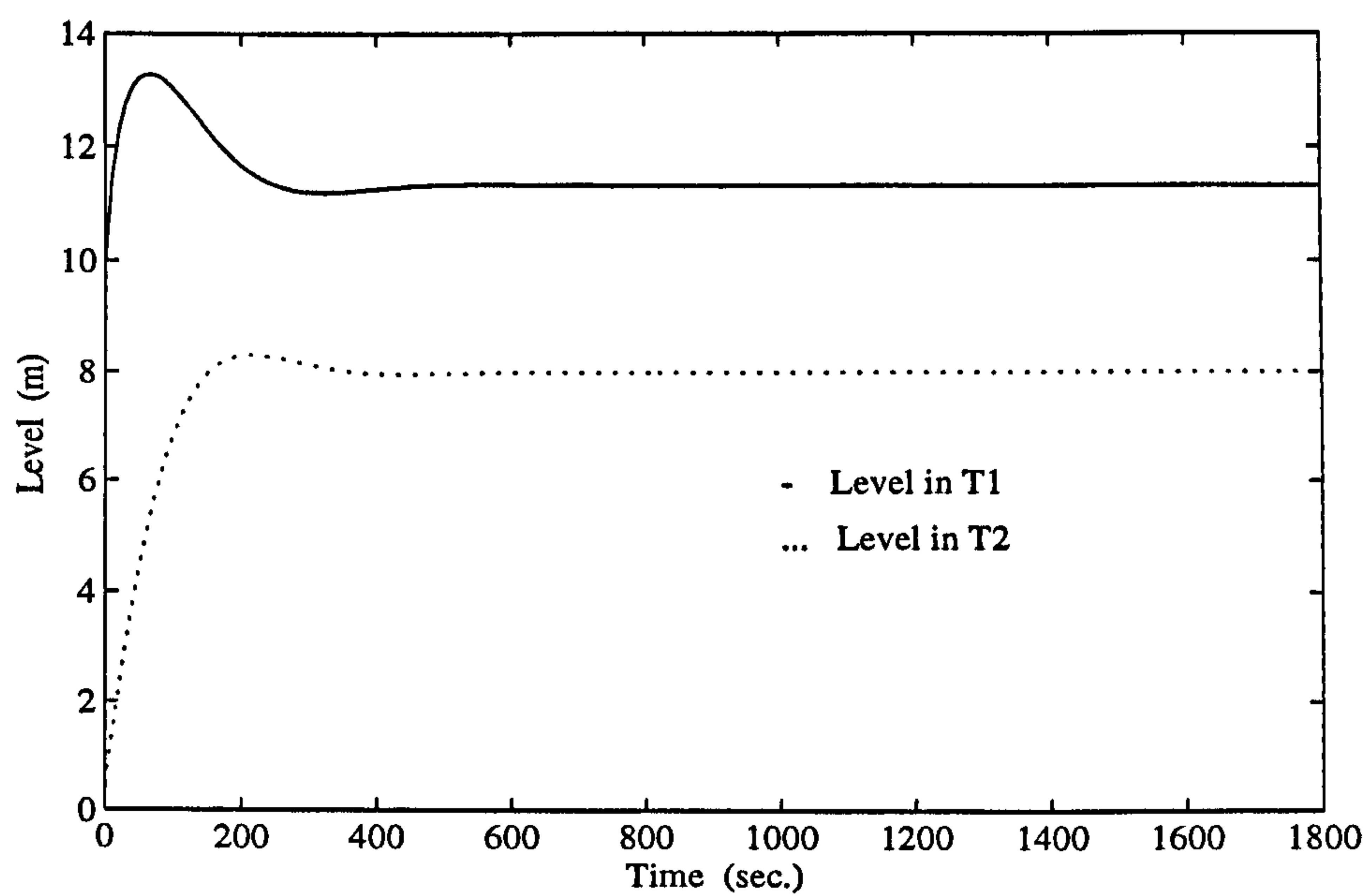


Figure 5.5 GLC control with perfect model

However, if the model parameter γ_1 used in the control law Eq (5.6) is 50% bigger than the true value, the same control law and controller tuning parameters yields a degraded performance as shown in Figure 5.6.

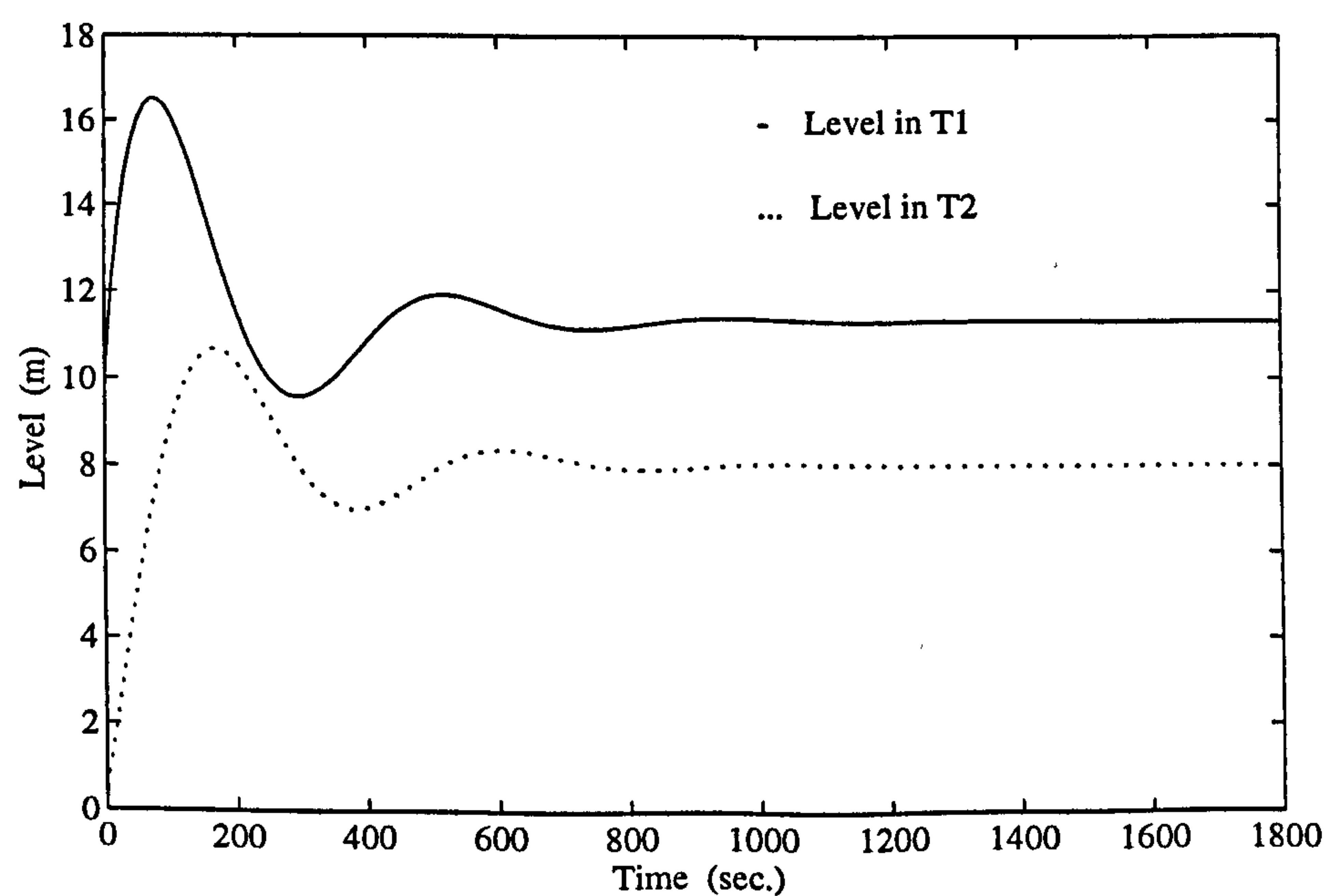


Figure 5.6 GLC control with mismatch on a model parameter

The simulation results demonstrate that the performance of the GLC controller in the given example depends on the quality of the model. Model mismatch leads to degraded control performance. As a consequence of the poor control, T1 must be designed with a big safety margin to avoid overfilling the tank which, in turn, increases its cost.

Considering the partitioning space approach, the system in Eq (5.5) can be partitioned into two subsystems by selecting the level in T1 as the intermediate control input (it is the only possible variable that satisfies conditions in Eq (5.c4) and Eq (5.c5)). The partitioned system model is shown in Eq (5.7).

$$\begin{cases} \text{SysI} \begin{cases} \dot{x}_1 = -\alpha\gamma_1 x_1 + \alpha u \\ y_c = x_1 \end{cases} \\ \text{SysII} \begin{cases} \dot{x}_2 = -\beta\gamma_2 \sqrt{x_2} + \beta\gamma_1 x_1 \\ y = x_2 \end{cases} \end{cases} \quad (5.7)$$

The GLC approach is used for the controller design of subsystem SysII which has a relative degree of one. The GLC control law is given in Eq (5.8).

$$x_1^* = \frac{1}{k_1\beta\gamma_1}(\nu - x_2 - \beta\gamma_2\sqrt{x_2}) \quad (5.8)$$

Compared to Eq (5.6), the control law for the partitioned system is much simpler and easier to derive. A PI controller with constants $k=2.0$, $\tau_c=0.002$ is used for SysI as a secondary controller. The same two cases considered above are tested with the GLC controller cascaded to the PI controller (Figure 5.7). Compared to Figures 5.5 and 5.6, the cascaded control structure suppresses the effect of the model mismatch on the control system performance.

The example gives a simple illustration of the use of the partitioning space approach for the design of controllers for nonlinear systems. Some conclusions may be only valid for specific problems, however, the suppression of model uncertainty offered by the cascade control structure is general for most nonlinear systems. The partitioning space approach also allows various control techniques to be used for controlling a system for which a model is well defined or partially known.

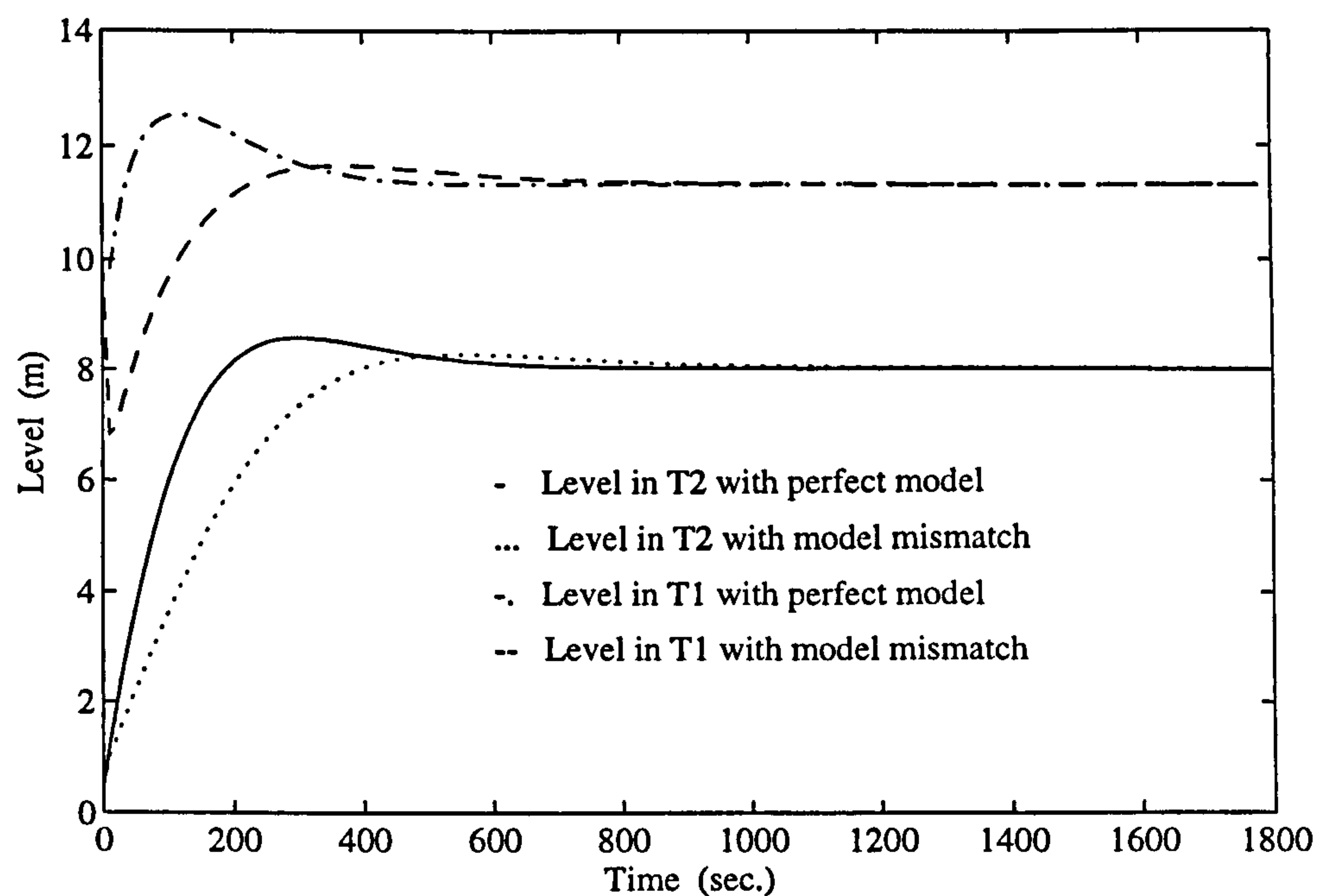


Figure 5.7 Level responses with a cascaded GLC and PI controller.

5.3 Multipurpose Batch Reactor System

In this section, temperature control of an enzymatic starch degradation process is investigated in a multipurpose batch reactor. The process consists of two processing stages including an initial starch gelatinization followed by starch degradation by α -amylase. Good temperature control is important for this process for several reasons:

- product quality is indirectly ensured by controlling the key operating variables such as temperature and pH along a pre-defined trajectory. Deviations from the desired trajectory may result in an off-specification product which cannot be detected until the end of the batch;
- enzyme is added to the reactor only when the operating variables are within a certain range. Poor control may lead to a delay of the enzyme addition which in turn results in a longer reaction time;
- poor control leads to higher operation cost;
- the manufacturing operation may require batches of different products, or a variable batch size for the same product. The controllers must be robust to all the changes.

A detailed description and a model of the system are given first. Characteristics of the system are discussed which motivate the use of nonlinear control techniques. The partitioning space design approach is used for the temperature control of the reactor and its effectiveness is evaluated by simulations and experiments.

5.3.1 Batch Reactor System

A schematic diagram of the reactor system was given in Figure 4.3.c. The reactor is a 100 L jacketed stainless steel vessel with a multiblade agitator. The reactor temperature can be controlled in several ways depending on the type of reaction and control requirements:

- *Sparge heating.* Steam is injected directly into the reactor through control valve CV1-3 and blocking valve AV1-5. The temperature of the reactor is controlled by manipulating the steam flow rate;
- *Steam jacket heating.* The temperature of the reactor is controlled by controlling the steam flow through the jacket. The condensed water is immediately withdrawn from steam trap ST1-1. The two way control valves AV1-2 and AV1-32 must be closed and AV1-33 must be open;
- *Recirculating the reactor contents through an external heat exchanger (HE1).* The reactor contents are recirculated through HE1 where either heating or cooling control can be performed;
- *Water jacket heating/cooling control through heat exchanger HE2.* In this scheme, the reactor temperature is controlled by manipulating the cooling water flow to the jacket recirculating route and the steam flow to HE2. To avoid simultaneous heating and cooling (and thus waste of resources), the valves are configured in a split-range as discussed in section 4.5. This may pose control difficulties which will be illustrated in the next section.

In this work, sparge heating is used for fast gelatinization of the starch slurry at a high temperature (105 °C) and water jacket control is used for the starch degradation phase.

For convenience, the model of the reactor system under the sparge heating control and water jacket control (described in Chapter 4) is summarised below:

Under sparge heating control:

$$\frac{dV_r}{dt} = \frac{1}{\rho} W_{ss} \quad (5.9)$$

$$\frac{dP_{dex}}{dt} = \frac{V_{\max} C_s}{(K_m + C_s)} \quad (5.10)$$

$$V_r \frac{dT_r}{dt} = -\alpha_1 (T_r - T_j) - \alpha_2 (T_r - T_{rm}) + \alpha_3 (H_{sp} - C_p T_r) W_{ss} + \alpha_3 Q \quad (5.11a)$$

$$\frac{dT_j}{dt} = \frac{1}{\tau_{sp}} (T_r - T_j) \quad (5.12a)$$

Under water jacket control, Eq (5.11a) and Eq(5.12a) are replaced by (5.11b) and (5.12b), respectively.

$$V_r \frac{dT_r}{dt} = -\alpha_4 (T_r - T_j) - \alpha_2 (T_r - T_{rm}) + \alpha_3 Q \quad (5.11b)$$

$$V_{jf} \frac{dT_j}{dt} = \beta_1 (T_r - T_j) + F_j (T_{jin} - T_j) \quad (5.12b)$$

$$T_{jin} = T_{cin} + \phi_{he} (T_{hin} - T_{cin}) \quad (5.13)$$

$$T_{cin} = T_j + (F_{cw}/F_j) \cdot (T_{cw} - T_j) \quad (5.14)$$

$$\phi_{he} = \frac{2 C_{ps} W_s}{C_{ps} W_s + 2 C_{ps} K_{he} F_j W_s + \rho_j C_{pj} F_j} \quad (5.15)$$

$$W_s = k^{he} \alpha^{(x^s/100)-1} - k^{he} \alpha^{(-1)} \quad (5.16)$$

$$F_{cw} = \begin{cases} k_1^{cw} x^{cw} & 0 \leq x^{cw} \leq 40 \\ k_2^{cw} (x^{cw} - 40) + 40 k_1^{cw} & 40 < x^{cw} \leq 100 \end{cases} \quad (5.17)$$

$$W_{ss} = \begin{cases} k_1^{sp} x^{sp} & 0 \leq x^{sp} \leq 15 \\ k_2^{sp} (x^{sp} - 15) + 15 k_1^{sp} & 15 < x^{sp} \leq 50 \\ k_3^{sp} (x^{sp} - 50) + 35 k_2^{sp} + 15 k_1^{sp} & 50 < x^{sp} \leq 100 \end{cases} \quad (5.18)$$

Where $\alpha_1 = A_{20} U_{sp} / \rho C_p$, $\alpha_2 = A_{10} U_{10} / \rho C_p$, $\alpha_3 = 1 / \rho C_p$, $\alpha_4 = A U_w / \rho C_p$, $\beta_1 = A U_w / \rho_j C_{pj}$

Description of the variables and values of the parameters are given in the nomenclature.

5.3.2 Characteristics of the Reactor System

The reactor system has the following characteristics:

1. Nonlinear jacket recirculation flow. Experimental results reveal that the jacket recirculation flow has a strong influence on the jacket dynamics. Figure 4.7 showed the three possible situations of the jacket recirculation route. During water jacket control, the desired heating rate is controlled by manipulating the heat exchanger steam flow rate and the reactor jacket is operating in the *full heating* region. The desired cooling rate is obtained by regulating the cooling water flow into the recirculation route, and therefore, either the inlet temperature of the jacket (*mixing* region) or the residence time of the cooling water in the jacket (*cooling* region).
2. Constraint on the jacket inlet temperature. During water jacket control, the jacket inlet temperature is constrained for safety reasons to below 98 °C. Violation of this constraint will cause the control sequence to alarm and an emergency control procedure to be initiated. This will be discussed in detail in Chapter 6. During sparge heating control, a constraint on the reactor volume contents is also imposed.
3. Variable control structure in the water jacket control. By combining Eq (5.13) and Eq (5.14), the last term in Eq (5.12b) can be expanded as follows:

$$u = F_j(T_{jin} - T_j) = \underbrace{F_{cw}(T_{cw} - T_j)}_{\text{term 1}} + \underbrace{\varphi_{he}[F_j(T_{hin} - T_j) - F_{cw}(T_{cw} - T_j)]}_{\text{Term 2}} \quad (5.19)$$

Term 1 represents the effect of the *cooling water flow rate* (F_{cw}) on the jacket dynamics and *Term 2* represents the effect of the *steam flow rate* (φ_{he} , and hence W_s). CV1-4 and CV1-7 are not permitted to open simultaneously thus avoiding counterproductive heating and cooling. Therefore, at least one of the terms will be 0 at any one time. When the steam valve is shut, the jacket route will be operated in the full cooling or mixing region and the φ_{he} term will be zero. This results in a bi-linear system. When the recirculation route is in the full heating region, the jacket temperature will be a non-linear function of the steam flow rate resulting in a nonlinear system. With such a system, a PID controller performs poorly and has to be detuned, as shown in the next section.

4. Multiple control schemes to cover a wide range of operating conditions. The starch liquefaction process consists of the gelatinization and degradation stages. To achieve a fast

gelatinization, a high operating temperature is required (*e.g.* 95-105 °C). This operating temperature is normally achieved by sparge heating in industry. As this is "one way" control (heating only), overshoots are particularly unfavourable as they cannot be counteracted. Steam injection will also dilute the reactor contents and in the worst case, the addition of a large amount of steam may cause the reactor to exceed its maximum capacity.

On the other hand, the starch degradation process is normally carried out at lower operating temperatures (*e.g.* 55-85 °C). This requires a control scheme that can perform both heating and cooling. Table 5.1 summarises the temperature range for each control scheme.

Control Scheme	Heating/cooling	Temperature range
sparge heating	heating only	up to 145 °C
steam jacket control	heating only	up to 145 °C
water jacket control	heating and cooling	25 - 95 °C
external recirculation through HE1	heating or cooling	25 - 85 °C

Table 5.1 Reactor temperature control range under each control scheme.

In summary, due to the above characteristics, a linear controller or control strategies that are based on a linearized model are expected to give poor control performance. That, in turn, can cause longer reaction times, higher operation costs and poorer product quality. In the worst case, poor control may also cause control sequence alarms and emergency shut downs, with loss of the batch.

5.4 Controller Design and Synthesis

In this section, the design of controllers for the temperature control of the reactor is investigated using the partitioning space design procedure and GMC and GLC techniques. A starch degradation process by α -amylase is considered, consisting of an initial starch gelatinization and starch degradation by the enzyme. The desired reference profile of the reactor temperature for the process is determined a priori and is given in Figure 5.8. In order to achieve a fast gelatinization of the starch slurry, a high operating temperature is required. This is the so called high-temperature-short-time (HTST) gelatinization process. As the α -amylase enzyme is

denatured very quickly at high temperature, the reactor temperature must be brought down to a new lower setpoint before the α -amylase is added to initiate the degradation. However, the relationship between enzyme activity and temperature is not available and the ideal reference conditions were determined experimentally.

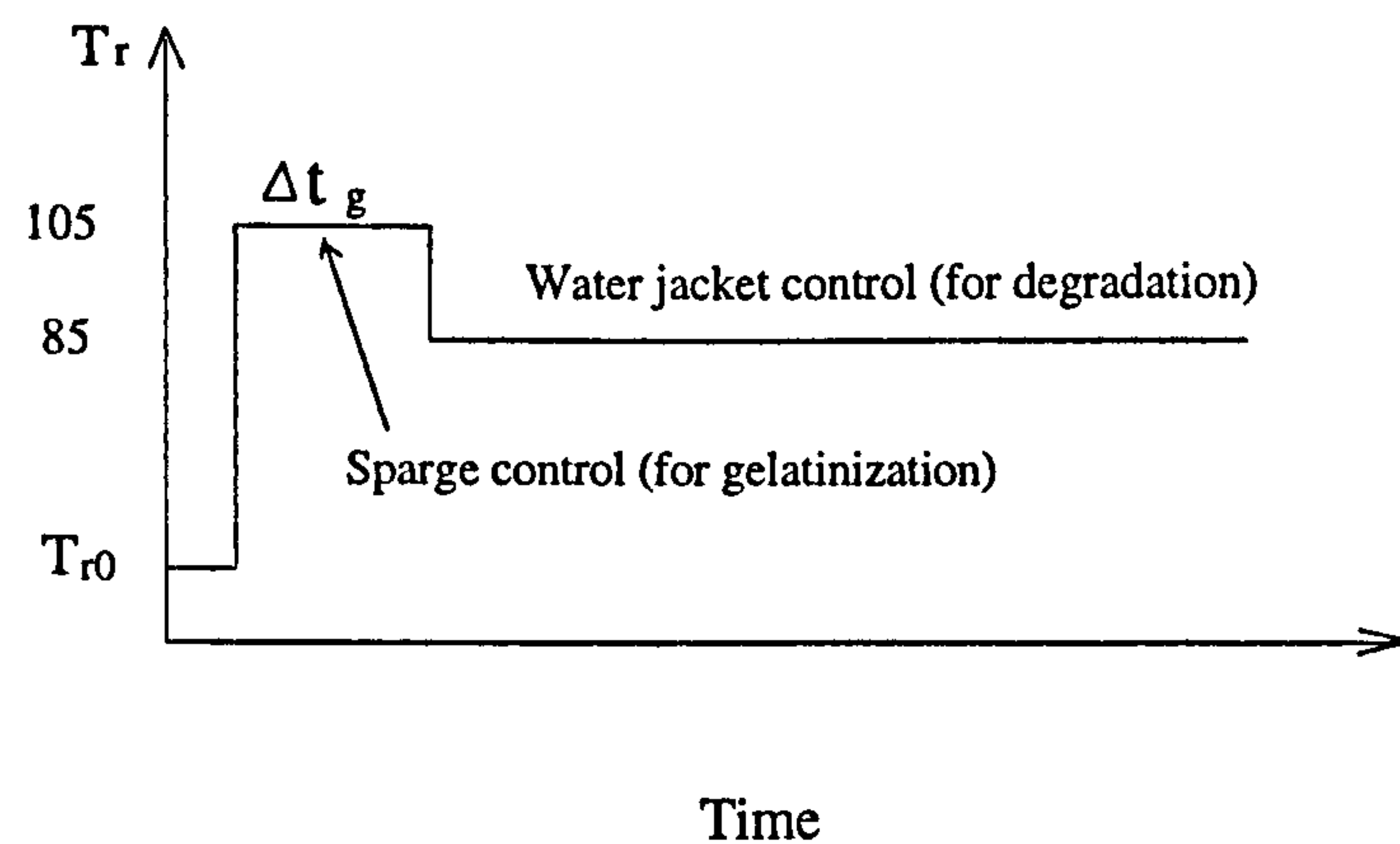


Figure 5.8 Reference temperature profile for the starch degradation process by α -amylase

5.4.1 Synthesis of Control Law

The GMC and GLC controllers for the temperature control of the reactor system described by Eq (5.9)-(5.18) are developed. Several control configurations are investigated using the partitioning space design approach and the GLC and GMC methods.

a). Synthesis of control laws under sparge heating control

For the reactor system under sparge heating, the relative degree of the system is 1 ($L_g L_f^0 h(x) = L_g h(x) = (H_{sp} - C_p T_r) / \rho C_p V_r \neq 0, \forall x \in \Omega, T_r \neq H_{sp} / C_p$). Therefore, the GLC and GMC approaches can be used directly for the controller design. The control laws based on the GMC and GLC approaches are given in Table 5.2. For the detailed design procedure the reader is referred to the original papers (Lee and Sullivan, 1988; Kravaris and Chung, 1987). Values of the controller settings are summarised in Table 5.5.

Controller	Under sparge heating
PI controller	$v^{sp} = k_1^{pi}(T_r^* - T_r) + k_2^{pi} \int (T_r^* - T_r) dt$
GMC controller	$W_{ss} = \frac{V_r v^{sp} + a_1(T_r - T_j) + a_2(T_r - T_{rm}) - a_3 Q}{a_3(H_{sp} - C_p T_r)}$
GLC controller	$W_{ss} = \frac{V_r / \beta (v^{sp} - T_r) + \alpha_1(T_r - T_j) + \alpha_2(T_r - T_{rm}) - \alpha_3 Q}{\alpha_3(H_{sp} - C_p T_r)}$

Table 5.2 Control algorithms under the sparge heating control

As the relative degree of the system ($r=1$) is less than the order of the system ($n=3$), the input-output feedback linearization does not guarantee that the internal dynamics are stable. The jacket temperature T_j will not exceed the reactor temperature under sparge heating as seen in Eq (5.12a). However, the stability of the reactor contents volume V_r is not guaranteed (*i.e.* it is not guaranteed that V_r will remain within its bounds).

Let assume the control objective is to maintain the reactor temperature at a defined setpoint. The reactor contents volume V_r is bounded only when the following condition is satisfied.

$$Q = \frac{a_1}{a_3}(T_r - T_j) + \frac{a_2}{a_3}(T_r - T_{rm}) \quad (5.20)$$

As the reactor temperature is not less than both the jacket temperature and the room temperature T_{rm} during the sparge heating, the right hand side (the heat of reaction Q) must be positive. This implies that the reaction must be exthothermic. If Q is less than the value of the RHS, steam is needed to maintain the reactor temperature at a constant value. This results in an increase in volume of the reactor contents. However, if Q is greater than the value of the RHS, the reactor temperature will increase and cannot be maintained constant. In practice, the above condition is rarely satisfied throughout the reaction. Therefore, the sparge heating control method is only suitable for the reactor heat-up and cannot be used for temperature regulatory control over a long period of time.

The GMC and GLC control laws define the control action which is needed in order to obtain the desired response. They contain both feedback control action to eliminate any offset in the reactor temperature and feedforward control action to compensate for the heat transfer between the reactor and its environment and for the "disturbance" caused by the heat of the reaction. To implement the control law, the steam flow rate must be converted into a control valve position

based on the control valve model. The implementation details of these controllers on the plant are given in Appendix.

b) Synthesis of control laws under water jacket control

When the temperature of the reactor contents is controlled by the water jacket heating/cooling scheme, the relative degree of the system is 2 with respect to the control input u . Neither the GMC approach nor the input-output feedback linearization method can be used directly for such a system due to the reasons discussed in the previous section. The partitioning space design approach is therefore considered.

The reactor system is partitioned into two subsystems, SysI and SysII which correspond to the jacket and the reactor, respectively. The jacket temperature T_j satisfies both conditions in Eq (5.c4) and (5.c5). It is selected as the new control input to SysII and the intermediate output of SysI. The partitioned system is given by Eq (5.21). The input of the system u is given by Eq (5.19).

$$\left\{ \begin{array}{l} \text{SysI} \quad \begin{cases} \frac{dT_j}{dt} = 1/V_j (\beta_1 (T_r - T_j) + u) \\ y_c = T_j \end{cases} \\ \text{SysII} \quad \begin{cases} \frac{dT_r}{dt} = 1/V_r (-(\alpha_4 + \alpha_2)T_r + \alpha_2 T_{rm} + \alpha_4 T_j + \alpha_3 Q) \\ y = T_r \end{cases} \end{array} \right. \quad (5.21)$$

In this study, the GLC and GMC approaches are considered for the controller design of the outer loop. In SysI, two cases are investigated:

- use PI controller for the servo control of the jacket temperature assuming that the subsystem model is poorly defined.
- use GLC or GMC approaches for the servo control of the jacket temperature assuming that the subsystem model is relatively adequate.

The resulting cascade control structures for the two cases are shown in Figures 5.9a and 5.9b.

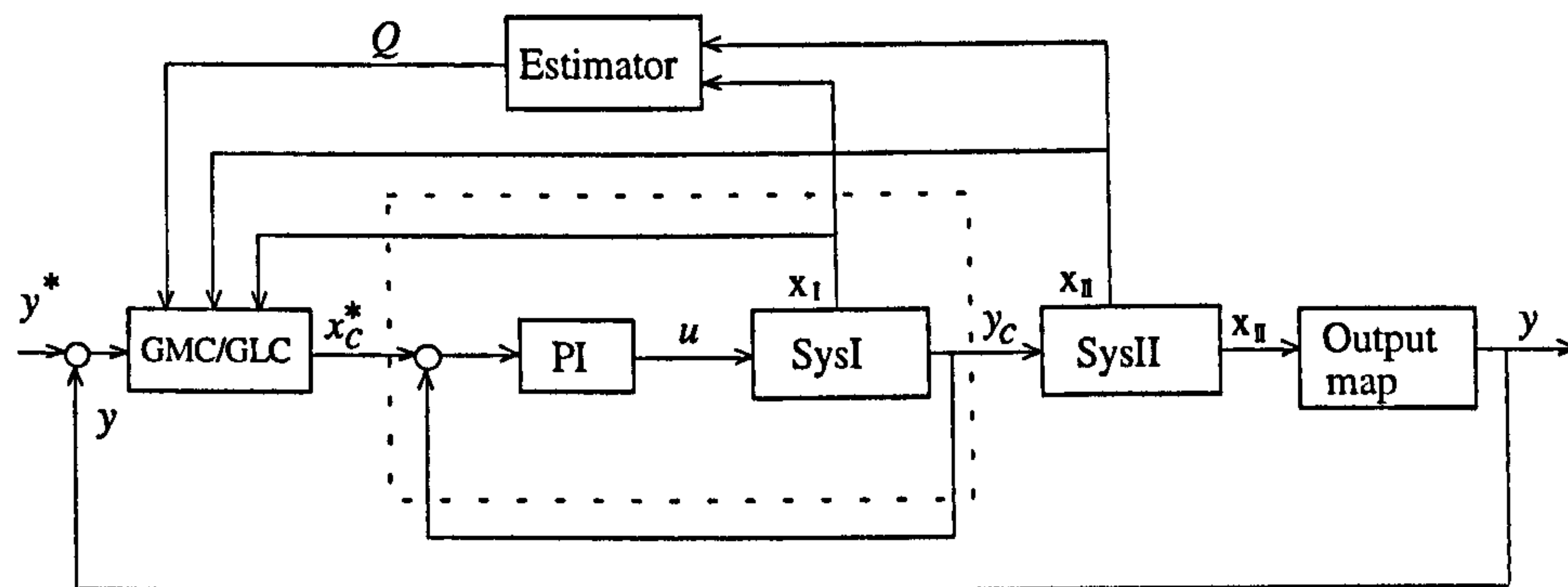


Figure 5.9.a Cascaded GMC/GLC and PI controller structure

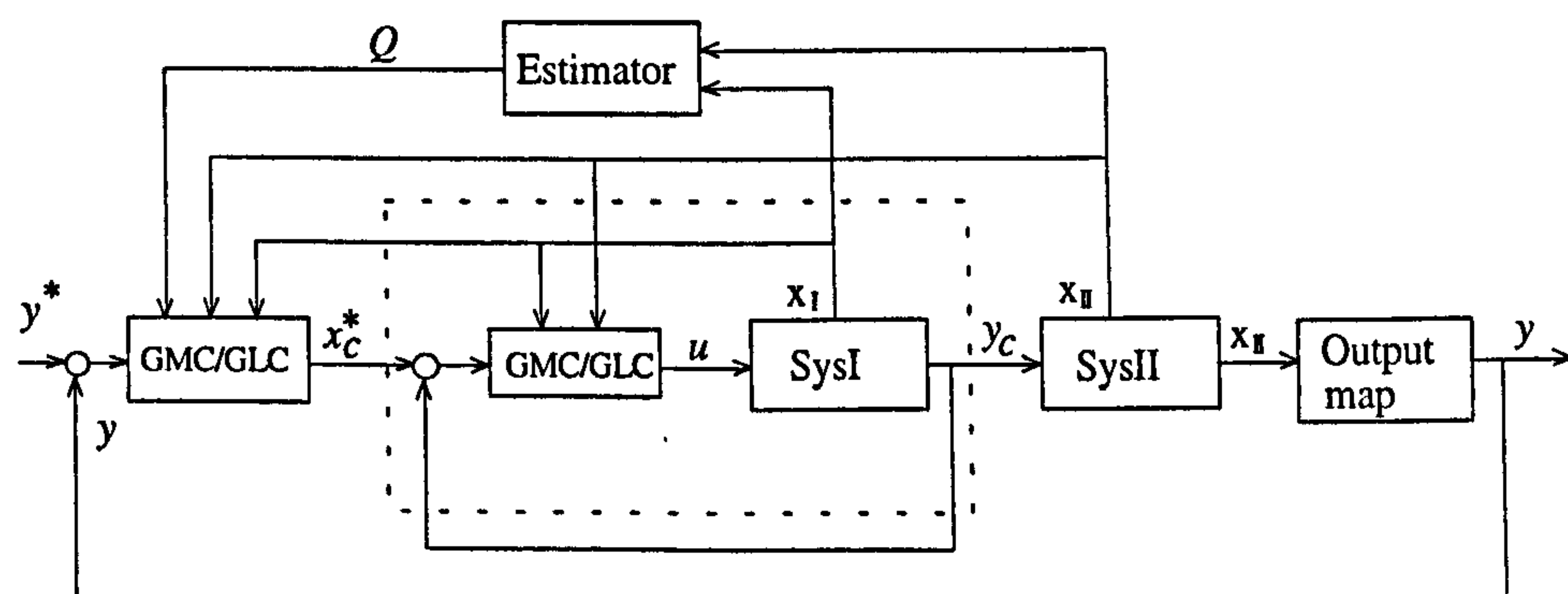


Figure 5.9.b Cascaded GMC/GLC controller structure

The GMC or GLC controller calculates the necessary jacket temperature in order to drive the output of the reactor (T_r) along a desired trajectory. The calculated jacket temperature T_j^* serves as the setpoint of the secondary controllers. The control action u is then calculated by the control law of the secondary controllers. The detailed GMC and GLC control algorithms are summarised in Table 5.3. Values of the parameters are given in the nomenclature. Values of controller settings are summarised in Table 5.5.

Controller	Under water jacket control
External linear controller	$v^{wj} = k_1^{wj} (T_r^* - T_r) + k_2^{wj} \int (T_r^* - T_r) dt$
External linear controller	$v^j = k_1^j (T_j^* - T_j) + k_2^j \int (T_j^* - T_j) dt$
GMC - primary	$T_j^* = T_r + \frac{1}{a_4} (V_r v^{wj} + a_2 (T_r - T_{rm}) - a_3 Q)$
GMC - secondary	$u = V_j v^j - \beta_1 (T_r - T_j)$
GLC - primary	$T_j^* = T_r + \frac{V_r}{a_4 \beta} (v^{wj} - T_r) + \frac{a_2}{a_4} (T_r - T_{rm}) - \frac{a_3}{a_4} Q$

Table 5.3 Control algorithms under water jacket control

5.4.2 Estimation of the Heat of Reaction

Most model based control approaches, including GMC and GLC, require that all state variables must be measured or estimated in order to implement the control law. Direct measurements are available for T_r , T_j and W_r (V_r) while the heat of reaction must be estimated. Various approaches have been used to estimate the rate of energy production (Juba and Hamer, 1986; Valliere and Bonvin, 1989). Estimation of heat of a reaction based on a detailed kinetic model requires a good knowledge of the kinetic parameters. These are difficult to obtain in food processes, and when available, are subject to a large degree of uncertainty. The empirical heat estimation method has the advantage of taking all the dynamics of the reactor into account. However, the resulting model is specific to the given process. In this work, the deterministic on-line energy balance approach was used for the estimate of Q .

Exponential filters (Cott and Macchietto, 1989) are used on the estimate of Q . Although direct measurement of the reactor temperature and jacket temperature is available, the derivative of the reactor temperature must be estimated on-line. As numerical differentiation is very sensitive to measurement errors, a high-order difference equation is used for calculating the derivative of the reactor temperature to improve the performance of the estimator and low-pass filters are used on the measurements to remove the high-frequency noise. The full description of the estimator in a discrete form is given as following:

During sparge heating:

$$T_{rf}^{(k)} = T_{rf}^{(k-1)} + \frac{\tau_s}{\tau_{rf}} (T_r - T_{rf}^{(k-1)}) \quad (5.22)$$

$$T_{jf}^{(k)} = T_{jf}^{(k-1)} + \frac{\tau_s}{\tau_{jf}} (T_j - T_{jf}^{(k-1)}) \quad (5.23)$$

$$\frac{dT_{rf}^{(k)}}{dt} = \frac{1}{2\tau_s} (3T_{rf}^{(k)} - 4T_{rf}^{(k-1)} + T_{rf}^{(k-2)}) \quad (5.24)$$

$$Q^{(k)} = \frac{V_r}{\alpha_3} \frac{dT_{rf}^{(k)}}{dt} + \frac{\alpha_1}{\alpha_3} (T_{rf}^{(k)} - T_{jf}^{(k)}) + \frac{\alpha_2}{\alpha_3} (T_{rf}^{(k)} - T_{rm}) - (H_{sp} - C_p T_{rf}^{(k)}) W_{ss}^{(k)} \quad (5.25)$$

$$Q_f^{(k)} = Q_f^{(k-1)} + \frac{\tau_s}{\tau_{qf}} (Q^{(k)} - Q_f^{(k-1)}) \quad (5.26)$$

During *water jacket control*, Eq (5.25) is replaced by Eq (5.27):

$$Q^{(k)} = \frac{V_r}{\alpha_3} \frac{dT_{rf}^{(k)}}{dt} + \frac{\alpha_4}{\alpha_3} (T_{rf}^{(k)} - T_{jf}^{(k)}) + \frac{\alpha_2}{\alpha_3} (T_{rf}^{(k)} - T_{rm}) \quad (5.27)$$

where the τ_s , τ_{rf} , τ_{jf} and τ_{qf} , are the sampling time, time constant on the reactor measurement, jacket measurement and Q estimation, respectively. Those values are given in the nomenclature.

5.4.3 Disturbance Decoupling

For the batch reactor system, the heat of reaction may be viewed as a major "disturbance" which pushes the reactor temperature away from a desired trajectory. This disturbance cannot be measured on-line and has to be estimated, as discussed in the previous section.

One way of dealing with the heat reaction is to represent Q as a function of the reaction rate. The estimation problem is therefore transformed into the estimation of the concentration of each component in the reaction. For processes with a complicated reaction mechanisms, obtaining a good kinetic model and parameters is very difficult. Even when they are available they are often subject to a great deal of uncertainty and the estimation of the concentration of each component is required.

In this work, the heat of reaction is viewed and modelled as a "disturbance" input to the reactor temperature control system, and estimated directly from the measurements of the reactor and jacket temperatures. Modelling the heat of reaction as a disturbance input has several advantages, compared to an estimation based on a detailed kinetic model:

- Q can be directly estimated based on the measurements of the reactor and jacket temperatures (available) rather than from the estimation of the concentration of each component (not usually measured on-line);
- for processes with unknown kinetics, the reference control trajectory is often determined experimentally. By modelling Q as a disturbance, the design and implementation of the controller for the temperature control of the reactor system can be explicitly separated from the detailed process kinetic studies. This will simplify the design procedure and widen its application.

However, the relative degree of the system with respect to Q is 1 ($L_p L_f^0 h(x) = L_p h(x) = 1/\rho C_p V_r \neq 0$, for $\forall x \in \Omega$). As the disturbance matching condition is not satisfied ($q < r$), the GLC approach cannot be applied to the controller design.

By partitioning the reactor system into two subsystems and by selecting the jacket temperature as an intermediate control input for SysII, $\frac{\partial h(x)}{\partial x_{II}} \gamma_{II} = \frac{\alpha_3}{V_r} \neq 0$. The disturbance Q can be decoupled from the reactor temperature T_r when a direct estimation of Q is available. The resulting control law for the jacket temperature contains a feedforward term to compensate for the effect of the heat of reaction on the reactor temperature.

5.4.4 Coordination Rule

During the water jacket control, the controllers derived above provide the value of the control action u which drives the system along a desired trajectory. To implement the control laws, this control action must be related to the manipulated variables, in this case the cooling water flow rate F_{cw} and steam flow rate W_s . As discussed in section 5.3, the control u consists of two parts. The first part $F_{cw}(T_{cw} - T_j)$ represents the effect of the cooling water flow on the jacket dynamics. The second part $\phi_{he}[F_j(T_{hin} - T_j) - F_{cw}(T_{cw} - T_j)]$ represents the effect of the heat exchanger steam flow and the combined action of the steam flow and cooling water flow on the jacket dynamics. The split setting of the two control valves guarantees that the heating and cooling control actions

will not take place simultaneously. Therefore, the following coordination rules between the output of the controller u and the two manipulated variables F_{cw} and W_s can be established:

$$\begin{cases} F_{cw} = u / (T_{cw} - T_j) \\ \varphi_{he} = 0.0 \quad (W_s = 0.0) \end{cases} \quad \text{For } -u_w^{\max} \leq u \leq 0.0 \quad (5.28a)$$

$$\begin{cases} F_{cw} = 0.0 \\ \varphi_{he} = u / F_j (T_{hin} - T_j) \end{cases} \quad \text{For } u_s^{\max} \geq u > 0.0 \quad (5.28b)$$

where $u_w^{\max}$ and $u_s^{\max}$ are the lower and upper bound of the controller output and correspond to the maximum cooling water flow rate and maximum steam flow rate respectively.

5.5 Simulation and Experimental Results

Several control configurations using the controllers developed in the previous section were investigated by simulation and experiment. Table 5.4 lists the various cases studied.

CASE	Gelatinization		Degradation under water jacket control			Reactor contents
	Temp.	Sparge heating	Temp.	Master controller	Slave controller	
I (a)	-	PI	65	PI	PI	Water
	(b) 105		85			Starch
II (a)	-	GMC	65	GMC	PI	Water
	(b) 105		85			Starch
	(c) 105		65			Water
III (a)	105	GLC	85	GLC	PI	Starch
	(b) 105		85			Starch
IV (a)	105	GMC	55	GMC	GMC	Water
	(b) 105		85			Starch

Table 5.4 Control system structures for the starch processes

5.5.1 Performance Criteria

Controller configurations are compared according to the following performance criteria:

1. *Rise-Time*. The Rise-Time (RT) indicates the speed of response of a controller with respect to a change in setpoint. It is defined as the time required for the output to reach 95% of the required change between its initial value and the new setpoint.

$$RT = \text{Time} \Big|_{y=y_0+0.95(y^{sp}-y_0)}$$

2. *Integral of the absolute value of the error (IAE)*, defined as:

$$IAE = \int_{t_0}^{t_f} |y^{sp} - y| dt$$

The integration is started once the Rise-Time is reached. Therefore, the IAE value indicates the performance of the controllers around the new setpoint. Integration time ($t_f - t_0$) is 75 min. for all experiments.

5.5.2 Controller Tuning

The partitioning space approach results in a cascaded control structure composed of two controllers, the inner loop secondary controller and the outer loop primary controller. The tuning procedure is carried out in two steps:

- Tuning the inner loop controller for the subsystem SysI to achieve a fast and stable servo response with respect to setpoint changes of the slave controller.
- Tuning the outer loop controller for the subsystem SysII to achieve good regulatory and tracking control with respect to the reference profile.

In order to obtain a faster inner loop servo response, smaller stable closed loop poles for the secondary than the primary loop must be specified. However, this requires a greater control action which may lead to a degraded control performance if saturation limits on the manipulated variable are imposed. Dore (1993) proposes guidelines for tuning geometric nonlinear controllers

and specifications of parameters for systems with relative degree of one and two are given by Soroush and Kravaris(1992).

To obtain tuning parameters for the reactor system, a number of open loop experiments were carried out to calculate the time constants of the system. The system has a dominant time constant of 16.4 minutes which corresponds to the reactor dynamics with respect to the jacket temperature, and a smaller time constant of 0.52 minutes which corresponds to the jacket dynamics. For the GLC controller, β parameters were determined based on the rules given by Soroush and Kravaris (1992). All external PI controllers were tuned by trial and error, in simulations first to obtain good initial values, which were then adjusted during the experiments. Tuning parameters used in experiments for each control configuration are given in Table 5.5.

	Gelatinization	Degradation under water jacket control	
CASE	Sparge heating	Primary controller	Secondary controller
I	$P=20, I=100$	$P=80, I=50$	$P=40, I=105$
II	$k_1^{pi}=0.012, k_2^{pi}=1.0E-4$	$k_1^{wj}=0.003, k_2^{wj}=3.05E-7$	$P=40, I=105$
III	$k_1^{pi}=96.0, k_2^{pi}=1.0E-4$	$k_1^{wj}=120, k_2^{wj}=1.22E-2$	$P=40, I=105$
IV	$k_1^{pi}=0.016, k_2^{pi}=1.0E-4$	$k_1^{wj}=0.003, k_2^{wj}=3.05E-7$	$k_1^j=0.05, k_2^j=1.0E-6$

Table 5.5 Controller settings in experiments

5.5.3 Servo Response

The control response based on the partitioning space approach depends on the quality of servo control in the inner loop. In this study, PI and GMC controllers were used for the servo control of the reactor jacket inlet temperature. To test the servo response of the PI and GMC controllers, several experiments were performed with various setpoint profiles generated by the master controllers. Some experimental results are shown in Figure 5.10-5.12.

The experimental results show that both the PI and GMC controllers in the secondary loop provided a fast and stable servo control of the jacket inlet temperature over a wide range of operating conditions.

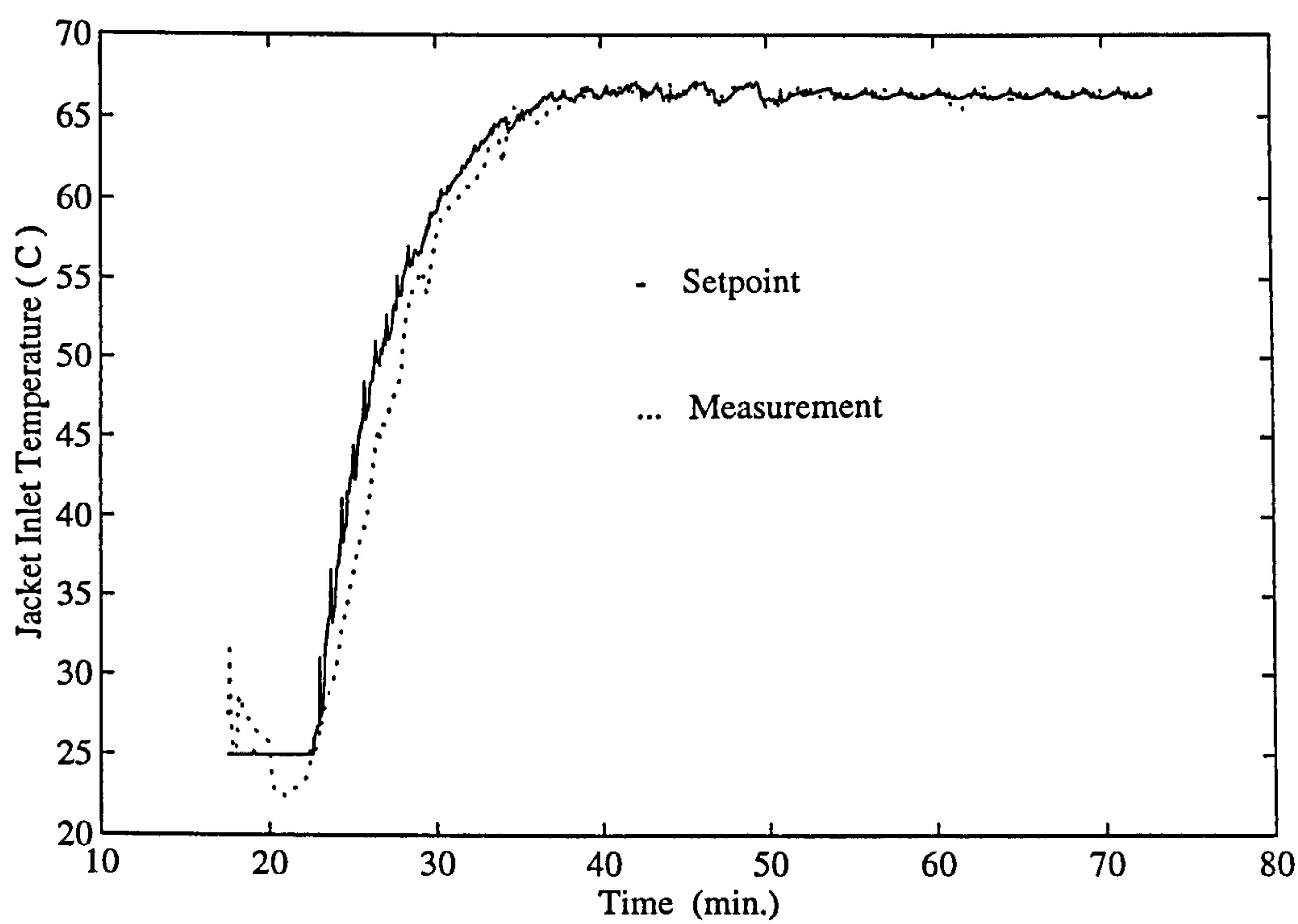


Figure 5.10 Servo control response of the PI controller when heating the reactor from room temperature to 65 °C (experiment 1)

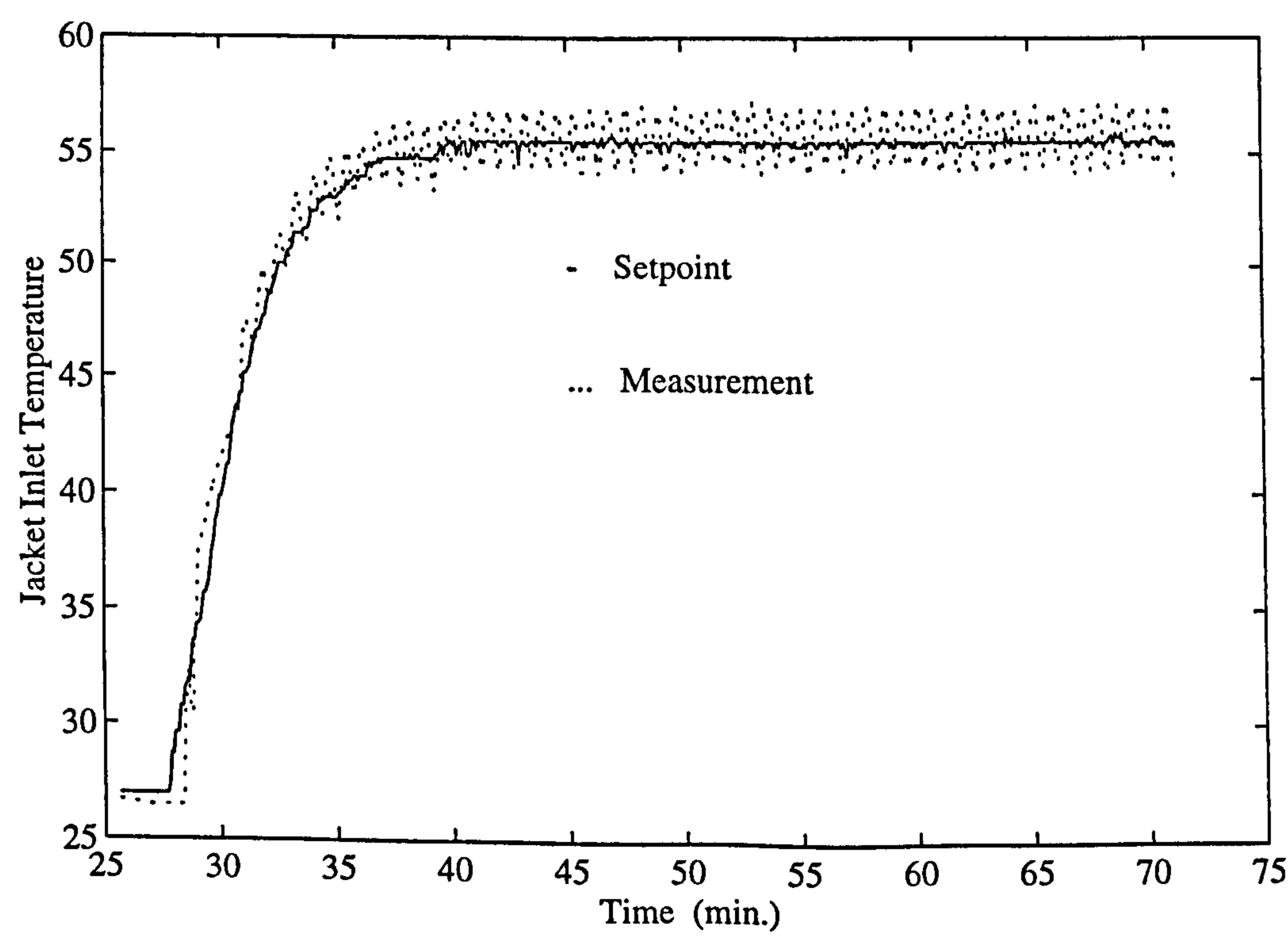


Figure 5.11 Servo control response of the GMC controller when heating the reactor from room temperature to 55 °C (experiment 2)

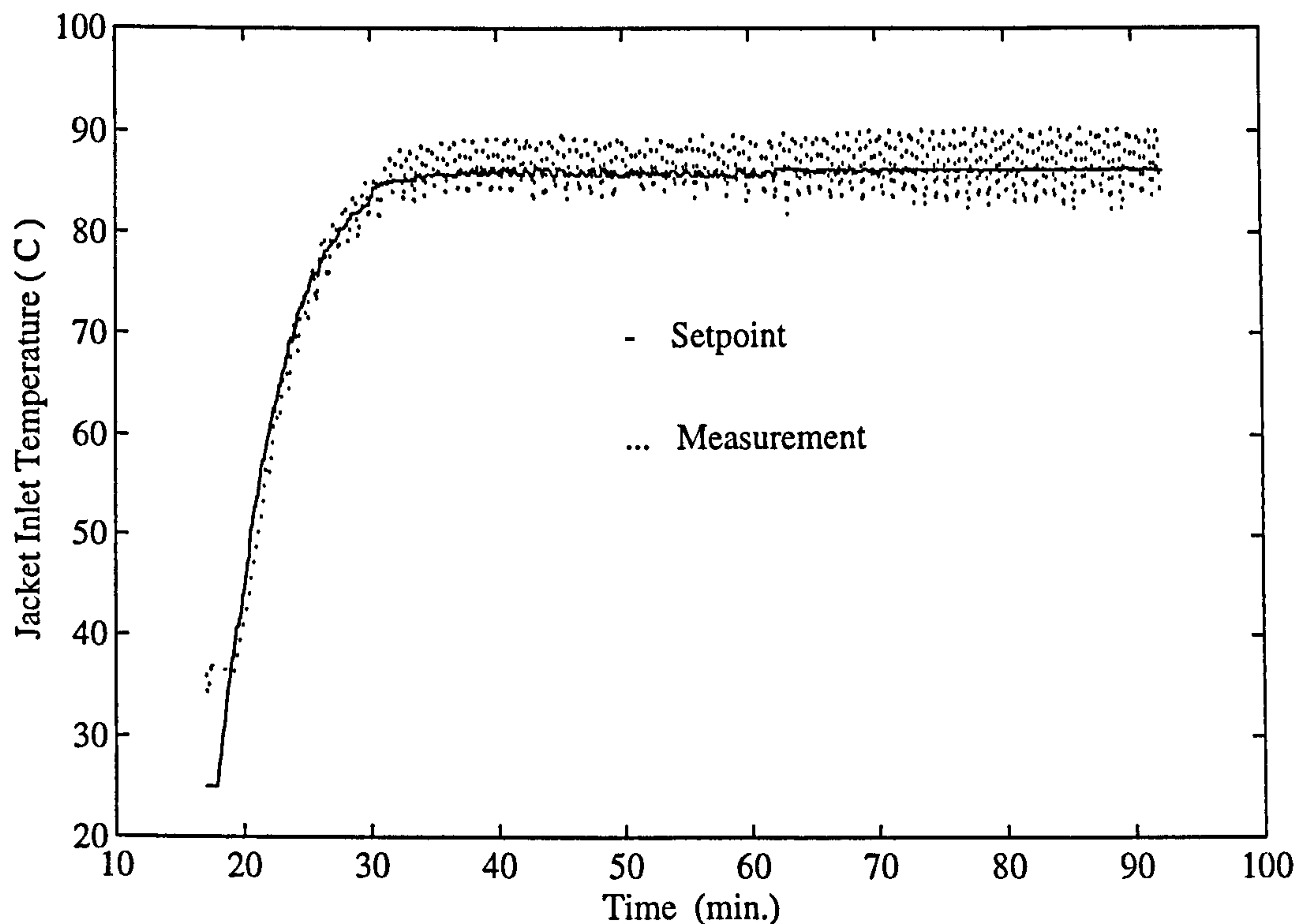


Figure 5.12 Servo control response of the GMC controller when heating the reactor from room temperature to 85 °C (experiment 3)

5.5.4 Estimation of Q

The heat of reaction was estimated on-line from the direct measurements of the reactor and jacket temperature using the estimation algorithms discussed in section 5.4.2. Figure 5.13 illustrates the performance of the estimator during the starch gelatinization and degradation phases.

The results show that heat is required during the starch gelatinization in order to convert the starch slurry into the gel form. The reaction appears to be endothermic. However, after the addition of the enzyme during the starch degradation stage, heat release from the reaction is initially high and then gradually reduces as the reaction proceeds. The estimated heat of reaction was more noisy when the control was switched from sparge heating to water jacket control.

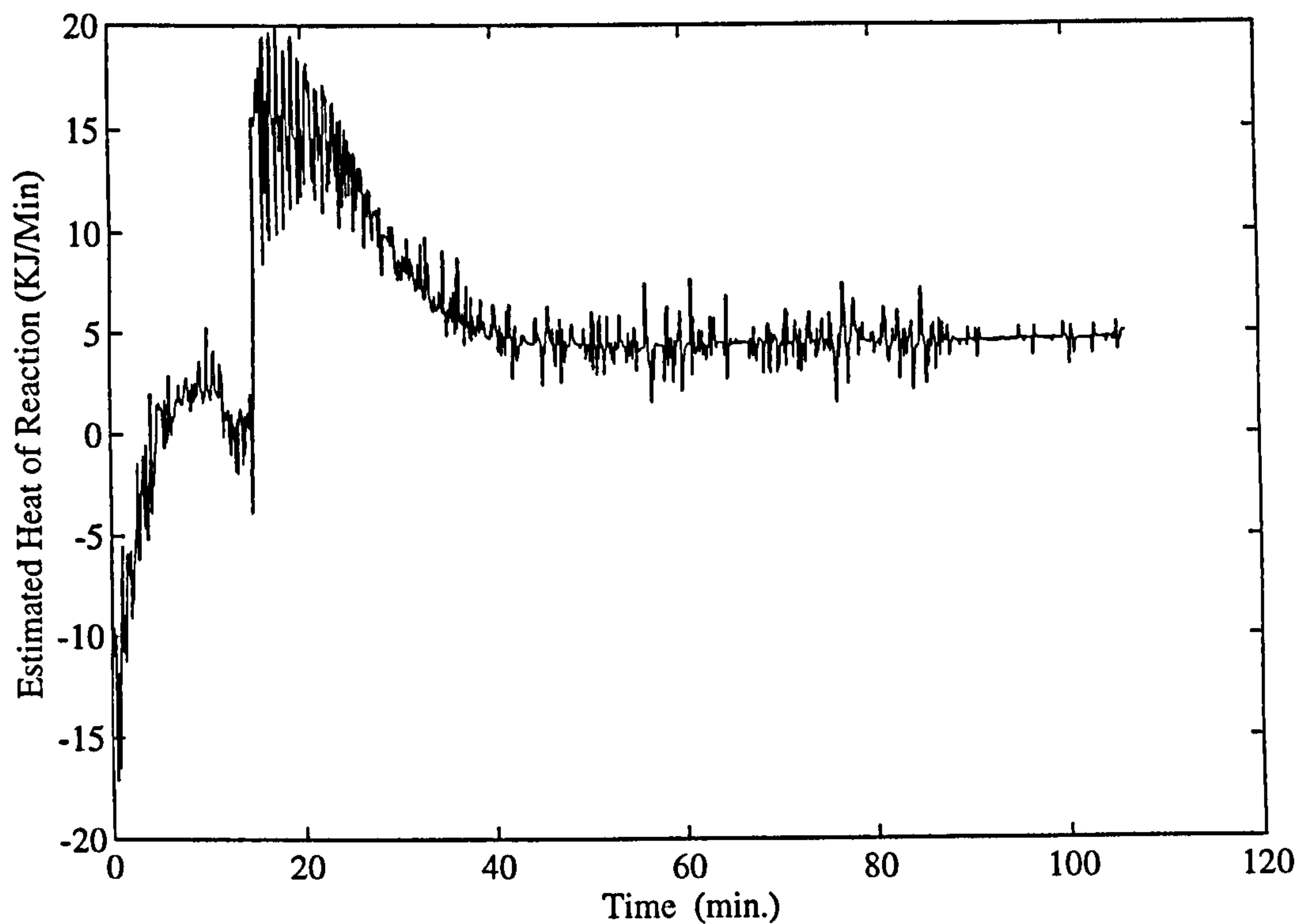


Figure 5.13 Estimated heat of reaction profile in starch gelatinization and degradation phases

5.5.5 Tracking Control Response

To test the tracking response of the controllers under each control configuration described in Table 5.4, simulations and experiments were carried out using a variety of reactor temperature reference profiles. An initial batch size of 80 Kg is used in all experiments (except *Case IIIa*). A 6% starch slurry is used. The sampling time is 10 seconds. The performance of controllers under each control configuration is compared using the performance criteria described in section 5.5.1.

5.5.5.1 PI Control

In Case I, PI controllers were used for both sparge heating and water jacket control. The controller parameters were extensively tuned by trial and error during the plant commissioning and are given in Table 5.5.

Case I(a). For the purpose of assessing the performance of the controllers over a wide range, the water jacket control scheme is used with cascaded PI controllers to bring the reactor temperature from initial room temperature (approx. 25 °C) to a desired temperature of 65 °C with a step change in setpoint. The experimental results are shown in Figure 5.14. An underdamped performance is obtained with a reactor temperature overshoot of 6 °C. The RT and IAE values for this case are 788 seconds and 3832, respectively.

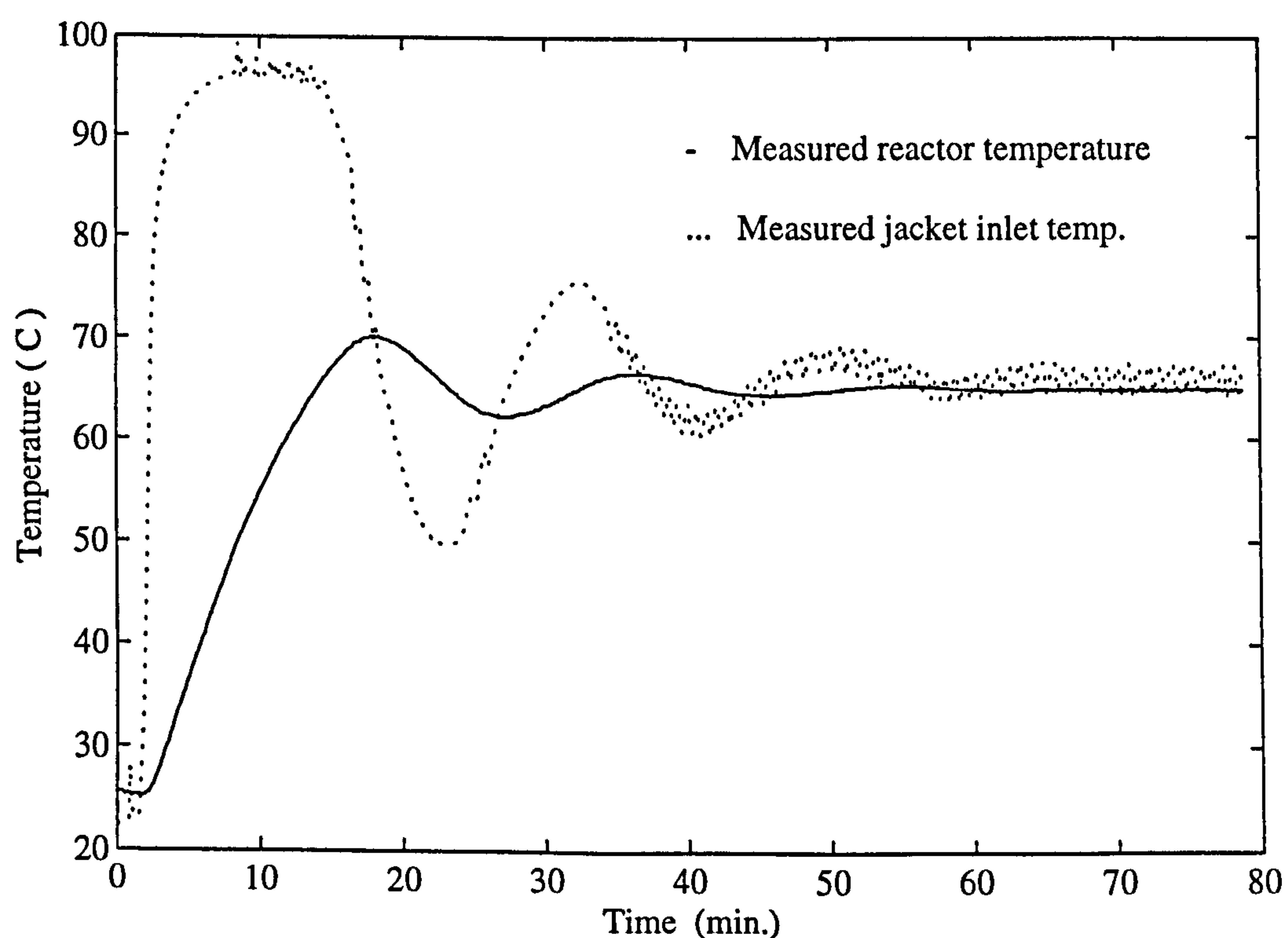


Figure 5.14 Reactor temperature responses under PID control - water jacket control to 65 °C (experimental results)

Case I(b). For fast starch gelatinization, the sparge heating control scheme is used with a PI controller and the reactor temperature setpoint of 105 °C. After the starch gelatinization, the temperature setpoint is changed to 85 °C for the enzymatic starch degradation phase under water jacket control. The same cascaded PI controllers as in *Case I(a)* are used in this phase but with a different initial temperature and setpoint. The temperature response is shown in Figure 5.15. With sparge heating, the reactor heats up quickly to the setpoint with an overshoot of 10 °C. As sparge heating is only "one way", the temperature cannot be brought down with this control scheme. When the temperature control is switched to the next setpoint (85 °C) under the water

jacket control, with the same tuning parameters used in *Case I(a)*, the reactor temperature undershoots by 16 °C, and takes 30 minutes to bring the temperature to the setpoint. This leads to an overall longer reaction time to achieve the desired starch conversion and to product quality problems due to a retrogradation reaction. The RT and IAE values for this case are 300 seconds and 9757 respectively.

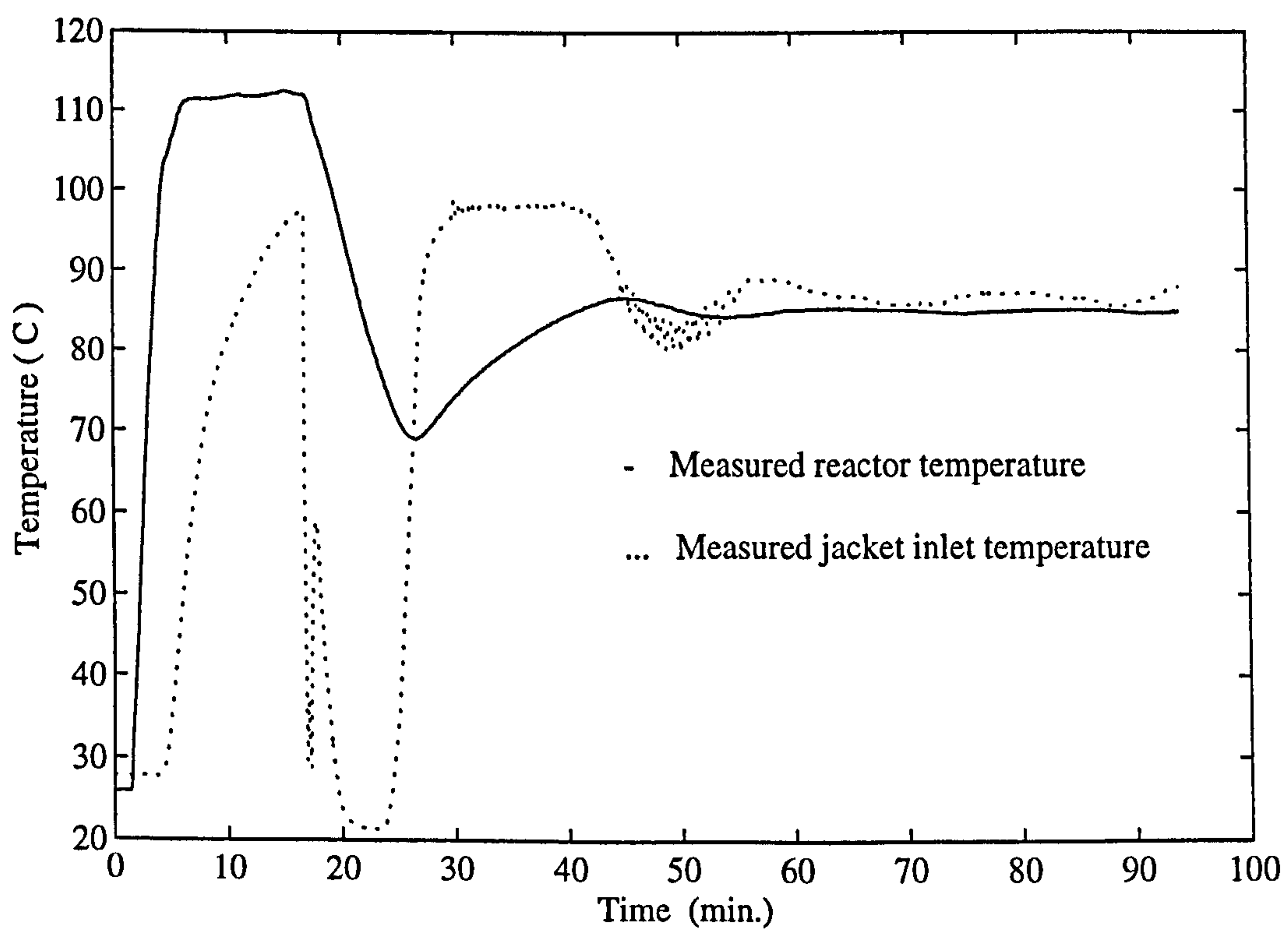


Figure 5.15 Reactor temperature responses under PID control - Sparge heating to 105 °C, then water jacket control to 85 °C (experiment)

These results demonstrate that PI type controllers have a rather poor tracking performance for the nonlinear process and should be retuned for different process conditions (*e.g.* reference profiles, batch sizes, raw materials). The experiments during plant commissioning have also shown that tuning PI controllers for this type of system is a difficult and time consuming task.

5.5.5.2 GMC and PI Control

In Case II, GMC controller is used for the sparge heating control of the reactor during gelatinization. A cascaded GMC (primary loop) and PI controller (secondary loop) is used for water jacket control during starch degradation. Several experiments were conducted involving four different reference profiles.

Case II(a). The reference temperature profile is as in *Case I(a)*. This tests the performance of the cascade controllers over a wide range of operating conditions. The experimental results are shown in Figure 5.16. The reactor temperature was brought to the setpoint in 20 minutes with a 0.5 °C overshoot. Compared to *Case I(a)*, the jacket inlet temperature decreases earlier to prevent reactor temperature overshoot. The performance of cascaded GMC/PI controller is superior to that of PI control alone as indicated by the RT value of 835 seconds and IAE value of 2021.

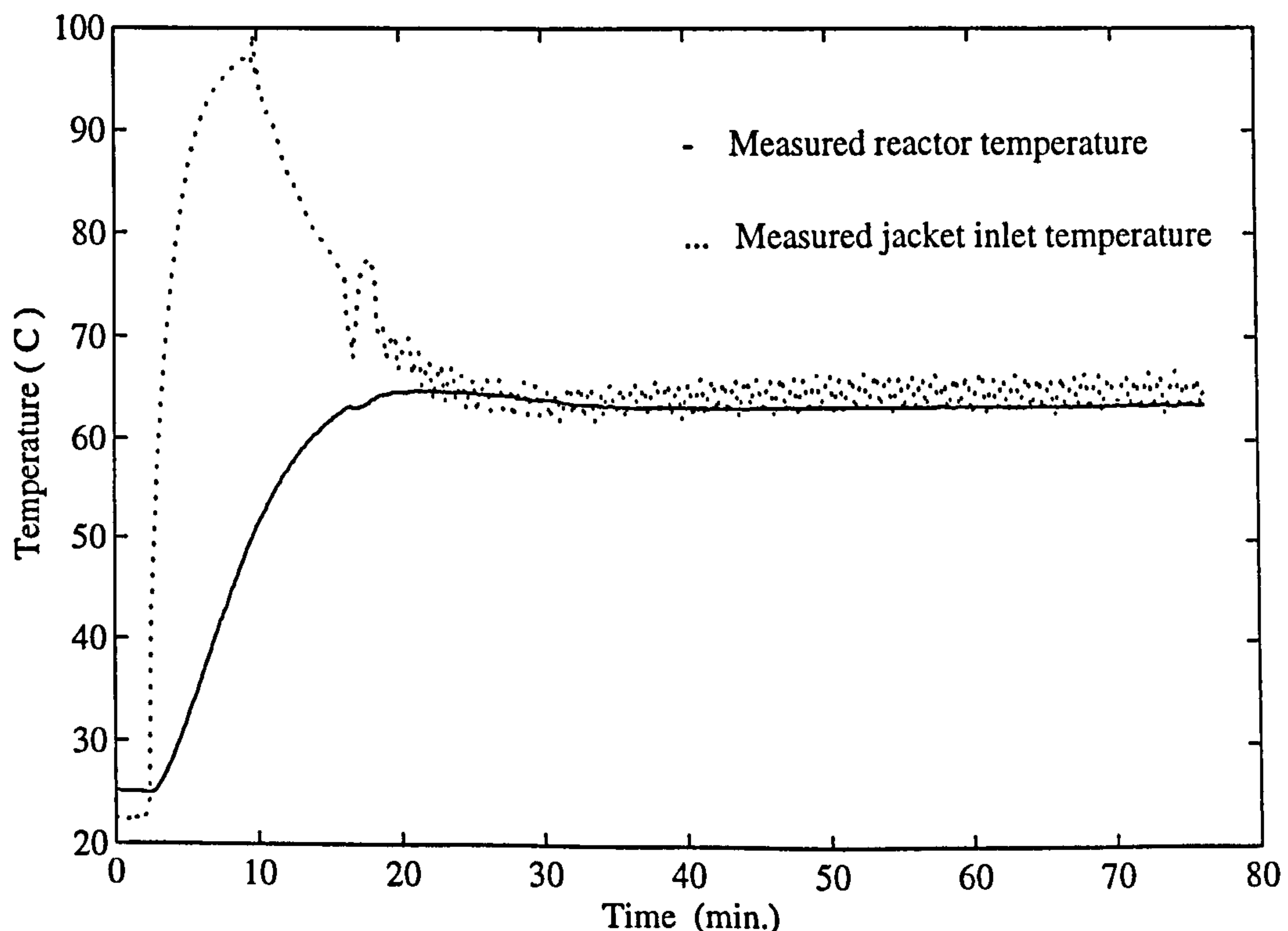


Figure 5.16 Cascaded GMC/PI controller -Water jacket control to 65 °C (experiment)

Case II(b). The reference temperature profile is as in *Case I(b)*. The reactor temperature is first brought to the gelatinization temperature setpoint (105 °C) without overshoot. After 15 minutes

of starch gelatinization, the reactor temperature is brought down to the new setpoint (85°C) with 4°C undershoot by the cascaded controllers with same parameters as in *Case II(a)*. The experimental results are shown in Figure 5.17. The performance of the controller is similar to *Case II(a)*, with RT value of 503 seconds and IAE value of 5386.

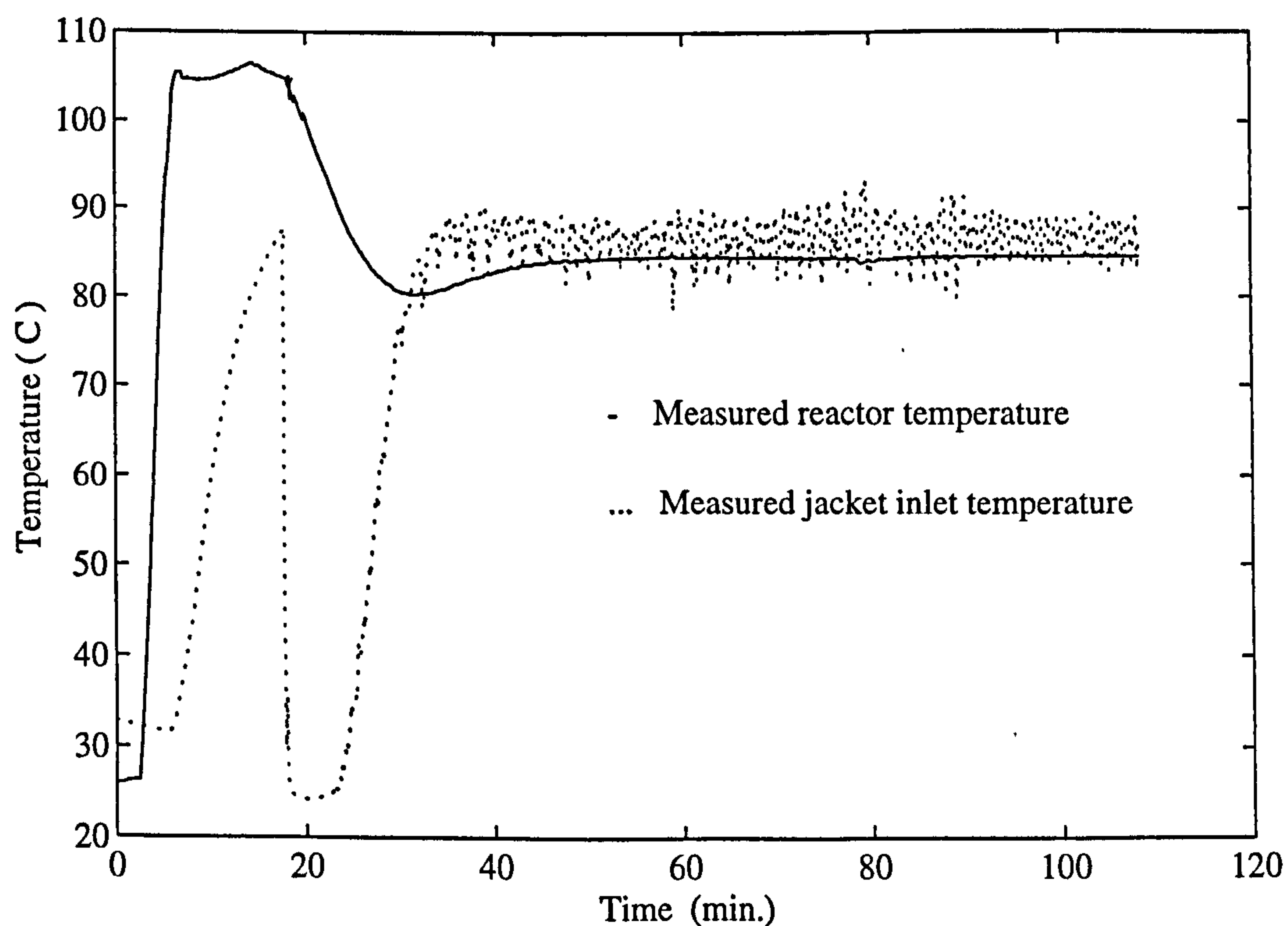


Figure 5.17 Spurge heating to 105°C under GMC controller, then water jacket control to 85°C under cascaded GMC and PI controllers (experiment with water)

Case II(c). To further test the robustness of the cascaded controllers in response to process variations (*e.g.* different temperature profiles and physical properties of the reactor contents), an experiment was performed similar to *Case II(b)*. In this case, the reactor temperature setpoint for starch degradation was 65°C and the reactor was fed with water rather than starch. This causes a mismatch of model parameters such as the heat transfer coefficients. The temperature responses are shown in Figure 5.18. The reactor temperature is again brought to the gelatinization temperature (105°C) quickly and without overshoot. The reactor is then cooled to the new temperature setpoint (65°C) with 4°C undershoot. The response is very similar to *Case II(b)*.

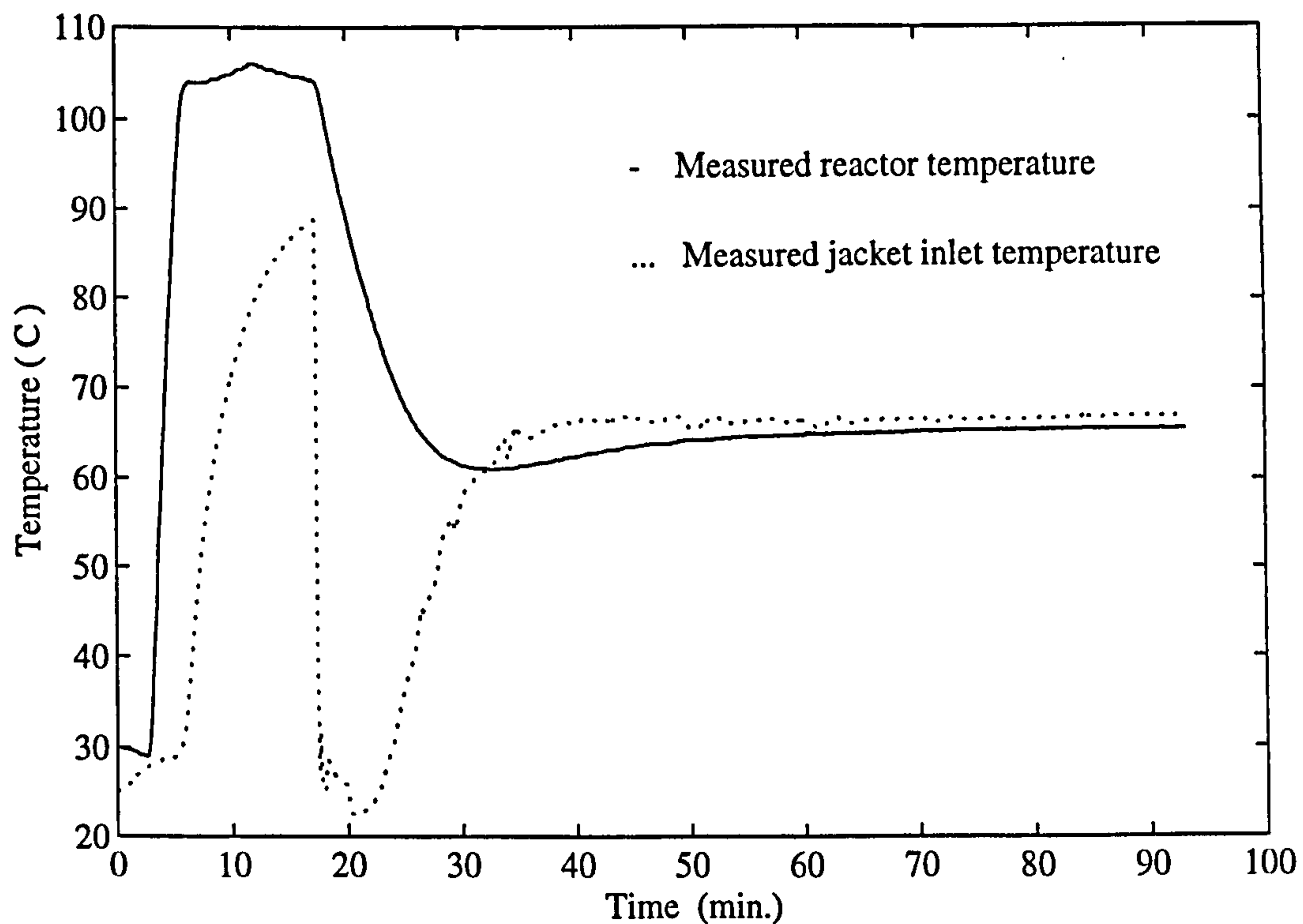


Figure 5.18 Sparge heating to 105 °C under GMC controller, then water jacket control to 65 °C under cascaded GMC and PI controllers (experiment with water)

Case II(d). The effect of the jacket temperature constraint on the controller performance is studied by simulation as experiments were deemed to be unsafe. With the constraint and the reference profile as in *Case II(b)*, the simulated reactor temperature response (Figure 5.19, continuous line) is similar to the profile in Figure 5.17. However, when the constraint is removed, the jacket temperature exceeds the 100 °C limit in order to prevent the reactor temperature from overshooting (Figure 5.19, dashed line). The simulation shows that the constraint on the jacket inlet temperature causes a small reactor temperature overshoot.

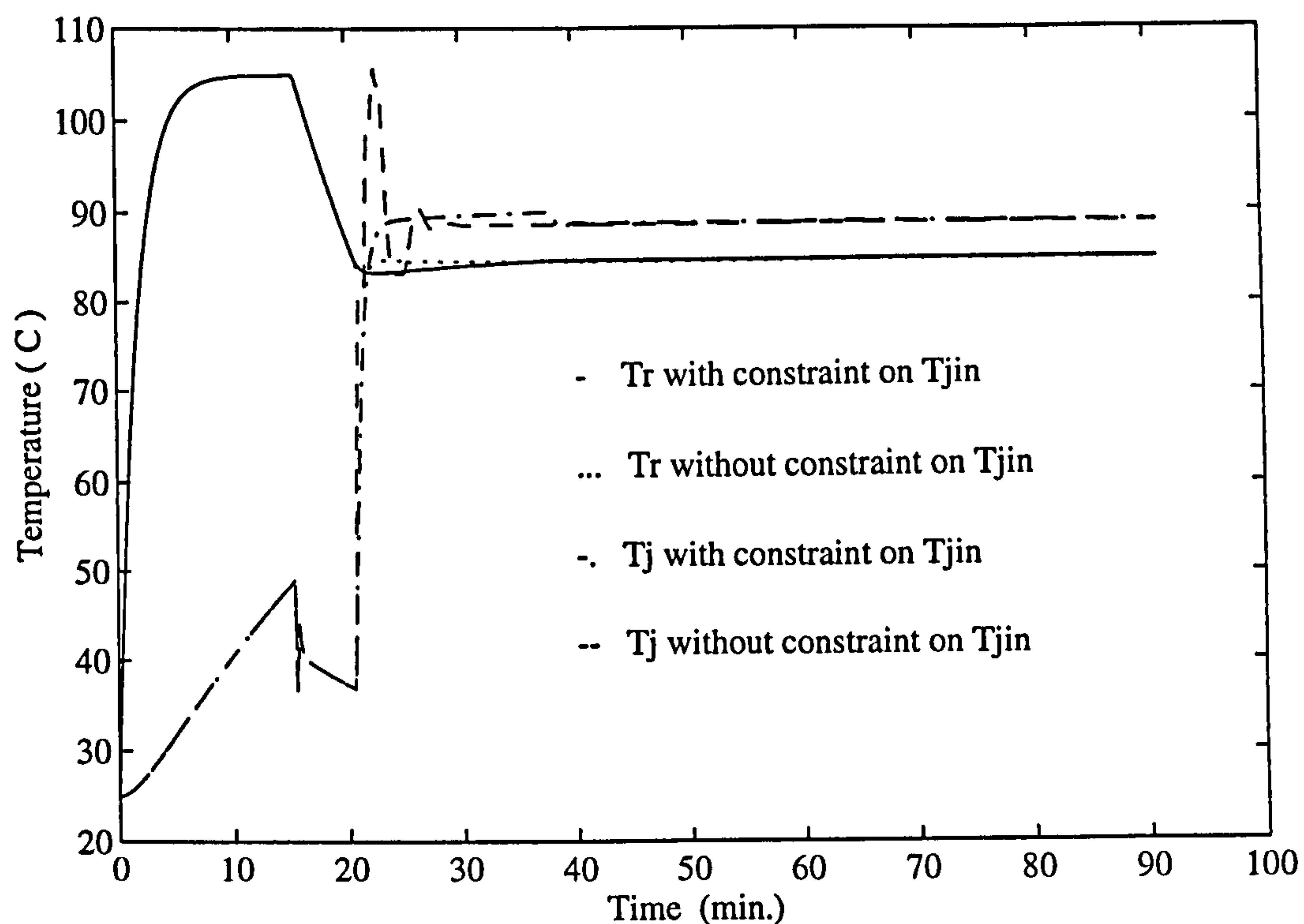


Figure 5.19 Effect of constraint on the controllers performance (simulation)

5.5.5.3 GLC and PI Control

The GMC controllers used in the previous section are replaced by GLC controllers. The robustness of the controllers with respect to measurement error is studied in the following two experiments:

Case III(a). The weight of the reactor contents (in Kg) is calculated from the load cell signal using an appropriate factor and offset (-17.3 Kg). With this offset removed, a 22% error in the reactor weight measurement was deliberately introduced. Therefore, a measurement of 80 Kg actually corresponds to a weight of 62.7 Kg. The error also causes an error in the starch concentration from the expected 6% (w/w) to an actual concentration of 7.8%, hence errors in the viscosity and heat exchange properties. This provides an excellent test of the robustness of the controllers in response to instrument error and model parameter mismatch.

Case III(b). The same experiment was conducted as in *Case III(a)* with the correct weight measurement.

The experimental reactor temperature responses for the two cases are shown in Figures 5.20 and 5.21. Figure 5.20 shows the initial sparge heating control results while the entire operation is shown in Figure 5.21. With the wrong weight measurement, the reactor temperature increased very slowly for the first 2 minutes. This is caused by the fact that the reactor temperature probe is only covered when the volume of the reactor contents is more than 70 Kg. As the actual content of the reactor was 62.8 kg, the temperature probe was actually not covered at the initial stage of the reaction. Therefore, the error in the weight measurement also introduced an additional error in the reactor temperature measurement at the initial stage. The temperature probe was covered by the reactor contents only after 2 minutes of sparge heating (due to the injected steam). Despite these severe disturbances, the responses of the reactor temperature for both cases were excellent.

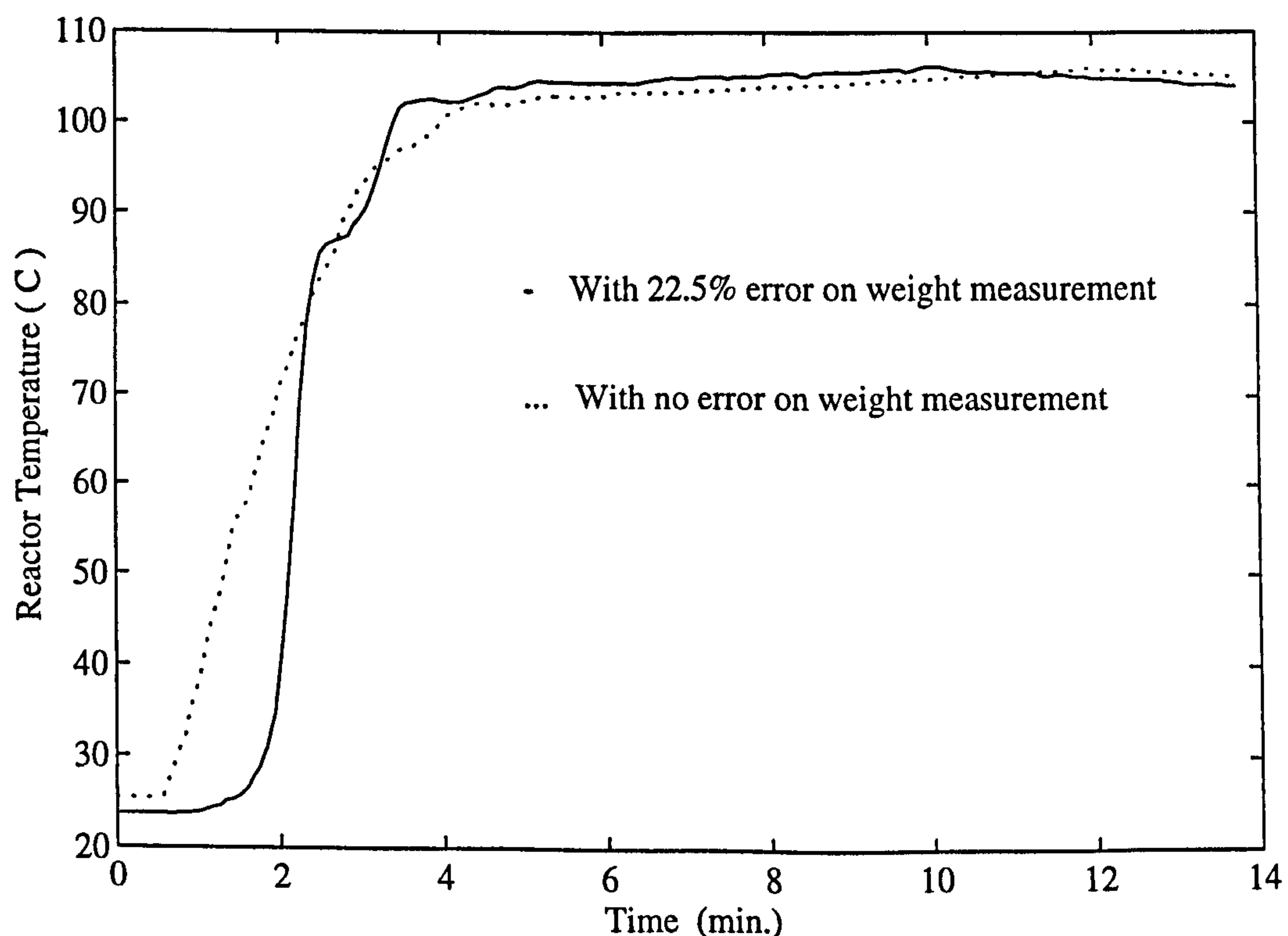


Figure 5.20 Cascaded GLC/PI controller with and without error on the measurement of reactor weight - initial sparge heating stage (experiment)

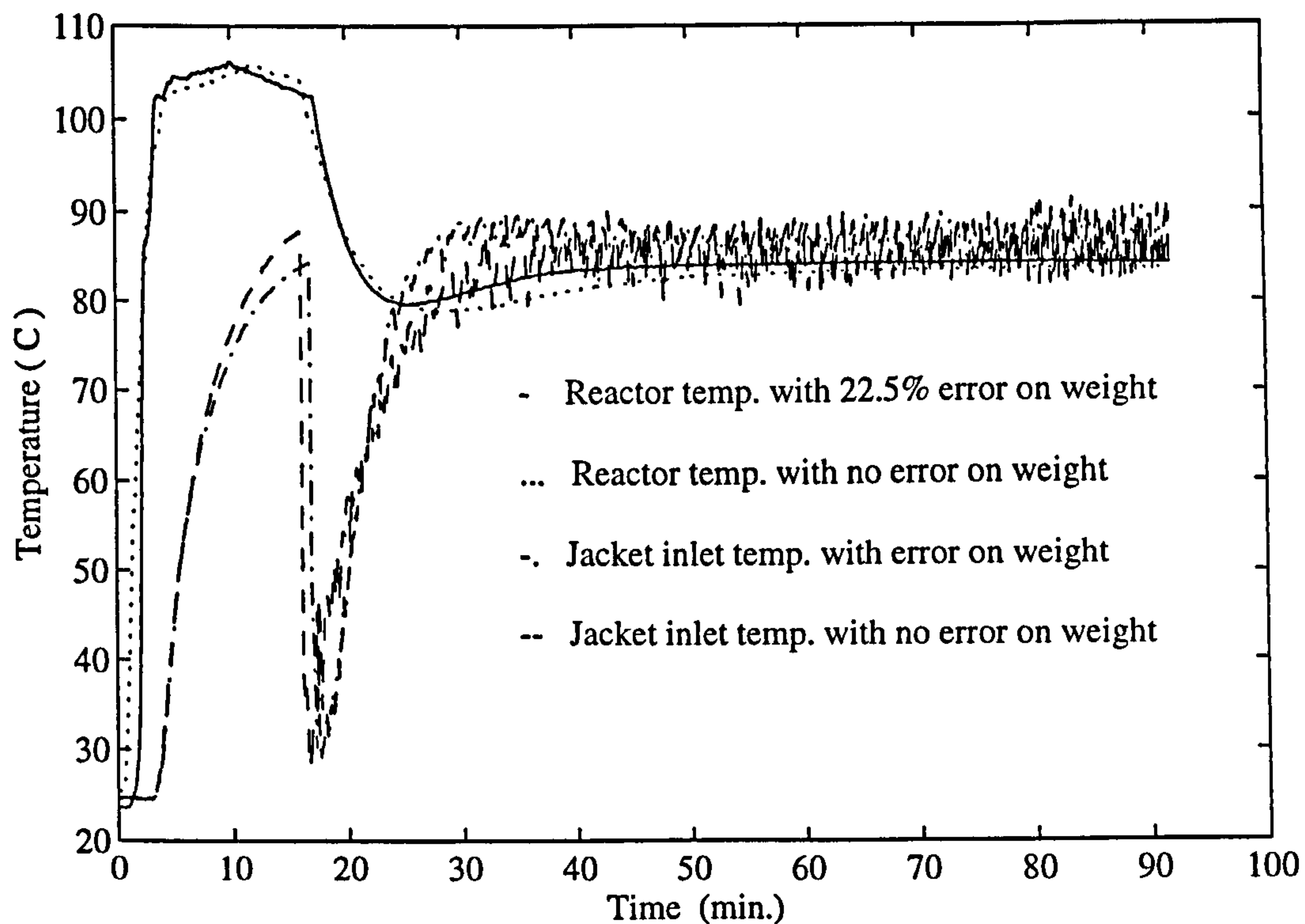


Figure 5.21 Cascade GLC and PI control with and without error on the measurement of reactor weight - entire operation (experiment)

5.5.5.4 GMC Control

In this section, the cascaded GMC control structure for the water jacket control is investigated by simulation and experiment. Two experiments were carried out. In order to get a direct comparison between simulations and experiments, the GMC controllers used to drive the simulation model and the plant are strictly identical. No measurement noise is introduced in the simulations.

Case IV(a). Similar to *Case III(c)*, however the reactor setpoint was changed to 55 °C under water jacket control to test the GMC controllers over a wide range of operating conditions. The temperature responses are shown in Figure 5.22. During sparge heating, the reactor temperature is brought to the setpoint quickly and without overshoot. When control is switched to the water jacket, the reactor temperature is again brought to the new setpoint without overshoot. The reactor temperature is well maintained at the new setpoint with a small IAE (391.4).

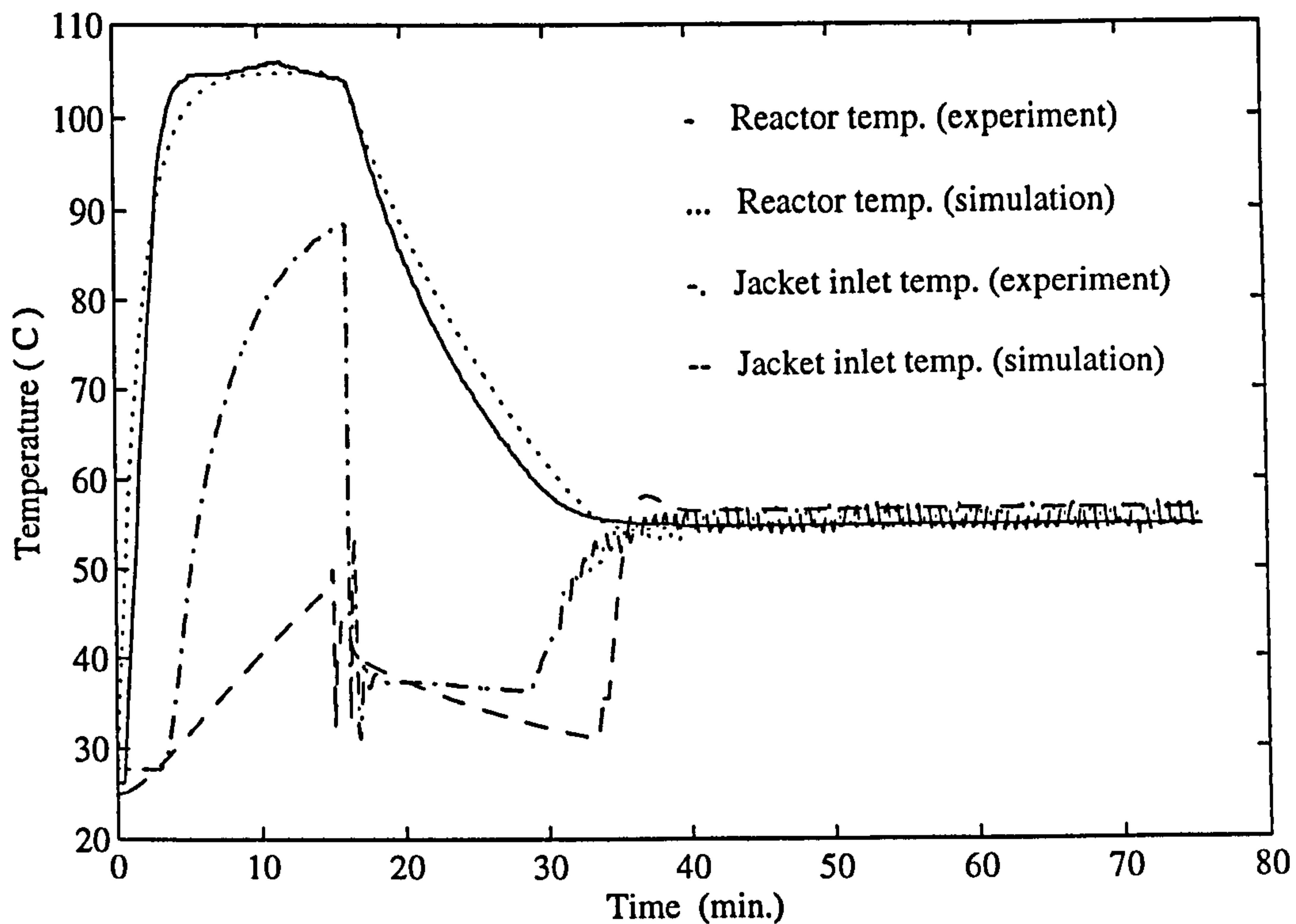


Figure 5.22 Cascaded GMC controller - sparge heating to 105 °C, followed by water jacket control to 55 °C (experiment and simulation)

Case IV(b). The reference temperature profile is as in *Case II(b)* and *Case III(b)*. With the same GMC controllers and tuning parameters as *Case IV(a)*, very similar response was obtained as shown in Figure 5.23. However, the jacket inlet temperature is more noisy.

Figures 5.22 and 5.23 show that cascaded GMC controllers performed better than the previous control configurations in the presence of the jacket temperature constraints. However, the response of cascaded GMC controllers is slightly slower than observed in Case II and Case III. The results also show a good agreement between simulation and experiment.

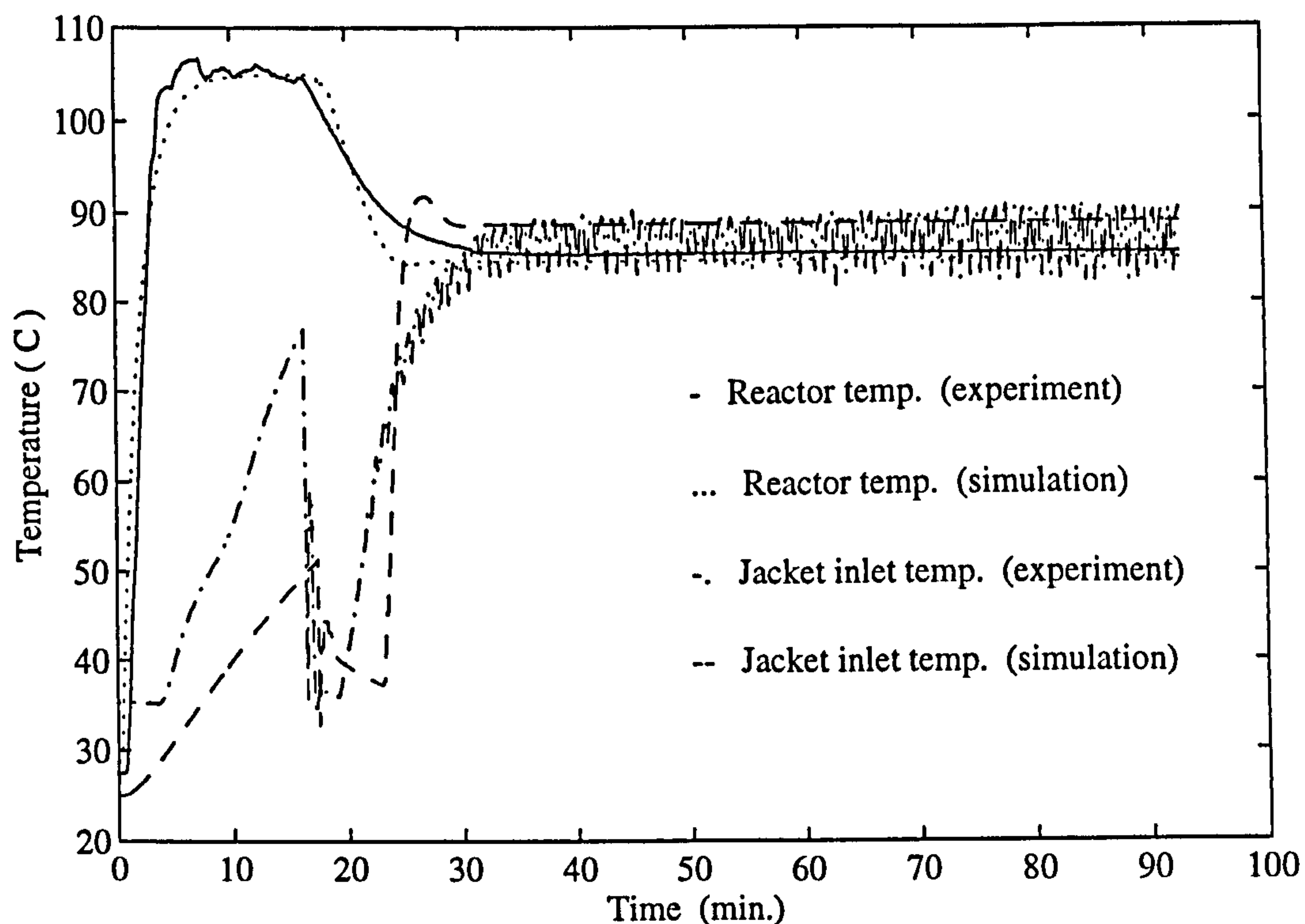


Figure 5.23 Cascaded GMC controller - sparge heating to 105 °C, followed by water jacket control to 85 °C (experiment and simulation)

5.5.6 Summary and Discussions

A dynamic model of a multipurpose batch reactor system was presented and the characteristics of the system were analysed, showing that some special control difficulties arise. Several control configurations were developed for the temperature control of the reactor using the partitioning space design approach and nonlinear control techniques. Temperature control of the reactor under each control configuration was investigated by simulation and experiment, and the robustness of the controllers was evaluated. From the simulation and experimental results, the following conclusions are drawn:

1. PI controllers perform poorly due to the nonlinearities and should be retuned for every different temperature profile and process condition;
2. The performance of GMC and GLC controllers is far superior to that of the PI controllers with a single set of tuning parameters. Table 5.6 summarises Rise-Times and IAE values for

each experiment. In all cases with the same reactor temperature reference profiles, the IAE values using GMC and GLC controllers were much less than those from PI controllers.

CASE		Sparge Heating		Water Jacket Control	
		RT (Sec)	IAE	RT (Sec)	IAE
Case I	(a)	-	-	788	3832.0
	(b)	195	4415.4	300	9756.6
Case II	(a)	-	-	835	2021.0
	(b)	186	331.8	503	5386.2
	(c)	190	350.9	405	5310.8
Case III	(a)	210	510.5	180	6929.0
	(b)	250	663.8	240	7882.0
Case IV	(a)	215	400.5	900	391.4
	(b)	185	507.7	890	1274.7

Table 5.6 Quantitative performance of controllers under each of control configuration of Table 5.4.

3. The response of the GMC and GLC controllers is identical for the reactor system studied. For systems of relative degree one, the GMC control method can be viewed as a special case of the GLC control approach.
4. Cascaded GMC and GLC controllers with a PI or GMC controller in the secondary loop provide robust control in the presence of severe model mismatch. This study only considered a single error in the weight measurement. The effect of model mismatch in other parameters, such as heat transfer coefficients and density of the reactant, on the controller performance is similar to a mismatch in the reactor weight. This can be seen from the controller algorithms in Table 5.3. In fact, the mismatch in weight measurement provides a very severe test, as discussed in section 5.5.5.3.
5. Cascaded GMC and GLC controllers with a PI or GMC controller in the secondary loop provide a consistent response over a very wide range of operating conditions, despite the presence of variable control structures, nonlinear jacket recirculating flow characteristics and jacket inlet temperature constraint. On the other hand, the PI controllers should be retuned for optimum performance at every operating point.
6. The presence of constraint on the jacket inlet temperature degrades the performance of the controllers in all cases. Although the cascaded GMC controllers controls the reactor

temperature without overshoot, this is balanced by sacrificing somewhat the speed of control response.

7. The jacket inlet temperature is noisy in most cases. This is largely caused by the jacket flow dynamics and the characteristics of two non-return valves. A low frequency filter could be used on the jacket inlet temperature measurement before it is used by the controllers to improve the inner loop response. However, as this does not degrade the reactor temperature response, this was not attempted in this study.
8. The implementation of the GMC/GLC controller cascaded to a PI controller is simple and straightforward. However, implementation of the cascaded GMC controllers requires an accurate model of the control valves. Furthermore, the tuning of cascaded nonlinear controllers is much easier than tuning PI controllers.

Simulation and experimental results show that the cascaded control structure resulting from partitioning in combination with nonlinear control techniques can be effectively applied to a class of nonlinear systems. A consistent, robust and better control performance over a wide range of operating conditions is achieved.

5.6 Conclusions

A partitioning space approach for nonlinear processes in combination with various controller design techniques has been described in this Chapter and demonstrated for the temperature control of a multipurpose batch reactor. The cascaded structure using conventional PID controllers is widely used in industry. However, the present work provides a theoretical justification for controller decomposition especially for nonlinear controllers. Two criteria have also been presented which permit the easy screening of candidate variables as intermediate outputs.

The simulation and experiment results show that the partitioning space approach provides an effective design procedure and has the following advantages:

1. It is amenable to various controller design techniques for nonlinear systems and results in a more flexible control system structure in which different controllers can be used for each subsystem.

2. Most model based control techniques require a good model of the system. In practice, obtaining a good model for the entire system may be difficult. However, a well defined model for part of the system may be available. In such cases, the partitioning space approach may be applied to make the best use of the well modelled part of the system for the controller design. Model based controllers often provide better control when the model is relatively well defined. For systems with uncertainty on the model, the performance of model based controllers is case dependent.
3. By partitioning the nonlinear system into two subsystems and by introducing the intermediate control input, the relative degree of each subsystem is reduced to less than that of the original system. This has the following advantages:
 - i) less computation is required for calculating the Lie derivatives and Lie brackets when differential geometry theory is used. This results in a control algorithm that is simpler and easier to implement;
 - ii) the disturbance in SysI is guaranteed to be decoupled if it can be measured. For the disturbance in SysII, the disturbance matching condition is always easier to satisfy. This will again widen the application of the GLC control technique, as demonstrated by the reactor temperature control example;
 - iii) the relative degree of SysI with respect to the new control input is always 1. Therefore, the GMC controller design technique can always be directly applied.
4. As demonstrated by the reactor temperature control example, constraints may be imposed on an intermediate state variable. Violating these constraints may endanger the plant, even though the input-output response is stable and bounded. In the partitioning space approach, by selecting the constrained state variable as the intermediate control input, and provided conditions Eq (5.c4) and Eq (5.c5) are satisfied, the constraint can be specified and considered in the controller design.
5. For systems of order greater than the relative degree, the input-output linearization techniques do not guarantee the stability of internal dynamics of the system. For a system with unstable zero dynamics (non-minimum phase system), the unstable internal dynamics present a major obstacle to the use of inverse based control, such as GLC. It may be possible that the partitioning space approach be used for a class of non-minimum phase systems by selecting the unstable state as the intermediate control input. This issue needs further investigation.

The partitioning space approach may only be effectively applied to systems which have some fast closed-loop poles. These poles should correspond to the dynamics of state variables which can be used as the intermediate control input. For systems which have only slow closed-loop poles of the same order, better control performance can not be guaranteed.

Chapter 6

Sequence Control of Batch Processes

Sequence control plays a key role in batch process automation in ensuring efficient and safe plant operation, and consistent product quality. However, the development of control sequences is a highly labour intensive and error-prone task. Incomplete knowledge and understanding of the process often is the key factor in leading to ill-defined, incomplete and inconsistent functional specification. A structured approach to specification of the sequential control logic and implementation into control code is typically used in practice. A large amount of judgement has often to be used when deciding both upon the sequence logic and the choice of operating parameters (*e.g.* response times). These parameters and logical choices may have a considerable effect on the resulting procedures, thus on the efficiency of the plant operation. However, they are often poorly defined at the stage of initial plant design and subject to uncertainties. A quantitative study of the effects of these sequences on the dynamic behaviour of the system is difficult to carry out using standard method.

In this chapter, a dynamic model and a simulation based approach are used to assist in the initial development of functional specification of sequence controllers, and in the subsequent validation of the resulting control sequences. The purpose of this chapter is to show that the process of developing and testing new sequential controllers is easily and efficiently carried out within the integrated operations support system developed in this work and this ease of development can lead to substantially improved procedures. To this purpose, the integrated development environment described in Chapter 3 is utilised. The dynamic simulation of the plant and controllers is carried out using gPROMS, a combined discrete/continuous simulator (Barton and Pantelides, 1991), using the detailed dynamic model of plant and control sequences

described in Chapter 4. Two typical problems are examined, involving the development of procedures for:

- sequence control of a cleaning-in-place (CIP) operation;
- emergency recovery during temperature control of the batch reactor.

6.1 Sequence Control of Cleaning-In-Place Operations

As already discussed, cleaning-in-place (CIP) systems play an important role in the food processing industry to ensure plant hygiene and product quality. However, in many plants, CIP cycles take a large proportion of the available production time. Therefore, reducing CIP operating times is an important issue. The main requirement of a CIP system is that reliable, reproducible microbiological results be achieved in a minimum cleaning time. The latter depends not only on the physical (cleaning solution flow rate, turbulence, *etc.*), chemical (type of detergent/agent, concentration) and thermal (heat capacities, temperature) characteristics of the cleaning solution, but also on the operating procedures under which the cleaning operations are performed. It is typically difficult to monitor microbiological results on-line due to lack of suitable sensors. Reliable and repeatable cleaning is therefore achieved by closely controlling the CIP operations to follow as close as possible designated procedures and operating conditions, developed based on experience and then tuned and validated during commissioning.

6.1.1 Cleaning Based On Fixed Time Policy

In the batch pilot plant, there is one CIP station with five cleaning circuits, one for each major process vessel and associated pipework, with a single high capacity CIP pump feeding water or detergent fluid to each circuit at fixed rate. Vessel cleaning is achieved by burst spraying the fluids at high pressure through a spray-ball. Intermittent action is obtained by switching on/off the CIP pump. Individual smaller capacity pumps return the detergent from each vessel to the CIP unit, and the cleaning water to drains. The five cleaning circuits share some common pipework and the design capacity of the CIP station is such that only one circuit at a time can be operated.

Cleaning of each vessel consists of three phases which are carried out in series:

- pre-rinse with soft water;
- detergent clean;
- final rinse with soft water.

The first and third cleaning phases utilise the same process route (from soft water supply, through the vessel and out to drain). The detergent clean phase utilises a different closed circuit route (from the CIP tank, through the vessel, and back to the CIP tank). Each route involves a sometime complex path through pipework and valve manifolds. Before the detergent clean phase is started, both temperature and concentration of the detergent in the CIP tank are checked. In the existing control scheme, a CIP service sequence is executed to bring the temperature and detergent concentration to desired set points. This is achieved by circulating the CIP fluid through a service heat exchanger (HE3), and by a dosing system, respectively. Detailed functional specifications for cleaning each vessel were originally written in structured English during the initial automation system design. For example, the specification for cleaning vessel T1 is represented in the form of flowchart in the first and second columns of Table 6.1.

The functional specification provided a base for developing detailed control sequences. However, a number of questions remain to be answered, as shown in the third column of Table 6.1. These questions are identified in the course of the software development and resolved by the individual engineer. This is inherently problematic and often is a major cause of poor documentation, inconsistency and errors. Difficulties in determining detailed parameters and in evaluating the effect of an operating procedure on the cleaning time are demonstrated by a simulation and were experienced during the plant commissioning.

Original Cleaning Sequence. The initial cleaning sequences supplied with the plant included fixed times for each action, to be tuned by the suppliers during commissioning. Level control in the vessels being cleaned proved rather difficult, due to the difference in capacity between CIP pump (inlet) and drain pumps (outlet), the intermittent flow from the CIP pump, the drain rate being dependent on the hydraulic head in the vessel and the different length (*i.e.* hold-up) of the five cleaning circuits (see Figure 3.2 and Figure 6.4). A large number of commissioning trials were needed to adjust the individual time parameters so as to avoid either flooding the vessel being cleaned or the drain pumps running dry. The cleaning time for each phase was adjusted during commissioning, with 5 minutes pre-rinse, 10-minutes detergent clean and 5 minutes final rinse for all vessels.

Flowchart	Further description	Problems/questions
<pre>graph TD Start([Start]) --> Inhibit[Inhibit sequences] Inhibit --> Emergency[Emergency actions] Emergency --> CheckRoute[Check Route] CheckRoute --> RouteError{Route error?} RouteError -- Yes --> Alarm[Alarm] Alarm --> CheckRoute RouteError -- No --> OpenRoute[Open route] OpenRoute --> PreRinse[Pre-rinse with soft water] PreRinse --> CallCIP[Call CIP service] CallCIP --> WaitSetpoints[Wait until Setpoints reached] WaitSetpoints --> DetergentClean[Detergent clean] DetergentClean --> FinalRinse[Final rinse with soft water] FinalRinse --> END([END])</pre>	Sequence ID All other clean and CIP service sequences, Fill T3 from T1/2, Fill T4/5 from T1/2, Fill T1/2 from T4/5, Fill T1/2. Stop P6, P10, P1, close AV1-4. SSV1-1/2/4/5, AV1-42/24/20/4, PSI-1, DDV1-1/3/6. All valves closed ? Open SSV1-3, AV1-41/40/47, DDV1-8. Pre-rinse time : 5 min. <pre>graph TD StartTimer[Start timer] --> InitialCharge[Initial charge-up] InitialCharge --> StartP1[Start P1 at max. speed] StartP1 --> FlowMonitor[Flow monitor] FlowMonitor --> StopP6[Stop P6 for 40 sec.] StopP6 --> RestartP6[Re-start P6 and run for 27 sec.] RestartP6 --> TimerOut{Timer out?} TimerOut -- No --> StopP6 TimerOut -- Yes --> ToDetergentClean[To detergent clean phase]</pre> Separate function specification Temp. in CIP tank : 78 °C Detergent conductivity: 300 µS Detergent clean time: 15 min. Others same as Pre-rinse. Same as Pre-rinse	Can be generated automatically How to determine the items in the emergency list? What are the effects of those emergency actions? Tank status (level, clean or dirty) should be checked. What is the alarm level (stop the sequence or hold?) How to determine the time? How much or how long for this step? Is the time correct? What to do with the route valves? How to determine the setpoints? Do we need to check both conditions? What to do with the temperature control loop when P6 is stopped?

Table 6.1 Functional specification of the control sequence for cleaning Tank 1.

Figure 6.1 shows a dynamic simulation of these control sequences, as developed based on the provided functional specification, and originally implemented in the plant for cleaning T1 (after commissioning and tuning). The result shows that, in spite of the efforts, the vessel is being slowly filled during the detergent clean phase and is 1/3 full by the end of this phase. This limits the effectiveness of the burst cleaning action (only the uncovered portion of the inside walls are cleaned). Also, frequent start/stop results in a high maintenance requirement for the CIP pump and associated valves, and may reduce their life.

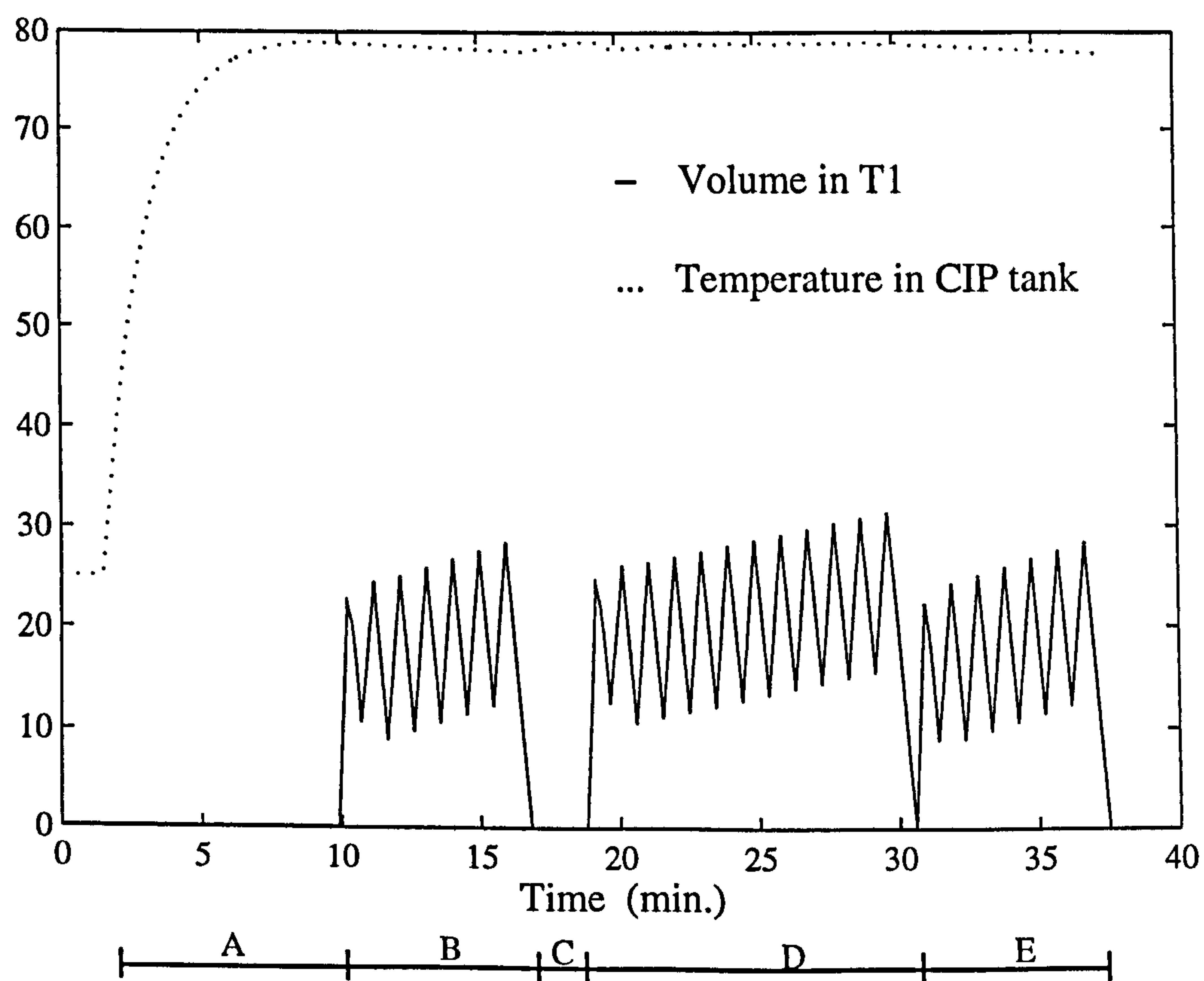


Figure 6.1 T1 cleaning simulation based on fixed-time policy. A - initial CIP service; B - pre-rinse; C - CIP service; D - detergent clean; E - final rinse.

Figure 6.1 also shows that it takes about 10 minutes in order to bring the detergent temperature in the CIP tank and the concentration of the detergent to the desired setpoints by the CIP service sequence. The detergent clean phase is not started immediately after the pre-rinse with soft water. This is due to the fact that the detergent temperature in the CIP tank (T7) drops below the setpoint during the pre-rinse (due to the heat transfers with the outside) and it must be brought back to the desired setpoint by the CIP service sequence before the detergent

clean phase can be started. However, it was observed that the detergent fluid is not directly discharged to the target vessel from the CIP tank but through the service heat exchanger. Therefore, the condition that the detergent temperature in the CIP tank must be at setpoint could be eliminated from being a pre-condition for the CIP service or as a pre-condition for starting the detergent clean, providing that the detergent stream outlet temperature could be controlled to the desired value (80°C) during the detergent clean phase itself.

Figure 6.2 shows the temperature response of the detergent in the CIP tank during the CIP service. The result shows that the outlet temperature of HE3 in the detergent stream is brought to the setpoint very quickly, and maintained at the setpoint despite of continuous change in the HE3 inlet detergent temperature. On the other hand, it takes more than 6 minutes to bring the temperature of detergent in the CIP tank to the setpoint value due to the large holdup of this tank. This results in a rather long CIP service time, delay in start of detergent clean phase as shown in Figure 6.1, and leads to a long total cleaning time.

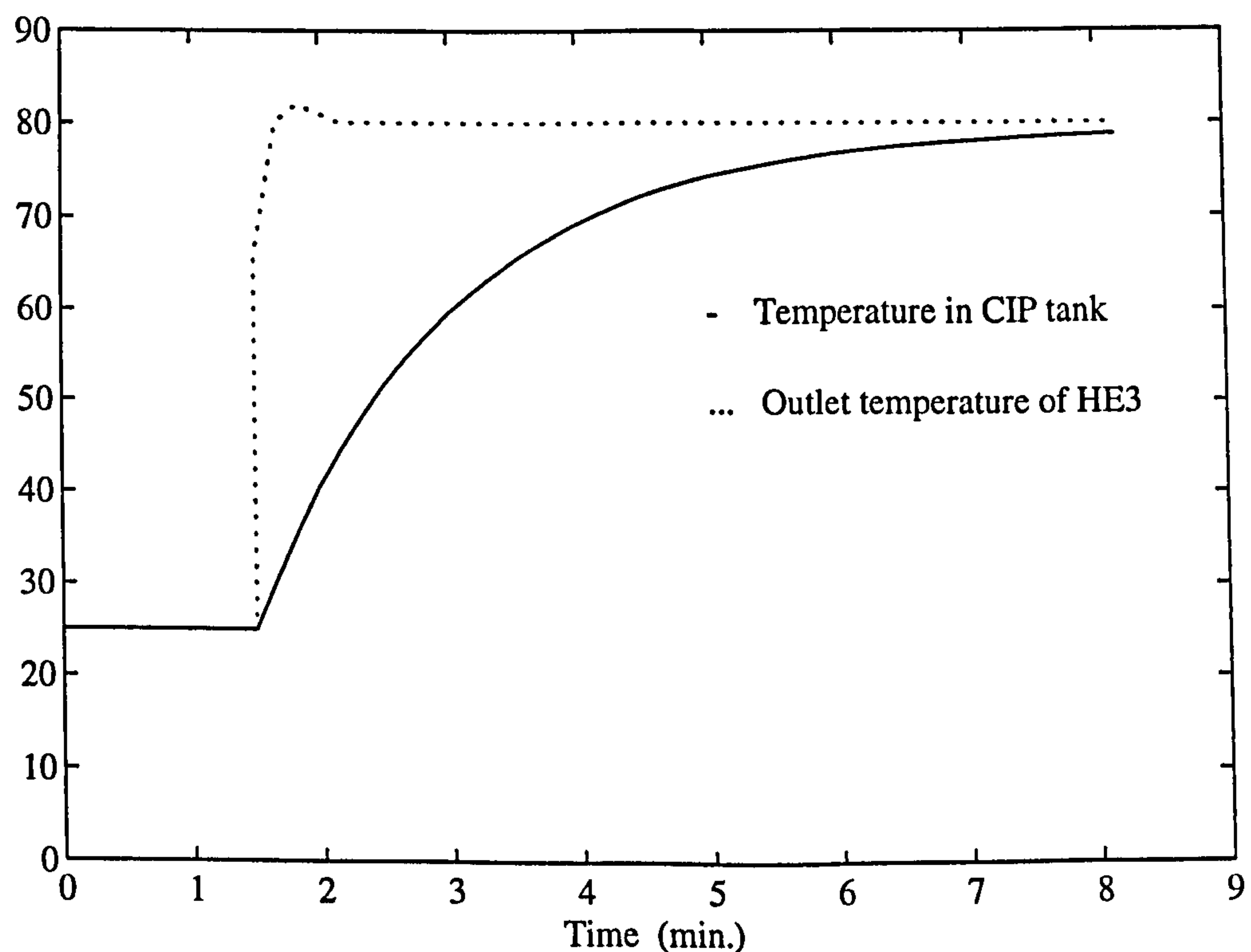


Figure 6.2 Detergent temperature response during the CIP service

6.1.2 Cleaning Policy Based on Volume

The simulation results and implementation during the plant commissioning of the cleaning control sequence in the previous section have shown that a proper definition of detailed operating procedures and selection of tuning parameters is a tedious, time-consuming task. On the other hand, these parameters have a considerable influence on the plant operations and overall cleaning times. As an alternative to the fixed time termination condition for the each cleaning phase, a very simple volume based cleaning policy has been proposed. This was developed and tested first on the simulation, then executed on the pilot plant. The basic idea is to control the CIP feed pump based on the volume/level of the cleaning solution in the vessels so as to prevent undesired accumulation/deflation of fluid in the tank utilising an available continuous measurement of the level in the tank. Only two parameters need to be determined: V_h and V_l , the high and low limits, respectively, for the volume of cleaning solution accumulated in the tank during cleaning. When the cleaning solution volume/level in the tank exceeds V_h , the CIP pump is shut down and whenever the solution volume/level in the tank is less than V_l , the pump will be restarted. To implement this policy in general, a level measurement is required in the vessel. Alternatively, a good estimate of the holdup in the tank can be calculated from inlet and outlet flow rate measurements. Furthermore, the CIP service sequence is performed at beginning, with the only goal of bring the detergent concentration to the desired setpoint. The detergent temperature is now controlled by means of HE3 during the detergent clean phase itself. A dynamic simulation of the cleaning of tank T1 based on this new policy with V_h of 20 l and V_l of 6 l is shown in Figure 6.3.

Figure 6.3 shows that the initial detergent service time is reduced to less than 3 minutes, and the detergent clean phase is started immediately after the pre-rinse (due to the removal of the pre-check on the detergent temperature in the CIP tank). The temperature of the cleaning fluid is controlled effectively at the desired set point. The further advantages of this new policy, as first demonstrated by simulation, are:

- the cleaning solution level/volume in the tank can be controlled directly and strictly between V_l and V_h . Therefore, accumulation of cleaning solution in the tank and emptying of the tank are avoided;
- a lower average level of fluid in the tank being cleaned can be safely achieved. Thus, the cleaning effectiveness is improved by the mechanical effects of burst rinsing and oscillation of cleaning solution in the vessel;

- it is very easy to implement and tune;
- any dynamic effects of changes in the inlet and outlet flows on levels are inherently taken into account.

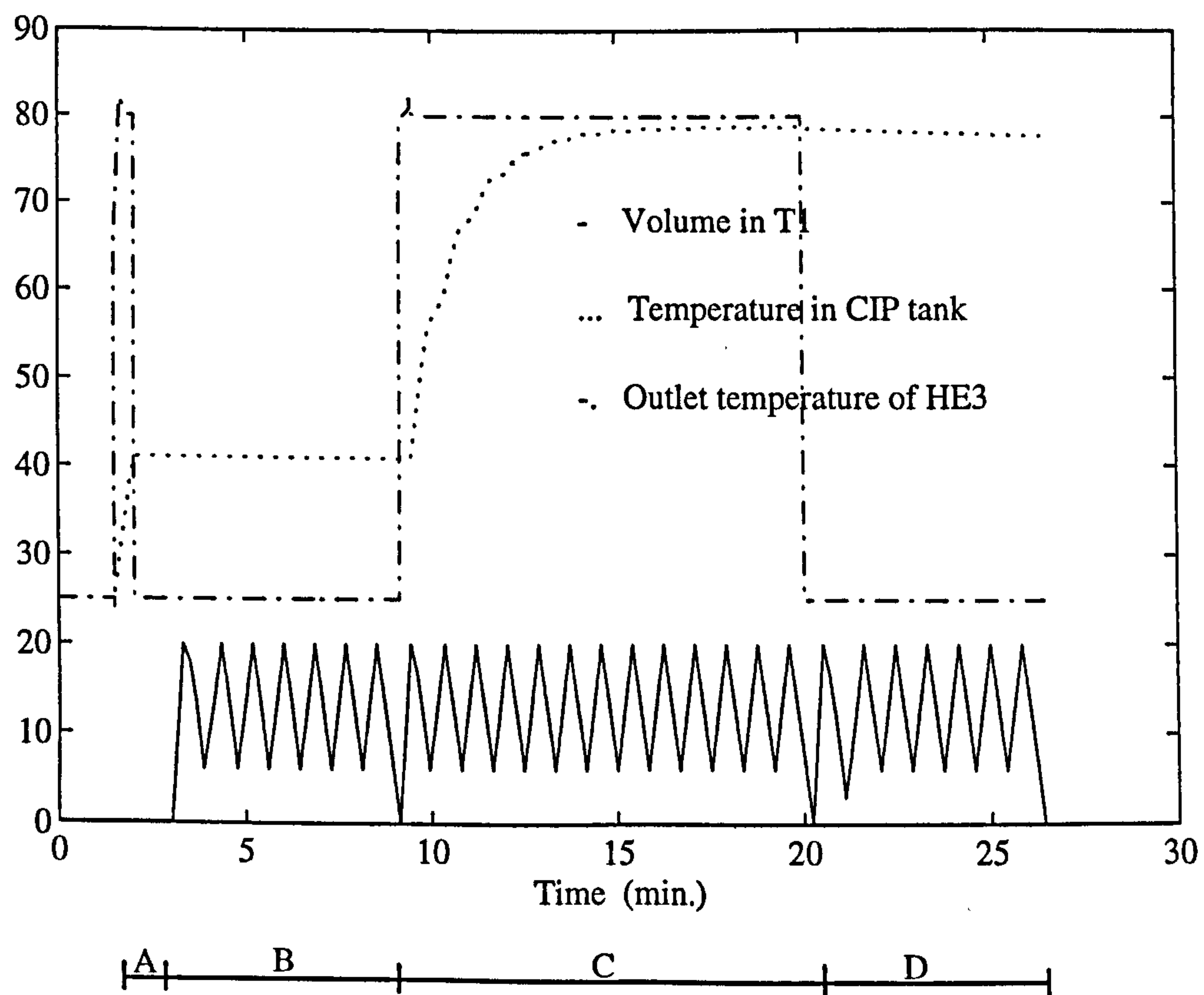


Figure 6.3 T1 cleaning simulation based on volume policy. A - initial CIP service; B - pre-rinse; C - detergent clean; D - final rinse.

Compared to the original control sequence, 31% of the original cleaning time is saved for a single vessel with less energy consumption and more effective cleaning. The volume based policy and improved CIP service control, first demonstrated on the simulation, were implemented in the pilot plant for all five cleaning sequences with very good results.

6.1.3 Cleaning of Parallel Vessels

Parallel operations are very common in food and pharmaceutical industries. If two vessels in parallel are free at the same time, a simultaneous cleaning of those vessels would mean a considerable time saving. The key question explored here is whether we can make use of the existing CIP system, without reducing the flow rate and turbulence of the cleaning solution delivered to each vessel so that the cleaning efficiency can be maintained. To achieve an effective flow turbulence, the CIP pump is normally designed with a much higher capacity than the capacity of process pumps (*e.g.* the CIP pump capacity in the pilot plant is more than twice the maximum capacity of pumps P1/P2 of discharging tanks T1/T2). Therefore, for cleaning a single vessel, the CIP pump must be stopped frequently to avoid accumulation of cleaning solution in the vessel, as was the case for example in Figures 6.1 and 6.3. We investigated the possibility of running the CIP pump continuously but of directing the cleaning solution flow to parallel vessels alternately, thus achieving a parallel cleaning.

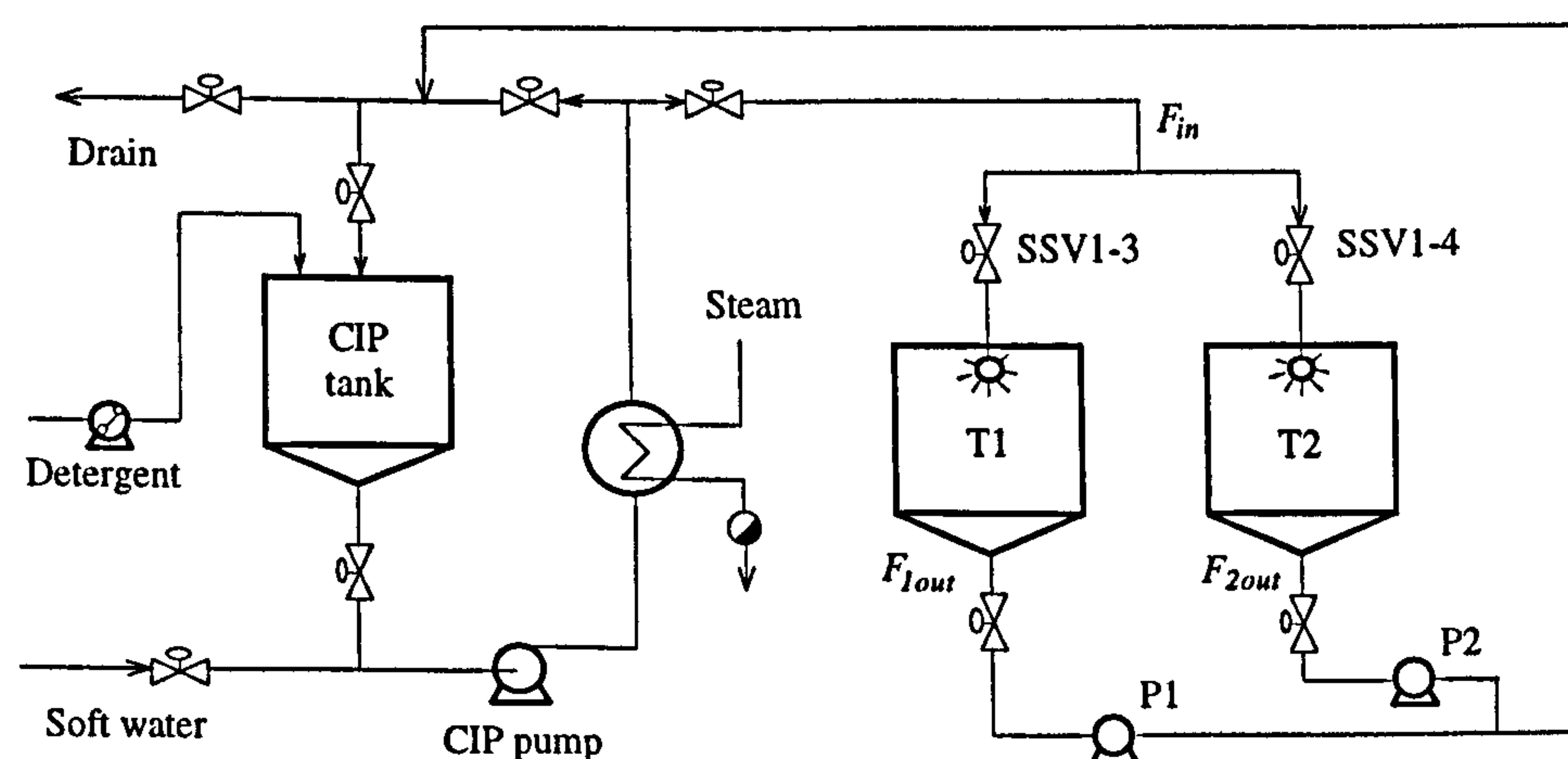


Figure 6.4 Cleaning circuits for T1 and T2 in the pilot plant.

Consider the parallel cleaning of two vessels with a single CIP pump (Figure 6.4). We wish to ensure the volume of the cleaning solution in each vessel is controlled within high (V_h) and low (V_l) limits (the same for both vessels), with a constant switching frequency of inlet valves (SSV1-3 and SSV1-4 in Figure 6.4), and that no shut down of the CIP pump. The following steady state conditions must therefore be satisfied.

Given the desired V_h and V_l , the flow rate of the CIP pump F_{in} , one of the continuous outlet flow rates from the cleaned vessel, F_{1out} or F_{2out} , the outlet flow rate of the other vessel is simply determined by Eq (6.1), and the initial volume of the cleaning solution to be charged into the first served vessel (V_{10}) is determined by Eq (6.2).

$$F_{in} = F_{1out} + F_{2out} \quad (6.1)$$

$$V_{10} = \frac{F_{1out}}{F_{in}} V_h + V_l \quad (6.2)$$

$$F_{1out}, F_{2out} \in (0, F_{in}), V_h, V_l, V_{10} \in (0, \min(V_{1max}, V_{2max}))$$

Figure 6.5 shows a simulation result for the parallel cleaning of tank 1 and tank 2 in the batch pilot plant when the two conditions are satisfied. The volume of the cleaning solution in each tank is controlled in the range of 6 to 20 l, according to the volume based policy described in the previous section. Tank 1 is fed first with an initial fluid charge of 13 l.

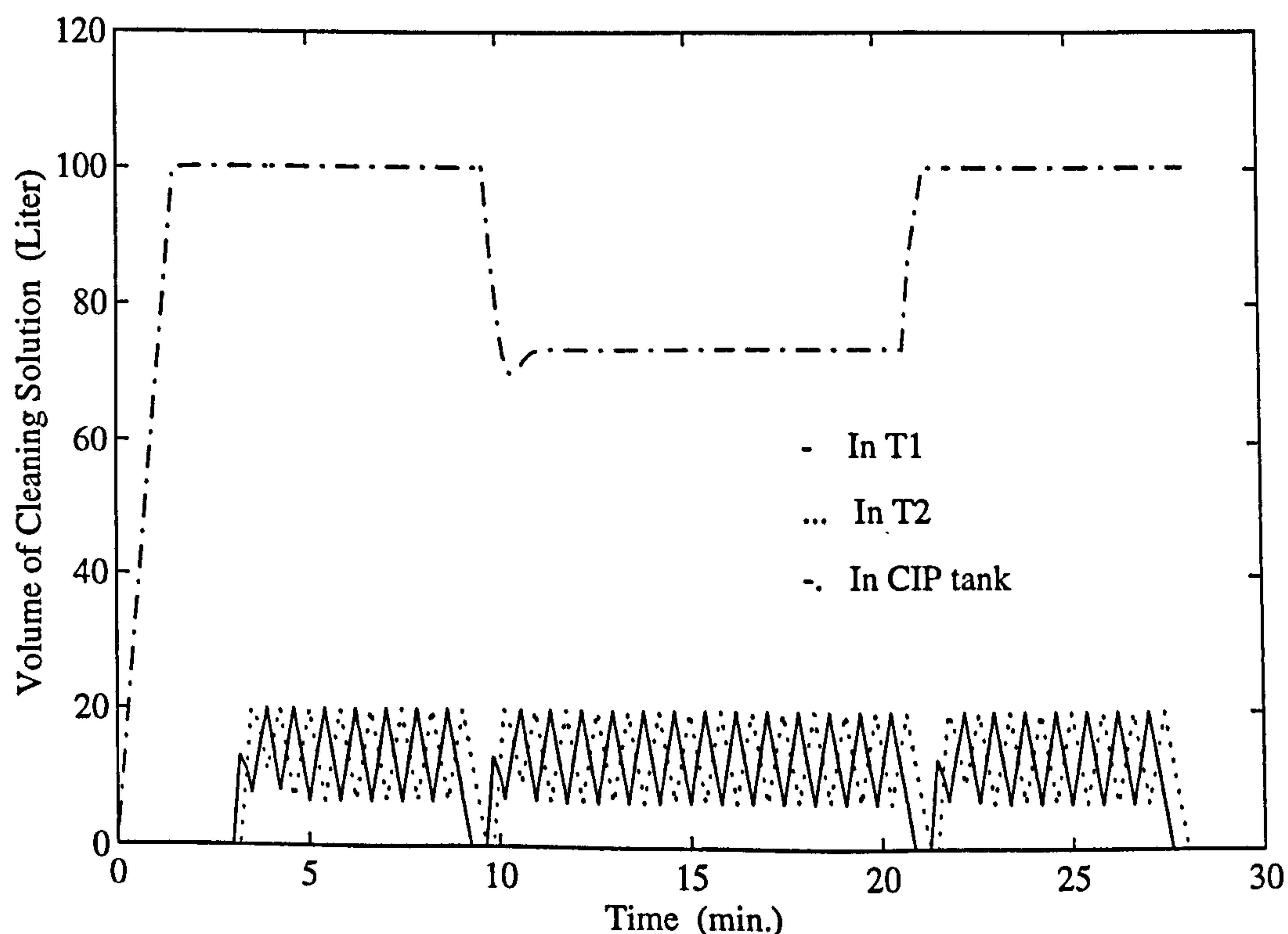


Figure 6.5 Simulation of parallel cleaning of T1 and T2.

However, the first condition is rarely satisfied in practice due to the effect of the pump dynamics, and very often $F_{in} > F_{1out} + F_{2out}$. In this case, the parallel cleaning with a

constant switching frequency between V_h and V_l can still be achieved by occasionally shutting down the CIP pump during the operation. Compared with cleaning of a single tank, the shut down frequency of the CIP pump will still be reduced by approximately half.

To validate and implement the volume based and parallel cleaning policies, a control sequence was developed for the parallel cleaning of T1 and T2 in the batch pilot plant. Figure 6.6 shows an experimental result with the parallel cleaning control sequence. The initial temperature of the detergent in the CIP tank (T7) was 66 °C and the initial concentration of detergent in T7 was already close to its setpoint as the detergent service operation was carried out in a previous experiment. The detergent clean phase started almost immediately after the pre-rinse finished, which would, otherwise, wait for at least 10 minutes under the original control sequence. The cleaning solution level is controlled roughly within the defined high and low limits (20 l and 6 l) in both vessels. As the actual flow rate of the CIP pump is much higher than the sum of outlet flow rates from T1 and T2, Eq (6.1) is not satisfied. The CIP pump was stopped occasionally.

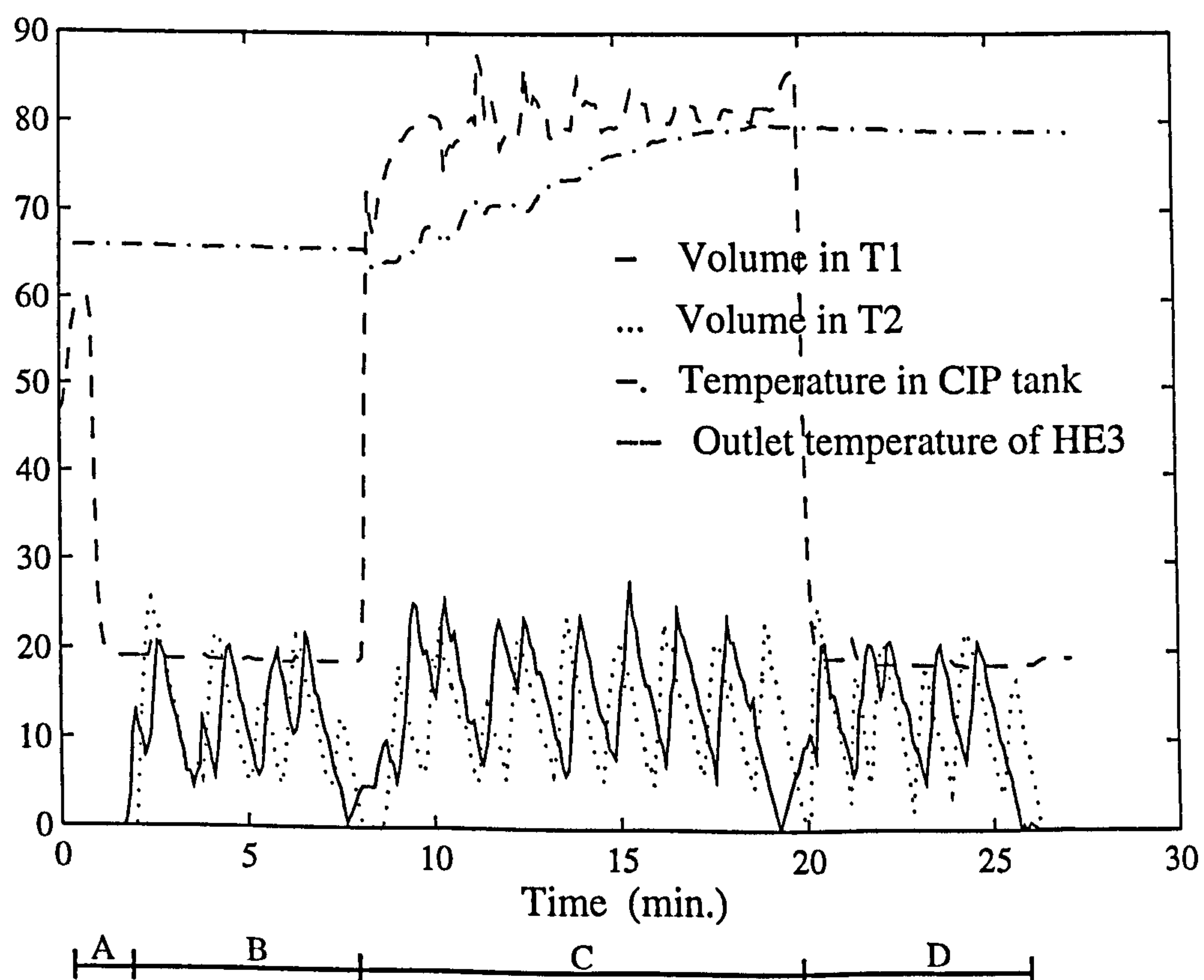


Figure 6.6 Parallel cleaning of T1 and T2 - experiment 1. A - initial CIP service; B - pre-rinse; C - detergent clean; D - final rinse.

The total time for cleaning both vessels is 28 minutes. This compares with a time for cleaning both vessels with the original policy of approximately 60 - 70 minutes, depending on the (variable) cleaning rinse time.

The simulation and experimental results show that the parallel cleaning policy has several advantages:

- Both vessels can be cleaned in the time required for cleaning a single vessel alone;
 - No shutdown or infrequent shutdown of the CIP pump is required;
 - The implementation is simple and straight forward;
 - No or less waiting time is required between the pre-rinse phase and detergent clean phase;
- These result from the improvements in the temperature control system and operating procedure.

Again, there was an excellent match between the improved operation developed on the simulation and actual results. The improved sequence worked first time in the plant, with no further tuning required.

6.2 Emergency Control and Recovery Procedure

When abnormal situations occur, such as failure of control valves, violation of operating constraints, drifting of operating parameters from the desired profiles, emergency control actions must be taken to ensure plant and personnel safety. This kind of control is usually superimposed on the analogue control loops as a safety backup. It may be triggered from software checks within a master control sequence, or triggered by control system hardware protectors. Various recovery procedures are also typically required in order to minimise the disruptions caused by the alarm and to bring the operation back to normal, if possible, rather than stop or suspend the operation completely. However, the development of detailed recovery procedures is not a simple task, and resulting schemes are often very case-dependent. The effects of emergency procedures are often difficult to assess a priori and experimental assessment may be difficult, hazardous or even impossible. These emergency procedures are therefore often defined in a very conservative way. In this section, the current emergency control procedure for the temperature control of the reactor in the batch pilot plant is evaluated. Problems of the existing emergency recovery procedure are

identified. A model based emergency recovery control scheme is then developed and tested using the simulation support environment, and finally validated experimentally.

6.2.1 Emergency Control Sequence

Under the water jacket control scheme (section 5.3), the reactor temperature control is carried out by temperature control sequence Seq 1.80 (a top sequence). This sequence calls a number of subsequences (slave sequences), that include a primary analogue control loop handling (Seq 1.103), a secondary control loop handling (Seq 1.104), a reactor high level monitoring (Seq 1.203), an analogue loop output handling (Seq 1.118), and an emergency control sequence (Seq 1.128). The tree structure of these control sequences is shown in Figure 6.7.

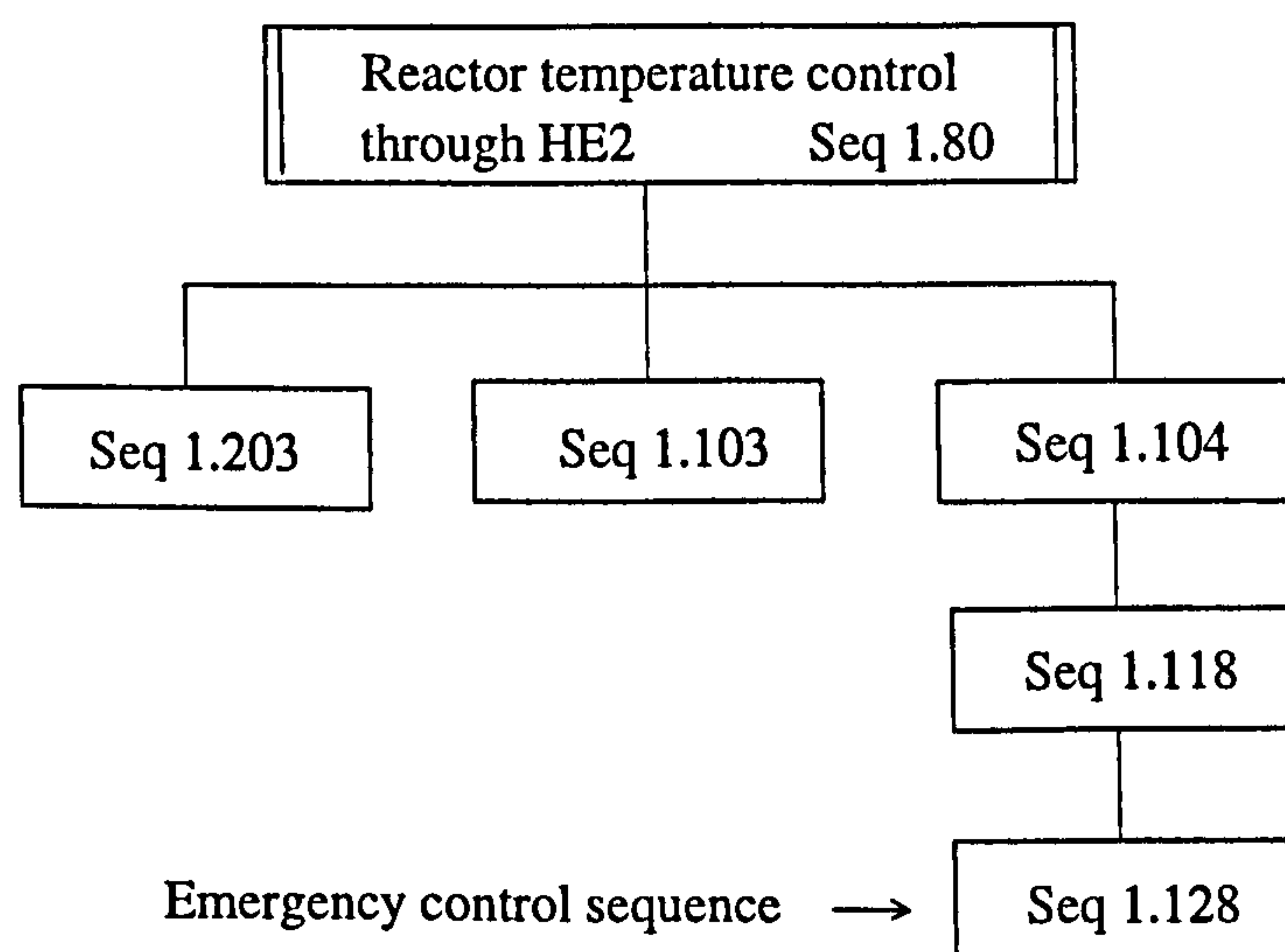


Figure 6.7 Tree structure of the control sequences in the temperature control of the reactor through heat exchanger HE2.

Due to safety consideration, the jacket inlet temperature is constrained to an upper limit (100 °C). When this limit is exceeded, the temperature control sequence alarms, and the emergency sequence is called. The current emergency control procedure disables the cascade temperature control loop, and causes crash cooling to be applied. A basic question, which was difficult to address, is whether, after this emergency procedure is activated, it is still possible to recover the batch of operation, and to determine when it is safe to resume the normal operation. Figure 6.8 shows the reactor and jacket temperature responses during one experiment using the original cascaded PID controllers. The jacket temperature safety

level is reached twice during the initial heating of the reactor, and the emergency control sequence is called. In each case, the reactor temperature control sequence was immediately restarted manually after the jacket temperature returned to a safe level (approximately 30 °C after about 2 minutes). The reactor temperature was eventually brought to the desired setpoint (85 °C) in this case. However, quick operator response is required to recover the operation. Any delay in switching back to the reactor temperature control from the current emergency control procedure will cause a longer overall reaction time, and possibly require rather more intense actions to recover the batch. The quality of the products will no doubt suffer (in our case, a retrogradation reaction may become significant) and, in the worst case, the batch will be lost. The development of a good recovery procedure is not at all simple and no recommended procedure was available in our plant/process.

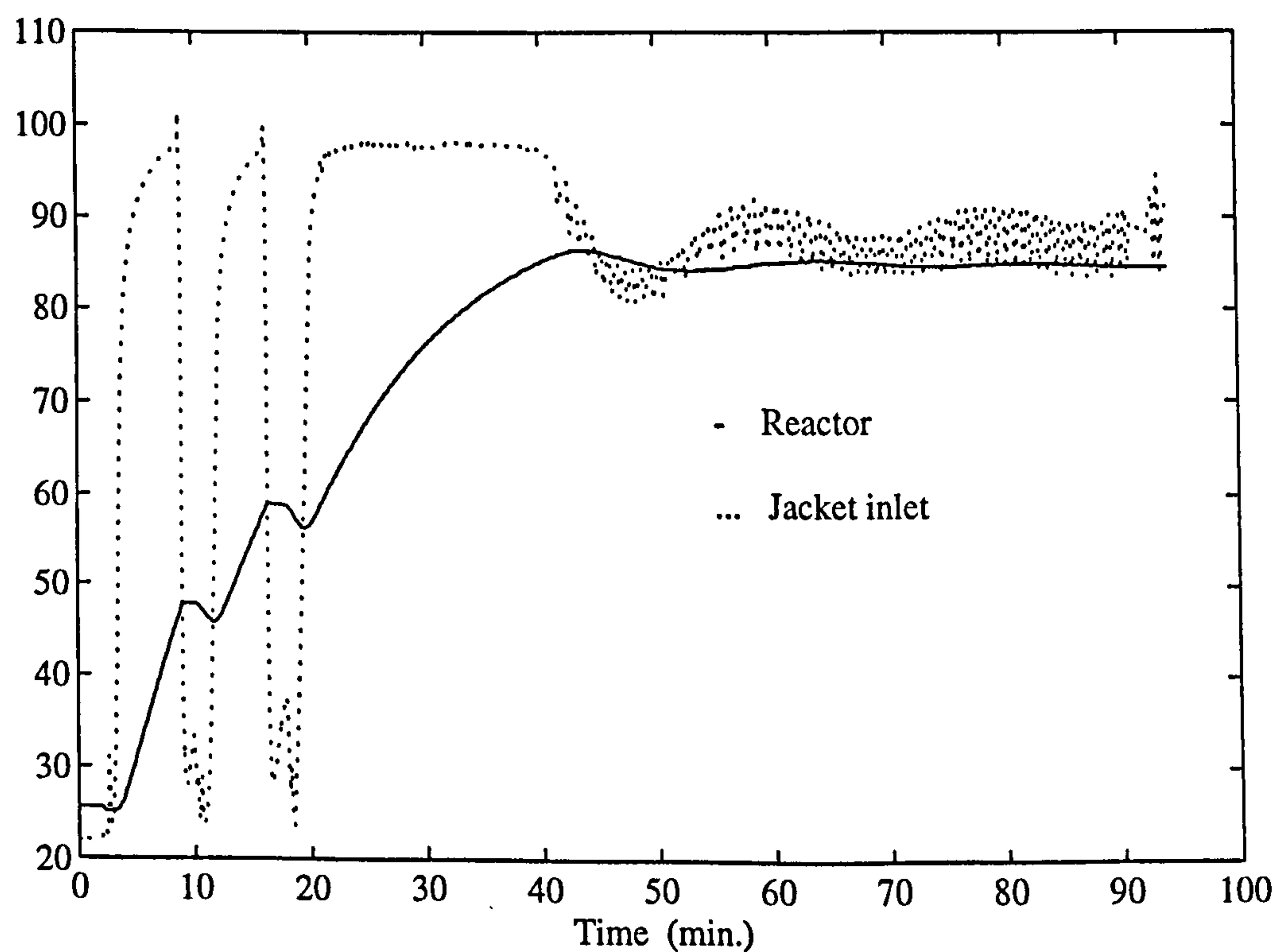


Figure 6.8 Reactor and jacket temperature responses with current emergency control procedure - experimental result.

6.2.2 Model Based Override Control

Utilising the operations support environment described in Chapter 3, a representative model based emergency recovery control procedure was developed for the temperature control of the reactor which is aimed at bringing the operation back to the normal region automatically and with minimal disruption. The reactor model was given in Chapter 4. The emergency procedure overrides the reactor temperature control loop and takes over control when a violation of the jacket inlet temperature constraint occurs. A schematic control structure is shown in Figure 6.9.

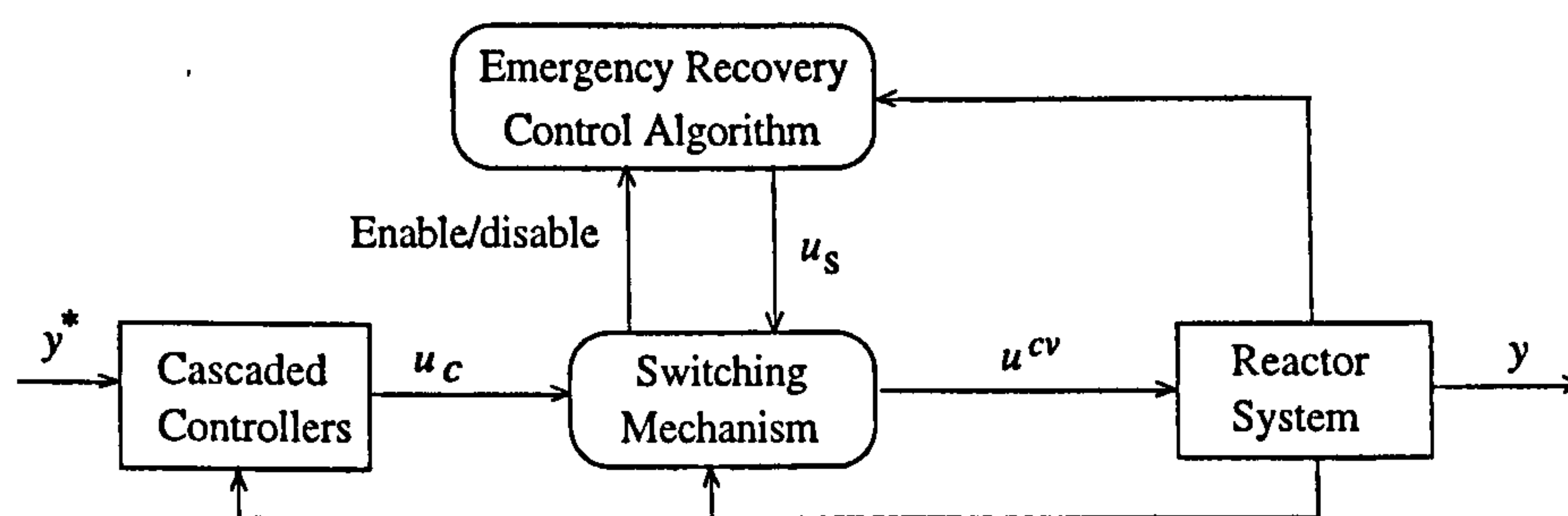


Figure 6.9 Override control scheme for emergency recovery.

The control output (u^{cv}) to the split-setting valves (CV1-4/CV1-7 in Figure 4.3.c) is selected as either the output of the cascaded control loop (u_c) or the output of the emergency recovery control algorithm (u_s) by a switching mechanism (SwM)

$$u^{cv} = \text{SwM} \{u_c, u_s\}$$

Switching Mechanism. The emergency recovery procedure is invoked by the SwM when the following logical constraint on the jacket inlet temperature becomes true:

$$T_{jin} > T_{jin}^{\max} \quad (6.3)$$

The SwM then switches the control output from the cascaded control loop to the emergency control output. The latter is determined by a model based control algorithm. The reactor is operated under the recovery algorithm until the following two conditions are both satisfied:

1. the rate of change in reactor temperature is small $\left| \frac{dT_r}{dt} \right| \leq e$ or the jacket temperature is below a lower bound $T_j < T_j^L$;
2. the jacket inlet temperature is below a specified safety margin $T_{jin} < (T_{jin}^{\max} - \Delta T_{jin}^{sm})$.

These logical conditions are represented by Eq (6.4).

$$\left(\left| \frac{dT_r}{dt} \right| \leq e \cup T_j < T_j^L \right) \cap (T_{jin} < (T_{jin}^{\max} - \Delta T_{jin}^{sm})) \quad (6.4)$$

Where e is a small positive real value, T_j^L is a lower bound on the jacket temperature ($T_j^L < T_{jin}^{\max}$), $T_{jin}^{\max}$ is the upper constraint on the jacket inlet temperature, and ΔT_{jin}^{sm} is a safety margin of the jacket inlet temperature.

When the logical condition in Eq (6.4) is again true, the emergency recovery control will be disabled and control is switched back to the cascaded control by the SwM. In other words, it is deemed that it is safe to resume the normal operation when the jacket inlet temperature has become sufficiently low and the reactor temperature has stabilised.

Emergency Control Algorithm. During the emergency control phase, the control action is determined by a control algorithm that is developed using the dynamic model of the system. The objective of the emergency control is to bring the violated variable back to the safety region automatically with minimum disruption of the normal operations. The control objective specified is to reach a safe jacket temperature value T_j^L along a reference trajectory, such that the rate of change in the jacket temperature is proportional to the distance from the safe limit. That is, the jacket temperature is assumed to move to T_j^L in a first order manner (Eq 6.5).

$$\dot{T}_j^* = K_{sj} (T_j^L - T_j) \quad (6.5)$$

The K_{sj} in Eq (6.5) is a tuning parameter that determines the speed at which the jacket temperature approaches the lower bound T_j^L . The emergency control law in terms of desired cooling water flow rate is derived by combining Eq (6.5) and the reactor model described in section 4.5 to give:

$$F_{cw} = \frac{(V_j K_{sj} (T_j^L - T_j) - \beta_1 (T_r - T_j))}{(T_{cw} - T_j)} \quad 0 \leq F_{cw} \leq F_{cw}^{max} \quad (6.6)$$

where F_{cw}^{max} is the maximum cooling water flow rate. From Eq (6.6), the desired emergency control output u_s (0-100%) can then be calculated from the model of the cooling water valve (CV1-7) described in section 4.5. This approach has been successfully used by Cott *et al.* (1989) and Brown *et al.* (1990).

A recovery procedure using this controller was tested on the simulator and then implemented in the pilot plant. An experimental result with the automatic emergency control recovery sequence is shown in Figure 6.10. As in the previous experiment (original emergency control and manual recovery in Figure 6.9), the normal operation objective is to bring the reactor temperature to a setpoint of 85 °C from initial room temperature (24 °C). During the reactor heat-up, the jacket inlet temperature exceeds its high limit under the cascaded PID control and the new emergency procedure is called that overrides the cascaded control loop. The jacket inlet temperature is brought back to a safe operating region by the emergency control algorithm (Eq 6.6) and then control is switched back to the cascaded control. Tuning parameters used in the automatic emergency control procedure are also given in Figure 6.10. Compared to the previous case in Figure 6.9, manual intervention is not required to restart the normal control sequence. The reactor temperature during the heating-up is much less disrupted than the previous case, thus reducing the reactor heating-up time, energy consumption, and the risk of quality problems.

Of course, a better control alternative would be to use a model based controller for the normal operation which can handle hard constraints. Even so, back up emergency and recovery would need to be implemented anyway for safety reasons.

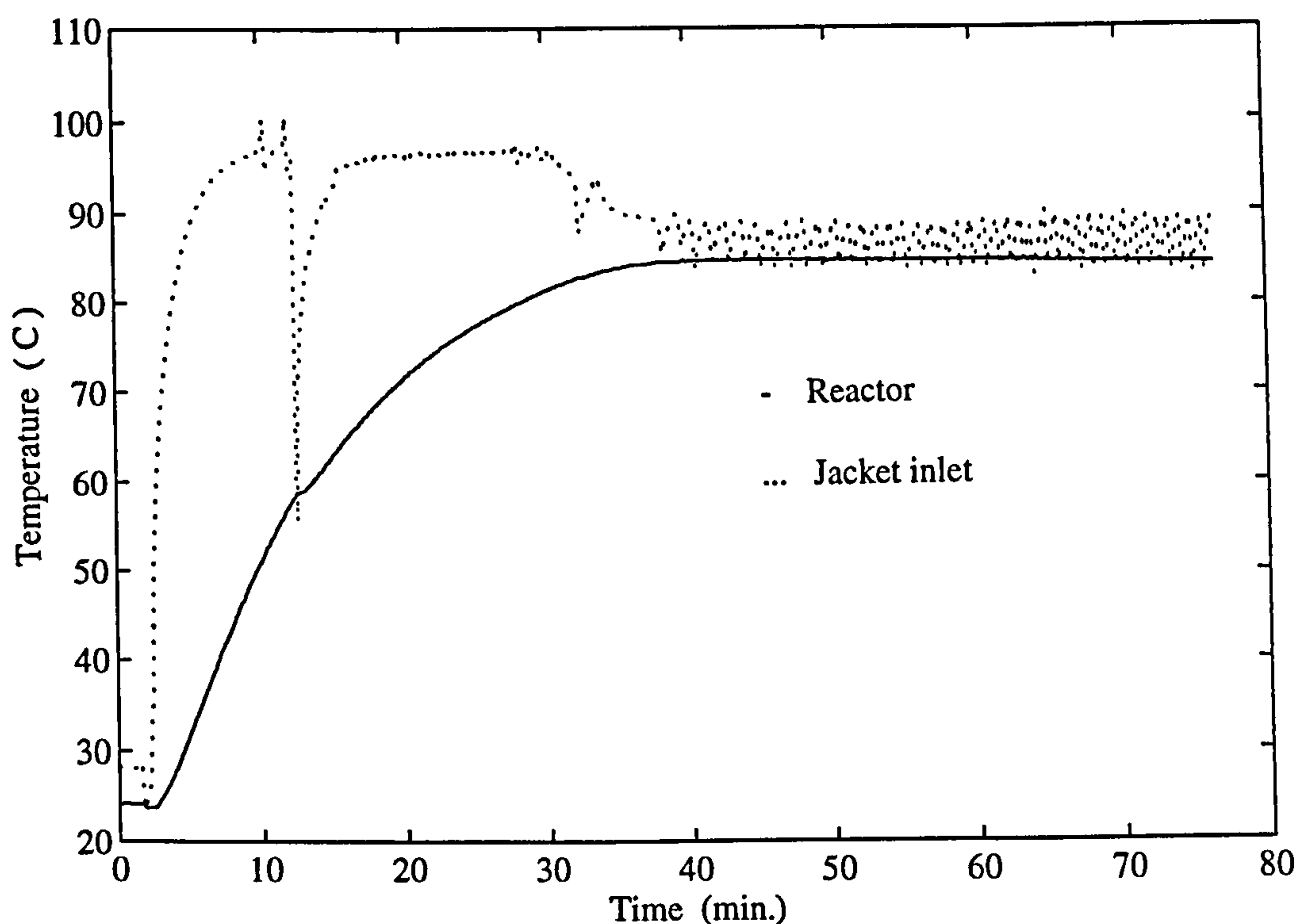


Figure 6.10 Reactor temperature control with model based emergency recovery procedure - experimental result (e : 0.075 °C/s, T_j^L : 65 °C, ΔT_{jin}^{sm} : 12 °C, K_{sj} : 10.0)

6.3 Conclusions

In this chapter, a dynamic model and a simulation based approach were used within the operations support environment described in Chapter 3 to assist in the development, testing and commissioning of several typical control sequences. We showed that by modelling the system and the detailed control actions imposed on the dynamic model, the effect of the operating procedures on the dynamic behaviour of the system can be evaluated quantitatively. The simulation techniques was applied to quickly identify improvements which could otherwise be difficult to assess and/or test experimentally. The suggested improvements in control procedures were easily tested on the simulator before implementation in the plant. Actual implementation could then be done very quickly and reliably. This was demonstrated in the CIP operation and emergency recovery control examples. In all cases, the control sequences thus developed worked in the plant first time and without need for further tuning.

The simulations and experimental results for the CIP operation show that the detailed cleaning procedures have a substantial effect on plant cleaning times and operability. With respect to the original procedures, up to 40% of total time can be saved for cleaning of the batch pilot plant by implementing the improved control sequences and parallel cleaning policy, with less energy consumption. The volume-based cleaning policy can be easily implemented in industrial plants and the policy for simultaneous cleaning of two vessels can be easily extended to cleaning N parallel vessels, at zero capital cost investment.

On the second example, the model based emergency recovery control algorithm provides a better control for our batch reactor system. Emergency recovery control is performed in a deterministic manner. The operation does not need to be stopped or held, therefore, the disruption to the higher level supervisory control is reduced. No manual intervention is required thus reducing the load on the operators.

It should be observed that specifying and modelling control sequences is an engineering intensive task. It would be highly desirable to be able to develop detailed control sequences automatically, starting from the plant and specification models, into the control language primitives required to drive both the simulation and the plant control system. This is beyond the scope of this study, but useful progress in this direction has been reported recently (Sanchez, 1994).

Chapter 7

Production Scheduling and Supervisory Control

Batch production requires executions of a number of individual control sequences in a defined order. Traditionally, this has typically been carried out by operators who are responsible for initiating proper sequence control commands and for taking proper actions in response to changes in process operation - *operator controlled operation*. Problems related to operation in this mode are discussed section 7.1. Section 7.2 addresses the implementation of supervisory control for the operation of the batch pilot plant. Optimal off-line production schedules for the starch based processes with CIP are first generated using MILP technique. The optimal off-line schedules are then used as inputs (representing production requirement) to the supervisory control system for on-line execution of the production schedules. The development of an alternative operating procedure for the starch degradation process is discussed in section 7.3. Conclusions are given in section 7.4.

7.1 Operator Controlled Operation

In order to determine the relative benefits of using supervisory control for scheduling and monitoring the pilot plant operations on-line, a production plant involving the starch liquefaction process by α -amylase to produce dextrins and CIP operation were first carried out under an operator (the author) control through an operator terminal. The operator is merely responsible for initiating top control sequences. This is done in practice by sending simple commands from the operator console, *i.e.* by setting control sequence flags in the simulator, or by entering Operator Entry Codes in the ACCOS control system corresponding to major operations, according to given schedule and recipe procedures.

Figure 7.1a and 7.1b show traces (in simulation) of some important variables for an operation consisting of the production of a single batch of dextrans, followed by cleaning in place at the end of production. The total time for this operation is 216 minutes in this example. The total time could vary, depending on the skill of the operators.

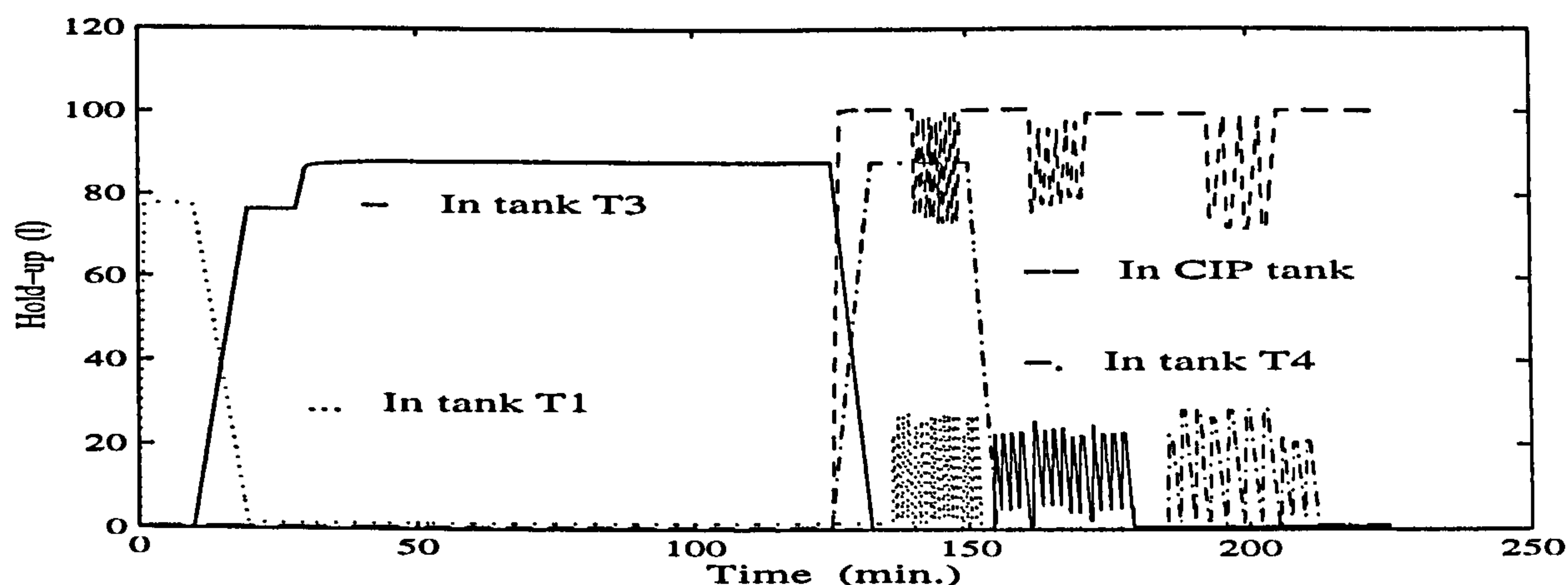


Figure 7.1.a Continuous hold-up measurements over the entire production.

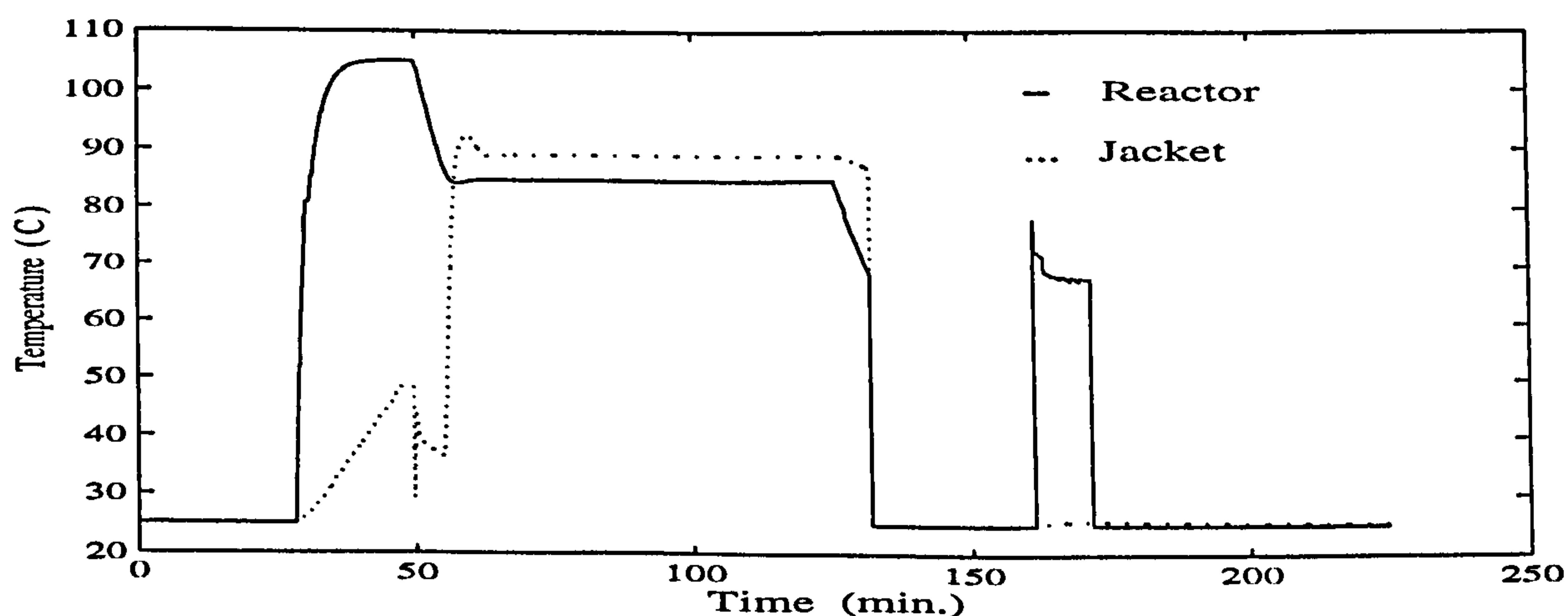


Figure 7.1.b Temperature responses of the reactor and jacket over the entire production

Table 7.1 summarises the control sequences to be used/initiated and control parameters to be specified by the operator for the above operation. Each individual operation (*e.g.* transfer of material from one vessel to another) is carried out automatically by the control sequence and

most of the control sequences are terminated automatically according to pre-defined termination conditions specified in the sequences (with only exception of the manual mixing operation in T1). Nevertheless, the operator is still very heavily loaded. The operator is not only responsible for monitoring the current status of the plant and of the active control sequences, but also for initiating proper control commands (*e.g.* start sequences) and for specifying desired control parameters (*e.g.* batch sizes, flow rates, and setpoints). Furthermore, the operator has to monitor all current and potential resource contentions when planning to start any operations.

Sequence descriptions	Sequence start/stop flag	Parameters	Type of operation
fill T1 with water	T1_W	weight of starch & water	auto
mixing starch slurry	T1_Mix		manual
transfer T1 to T3	T1_To_T3	batch size, flow rate	auto
pH control	T3_pH	pH setpoint	auto
sparge heating control	T3_Sparge	temp. setpoint	auto
water jacket control	T3_WJ	temp. setpoint	auto
transfer T3 to T4	Tr_T3_T4	temp. setpoint	auto
discharge T4	T4_P		auto
CIP service	Ser_CIP	temp. setpoint, detergent con.	auto
cleaning T1	CL_T1	-	auto
cleaning T3	CL_T3	-	auto
cleaning T4	CL_T4	-	auto

Table 7.1 Summary of operations for starch degradation process with CIP - production of one batch under the operator control

Operation under operator control means that the decision making is carried out essentially in an open-loop. This has a number of disadvantages:

- total operation time is often longer than that under the closed-loop control (see next section). This depends on the operator involved and the complexity of the operations;
- there is considerable room for mistakes to be made. The number of actions that operators must take is often large, and these actions may need to be taken in a certain order;
- the safety of the operation also relies on the skill and responsibility of individual operator;
- repeatability of the operations is low as it depends on the individual operator.

Furthermore, it has been found particular difficult in the operator-controlled operations to deal with processes which involve unstable materials (*e.g.* the starch slurry has to be transferred/processed immediately), or which materials are desired to be processed with zero wait. In such cases, the operator has to monitor and predict the availability of processing units which may be potentially used to process the unstable material. This became extremely difficult, even impossible, when complex operations are involved, or a large number of batches are to be processed.

7.2 Production Planning and On-line Supervisory Control

The example in the previous section has demonstrated that even for a simple process with a small number of processing tasks, operation of batch plants could be very complicated. There is a clear need for closing the production control loop at a supervisory control level. As mentioned in Chapter 2, the STN representation and a MILP model provide a rather general systematic framework for formulating production planning and scheduling problems arising in batch processing system. However, the optimal scheduling algorithms often require a significant amount of computation time and make serious modelling assumptions (*e.g.* time discretization, instantaneous transfers). It was found that the optimisation times, even for the relatively small examples involved in our pilot plant, can be very high. On the other hand, fast heuristic algorithms (*i.e.* the minimum makespan algorithm used in SUPERBATCH) can be used on-line for supervisory control, but are unable to deal with all scheduling decisions, complex interactions and optimality. Therefore, two levels of decision making are considered here:

- off-line optimal production planning and scheduling. A more rigorous planning and scheduling system (*e.g.* gBSS) based on mathematical programming techniques is considered to generate initial optimal production plan and schedule;
- on-line supervisory control. The SUPERBATCH system based on a fast heuristic search method is used for on-line scheduling and production control.

The off-line optimal production schedule is transformed into a production plan required by the supervisory control system which supervises, monitors and updates the execution of the production plan on a minute-by-minute base in a closed loop manner.

7.2.1 Off-line Optimal Production Planning and Scheduling

The objective of this section is to generate off-line optimal production schedules for the starch based processes and CIP operation based on MILP techniques and the aggregated plant and operation models discussed in Chapter 4. The off-line production scheduling is performed using the gBSS software (Shah, 1992). The following cases are considered:

- **Case I** produce a single product with production goals of:
 - a delivering 170 Kg of dextrins between time 0 and 285 minutes with a time horizon of 285 minutes. A time discretization of 5 minutes is used in the MILP model;
 - b delivering 950 Kg of dextrins between time 0 and 1440 minutes (24 hours production) with a time horizon of 1440 minutes. Again, a time discretization of 5 minutes is used in the MILP model.
- **Case II** produce two products (dextrins and cyclodextrins) with a production goal of:

delivering 170 Kg of dextrins and 150 Kg of cyclodextrins between time 0 and 295 minutes. A time discretization of 5 minutes is used in the MILP model.

In all cases considered, the entire plant is assumed to be initially clean and idle, and it is required that all equipment units and related transfer routes used in processing must be cleaned by the end of production. No intermediate cleaning between batches is required if a unit is used to carry out two consecutive instances of the same processing task. However, intermediate cleaning is needed if a unit is used to perform consecutively different tasks. In all cases, independent cleaning is used for each vessel (no parallel cleaning). The STN models of the processes were given in Chapter 4.

In all cases considered, the scheduling objective is to deliver the required quantity of product(s) as soon as possible. A linear penalty term for every time interval beyond a minimum time (here, 0) is used, with a penalty weight of two for all cases. Solutions are optimised within a margin of 10% for all cases.

The optimal MILP schedules obtained for the above cases are shown in Figure 7.2, 7.3 and 7.4, respectively. In the Gantt chart schedules, each unit (on the left hand side of the chart) is

represented along the Y-axis with the allocation of the unit to processing or storage tasks over time (in minutes). A unit allocation is indicated by a horizontal bar bounded by short vertical bars for each task. The tasks number (according to the legend at the right hand side of the chart) is indicated above the allocation bar and the task size is shown below the bars. In the *Case Ia* (Figure 7.2) and *Ib* (Figure 7.3), all of the "Make_Sl_A" tasks are allocated to the unit T1 only, although unit T2 is also suitable for performing the task. Similarly, the "Discharge_A" tasks are all allocated to the unit T4 only, even though T5 can also be used for carrying out the task. This is due to the inclusion of cleaning constraints which require each units used by the production must be cleaned at the end. The "Cleaning" task is therefore only imposed upon the units T1, T3 and T4 at the end of production. Production is completed in 280 minutes in Case Ia (2 batches of product and cleaning) and in 1420 minutes in Case Ib (12 batches with cleaning).

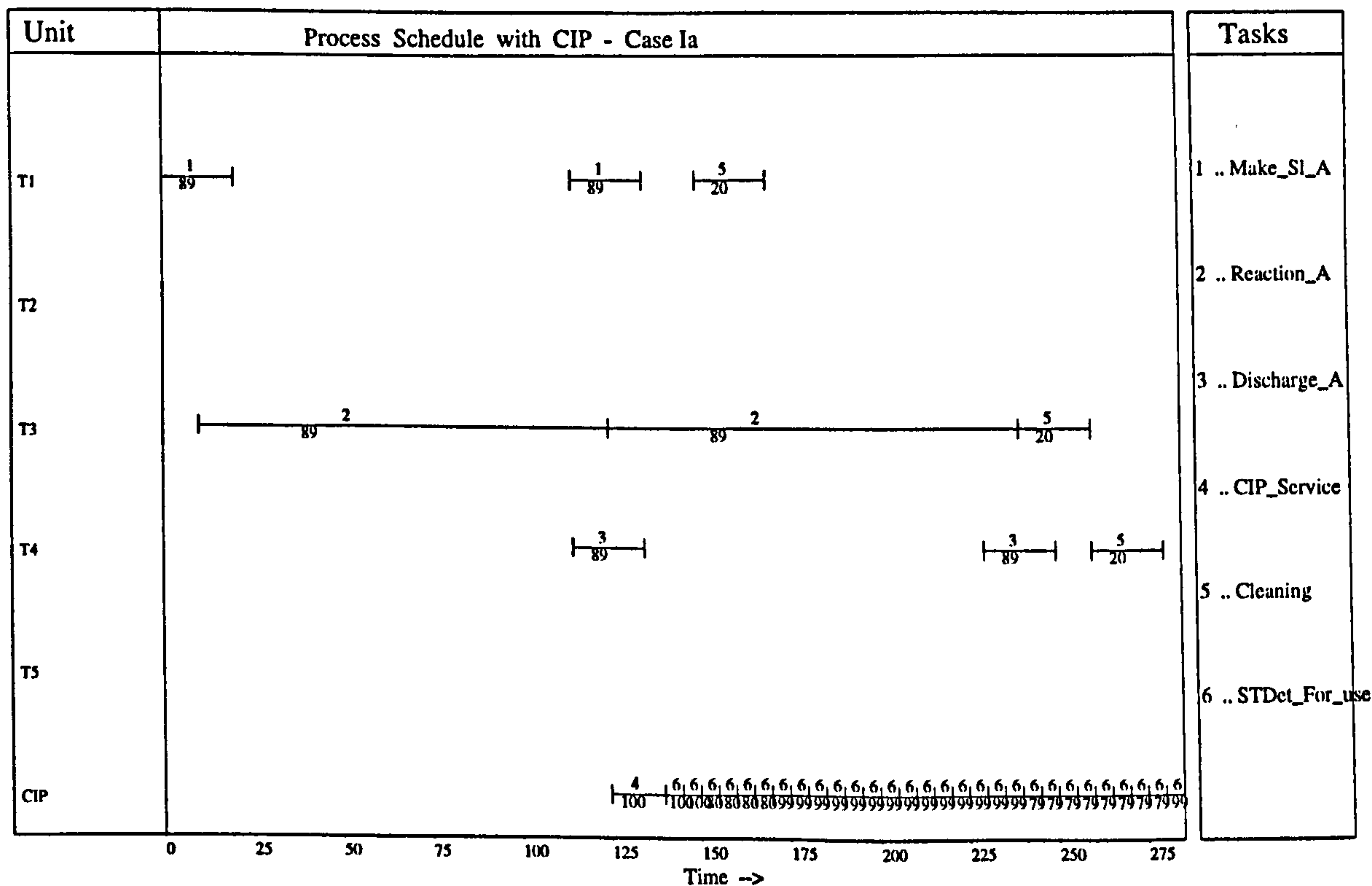


Figure 7.2 *Case Ia* - optimal production schedule for producing a single product with final CIP

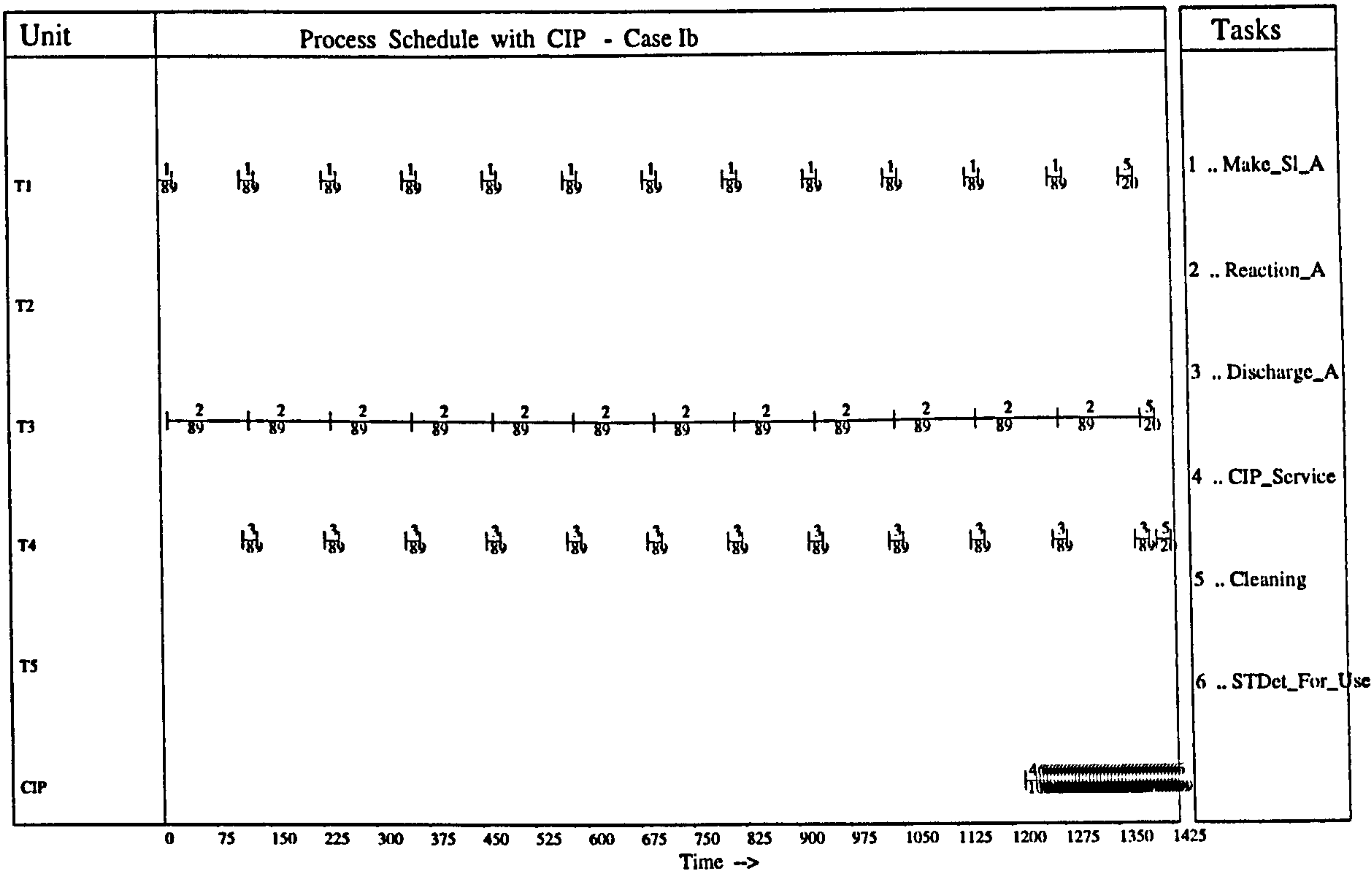


Figure 7.3 Case Ib - optimal production schedule for producing a single product with final CIP

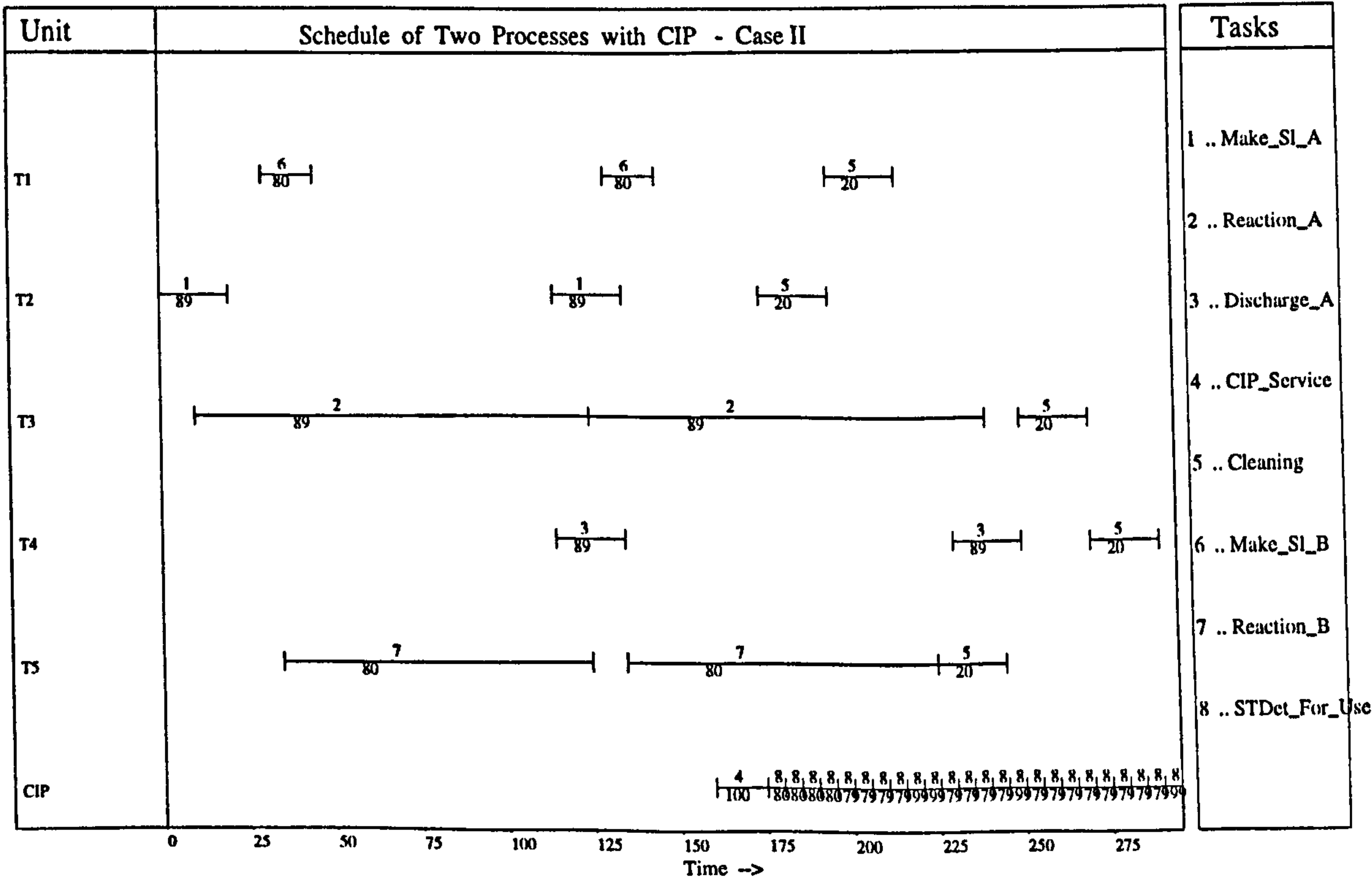
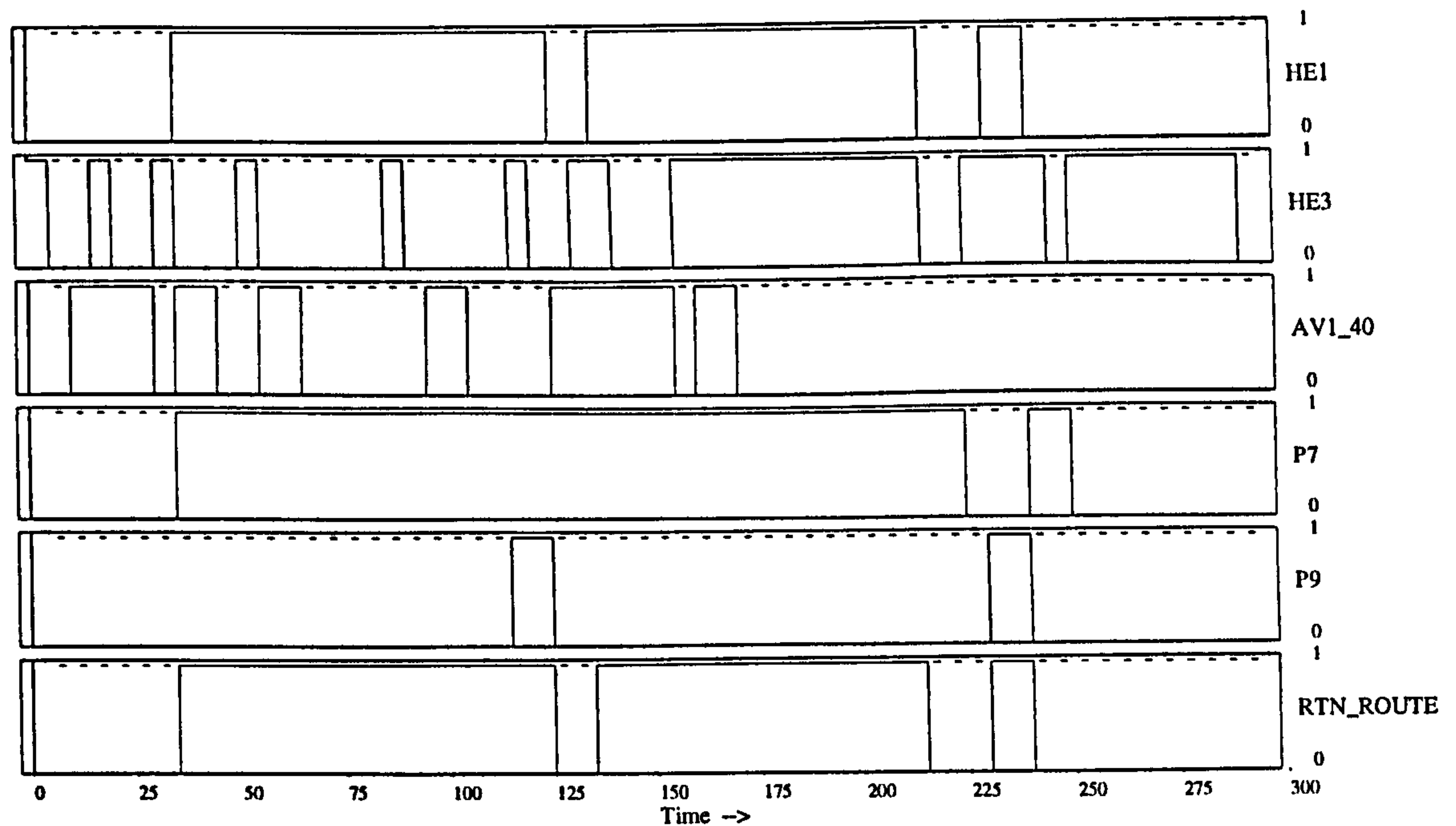


Figure 7.4 Case II - Optimal production schedule for producing two products with final CIP

Figure 7.5 Utility profile for the *Case II*

For the single product case, the optimal schedules in Figure 7.2 and 7.3 show clearly that the "Reaction_A" task, which can only be performed in the reactor T3, is the bottleneck in the production schedule. While the reactor T3 is busy almost all the time throughout the production horizons, other units are idle most of the time.

Plant resources, on the other hand, are much better utilised in the two products case (Figure 7.4). Two batches of each product are produced, in a total time of 290 minutes. All the "Reaction_A" tasks are allocated to the unit T3 and the "Reaction_B" tasks are all allocated to the unit T5. No intermediate cleaning is scheduled on any of the five units. The schedule for this case is rather complicated due to the "utility" constraints (common transfer routes, equipment, etc.). Although the unit T4 can also be used to perform the "Reaction_B" task, it is not scheduled as "Reaction_B" in T4 would require the same heat exchanger (HE1), recirculation route (Rtn_Route) and pump (P7). The plant connectivity constraints play a key bottleneck in the case. All of the utilities are used most of the time in the production. The utility profile for this case is shown in Figure 7.5.

For *Case Ia* with a small product demand and a short time horizon, an optimal solution is obtained within 240 seconds CPU time on a Sun Sparc 10. For other two cases, the computation times are prohibitively high. The computation statistics for the MILP problems are summarised in Table 7.2.

Computational Details	<i>Case Ia</i>	<i>Case Ib</i>	<i>Case II</i>
No. of Integer Variables	1135	6195	2221
Total No. of Variables	2925	15433	4931
No. of Constraints	3134	16245	4500
No. of Non-zero Elements	13405	73009	24984
No. of Simplex Iterations	2244	33304	94513
CPU Time (s) Sun Sparc 10	240	19965	40504

Table 7.2 MILP computation statistics

7.2.2 SUPERBATCH Production Input

For on-line production control, the MILP optimal schedules generated by gBSS need to be translated into production schedules (in terms of sequence of batches, task allocations and batch sizes) as required by the SUPERBATCH software. As the mapping between the tasks in the MILP model and the control phases used in the SUPERBATCH has already been established (as discussed in Chapter 3 and Chapter 4) and material transfers were also implicitly considered in the MILP model to ensure accurate production schedules, the transformation of MILP schedule into the SUPERBATCH input (production requirements) becomes relatively straightforward. Related tasks are grouped together into well defined master procedures as given in Chapter 4. Each master procedure corresponds to a batch and is given a unique batch number by SUPERBATCH. The sequence of batches is determined by the time precedence of the first task of the master procedures in the optimal production schedule. The production plans transformed manually from the optimal MILP schedule for *Case Ia* and *Case II* are given in Table 7.3 and 7.4, respectively. The production plan for the *Case Ib* is similar to the one for *Case Ia* (only larger) and is omitted here.

No.	Description	Batches ID & Size	No.	Description	Batches ID & Size
1	Produce maltodextrins	BNo1_T3 89.0	5	Cleaning T1	CL_T1 20.0
2	Discharge product	END_BNo1 89.0	6	Discharge product	END_BNo2 89.0
3	Produce maltodextrins	BNo2_T3 89.0	7	Cleaning T3	CL_T3 20.0
4	CIP service	CIP_Service 100.0	8	Cleaning T4	CL_T4 20.0

Table 7.3 SUPERBATCH production plan transformed from the MILP schedule from Case Ia

No.	Description	Batches ID & Size	No.	Description	Batches ID & Size
1	Produce maltodextrins	BNo1_T3 89.0	9	CIP service	CIP_Service 100.0
2	Make slurry_B	BNo2_A 80.0	10	Cleaning T2	CL_T2 20.0
3	Produce cyclodextrins	BNo2_B 80.0	11	Cleaning T1	CL_T1 20.0
4	Discharge maltodextrins	END_BNo1 89.0	12	Discharge cyclodextrins	END_BNo4 80.0
5	Discharge cyclodextrins	END_BNo2 80.0	13	Discharge maltodextrins	END_BNo3 89.0
6	Produce maltodextrins	BNo3_T3 89.0	14	Cleaning T5	CL_T5 20.0
7	Make slurry_B	BNo4_A 80.0	15	Cleaning T3	CL_T3 20.0
8	Produce cyclodextrins	BNo4_B 80.0	16	Cleaning T4	CL_T4 20.0

Table 7.4 SUPERBATCH production plan transformed from the MILP schedule in Case II α

For more complicated cases, it is obviously desirable to generate supervisory procedures automatically from a MILP schedule. An automatic subgoalng method was developed by Crooks (1992) which utilises a similar MILP formulation, with additional detailed description of the process, equipment connectivity, and operating constraints, to generate an optimal task schedule required for achieving an operating goal. The optimal task schedules produced by the subgoalng program are then further analyzed to generate supervisory procedures which are in a format suitable for input to the SUPERBATCH system. However, the STN model must be in sufficiently detail, e.g. each control phase must be modelled as a task in the STN model. The

size of the mixed integer mathematical programming problems from this approach is often too large to be tractable. The optimal production schedules in this study are transformed into the SUPERBATCH production input manually since the production schedules for the processes in this application are relatively simple and the manual generation of the production schedule did not present as a major difficulty.

7.2.3 Supervisory Control of Batch Plant

Given the production requirement obtained in the previous section, the objective here is to show how the production plans can be executed on-line by the supervisory control system which is linked to the low level controllers. At the supervisory control level, operational feasibility and computational efficiency become very important since this is an on-line activity. The SUPERBATCH system (Version 1.06, 1994) is used to perform this control function. The major economic decisions obtained from the external higher level planner/scheduler are imported to the system and verified for feasibility. A more detailed production schedule is generated first in an off-line version of the software. Upon satisfactory confirmation that the more detailed schedule is acceptable, it is then passed on to an on-line portion of the system, an on-line monitor, which runs continuously. New requirements are processed by the monitor and executed through the control systems. The monitor can be connected to either the dynamic simulator or the actual plant through Control Link. The main functions of the SUPERBATCH system are:

- to coordinate off-line planning and scheduling activities with those of the production control system;
- to initiate, supervise and coordinate the execution of the scheduled activities carried out by lower level controllers (*e.g.* sequence control);
- to accommodate any variation at either higher or lower levels while enforcing feasible and safe operation. This is done by on-line rescheduling the production plan.

In this study, the supervisory control is linked to the gPROMS dynamic simulation of the plant described in Chapter 4. As the simulator is essentially a "duplication" of the real control system and the plant, the link with the actual plant is similar. The execution of the production schedule is initiated and supervised by the on-line monitor in real-time, and control commands (*e.g.* start a sequence) and control parameters (*e.g.* batch size, setpoints, *etc.*) are passed to the simulator. The status of the control sequences (*e.g.* started, completed, alarmed, on hold, *etc.*) and of the plant (inventory levels, final batch sizes achieved, *etc.*) are fed back to the supervisory control so that updating and rescheduling can then be performed. Detailed

information related to the current production schedule and the status for the execution can also be obtained through an operator terminal. Authorised operators are also allowed to intervene on the production during execution if necessary through the operator terminal. This also permits the execution of manual activities and the definition of major process disturbances, such as equipment failures, and changes of production demand and schedule (e.g. to add new orders, delete existing batches, etc.). A more detailed system configuration when used to drive a simulator is schematically shown in Figure 7.6.

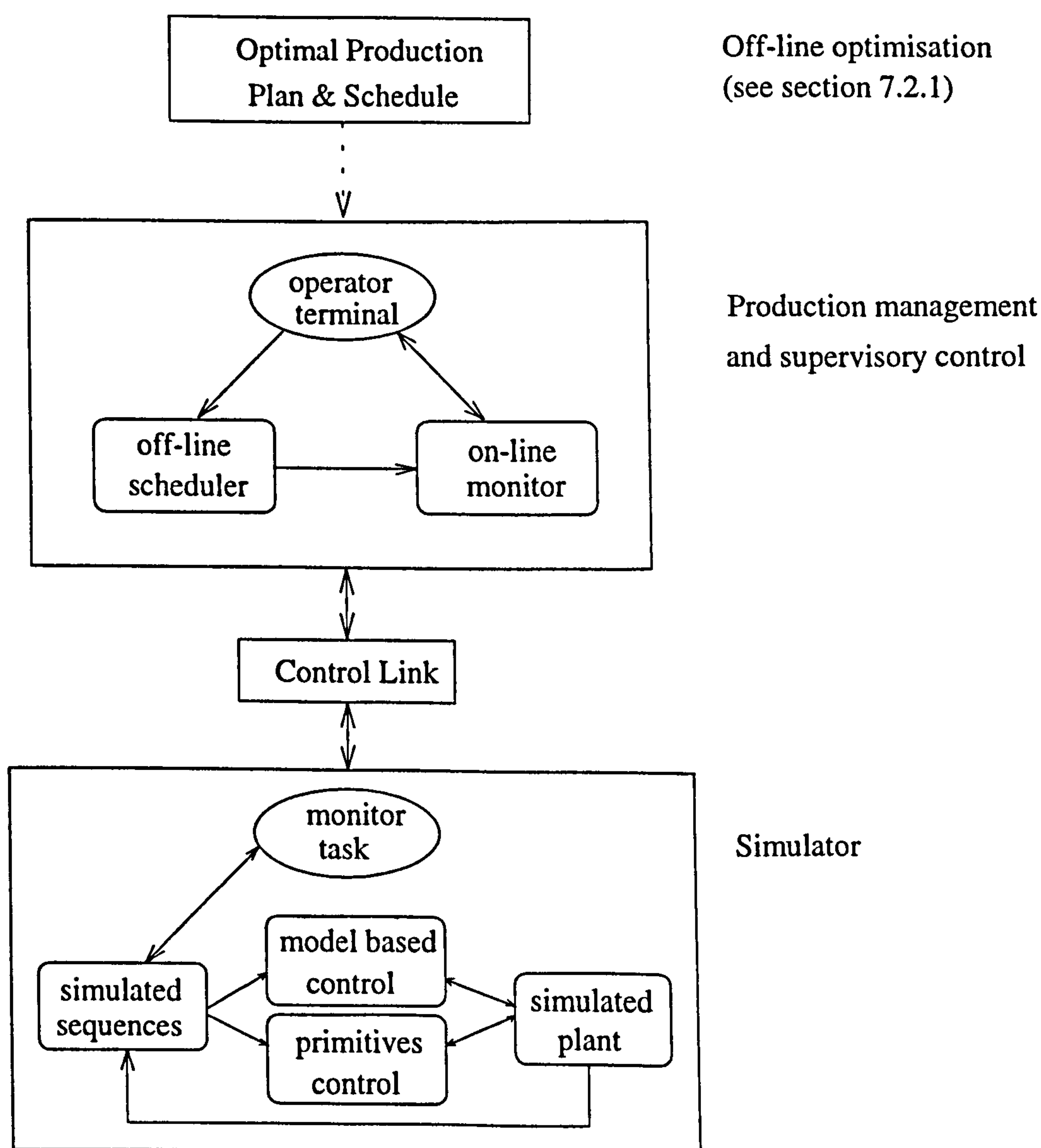


Figure 7.6 Detailed supervisory control system configuration linked with simulator

To demonstrate the capability of the supervisory control system of handling process and production variations, the following types of "disturbances" are introduced:

- "disturbances" from the planning level (from upper level)
 - deletion of existing batches and addition of new batches to the existing schedule;
 - taking equipment out of services.
- "disturbances" from the process and its operation (from lower level).
 - variation of processing times in the simulator;
 - delay of operator responses during manual operations.

Given the production plans obtained in the previous section and based on the detailed model of the plant and the control system discussed in Chapter 4, on-line execution of the production schedules is then performed using SUPERBATCH (Version 1.06, 1994). The initial start times are set to be 12:00, 1 January, for all the cases. The production schedules are updated at a frequency of one minute.

Case Ia. Figure 7.7 shows a detailed schedule for *Case Ia* at time 12:00 produced by SUPERBATCH off-line schedule for the production plan given in Table 7.3. The expected finishing time is 16:41 (shown in the bottom right of the figure). The Gantt chart representation used here is slightly different from that of gBSS. All the processing, storage, and utility resources are shown on the same chart (left hand). Storage resources are displayed as the predicted inventory level profiles (0-100% scale) and final status (final material types stored and inventory levels). The allocation bars for batch resources refer to phases with the number above each bar refers to the batch number, and the number below the bar refers to the phase step number within the corresponding master procedure. The current time is shown at the left of the horizontal scale. Once the off-line schedule is accepted (by the plant engineer, operator, *etc.*), it is passed to the on-line monitor for execution. This is checked again against the state of the plant at this time, and if ok, execution begins. In this example, at the time 12:01, the batch BNo1_T3 is being executed and its status of execution is OK (EXECOK). Seven control phases, including START and STOP, are required to process the batch. The control phase STEP_1 for this batch (transferring process water into tank T1) was initiated at 12:01, and expected to be completed at 12:06. A summary of production as scheduled at this time is given in Table 7.5.

Figure 7.8 shows the new production schedule at time 12:27. The expected finishing time is now changed to 16:38. The phase STEP_1 was actually completed at 12:02 rather than at 12:06 as estimated in the initial schedule (based on nominal times for scheduling). A batch summary at this time is given in Table 7.6. By the time 12:27, four control phases have finished successfully (FINIOK) and STEP_5 (corresponding to the gelatinization phase in T3 under sparge control) is being executed.

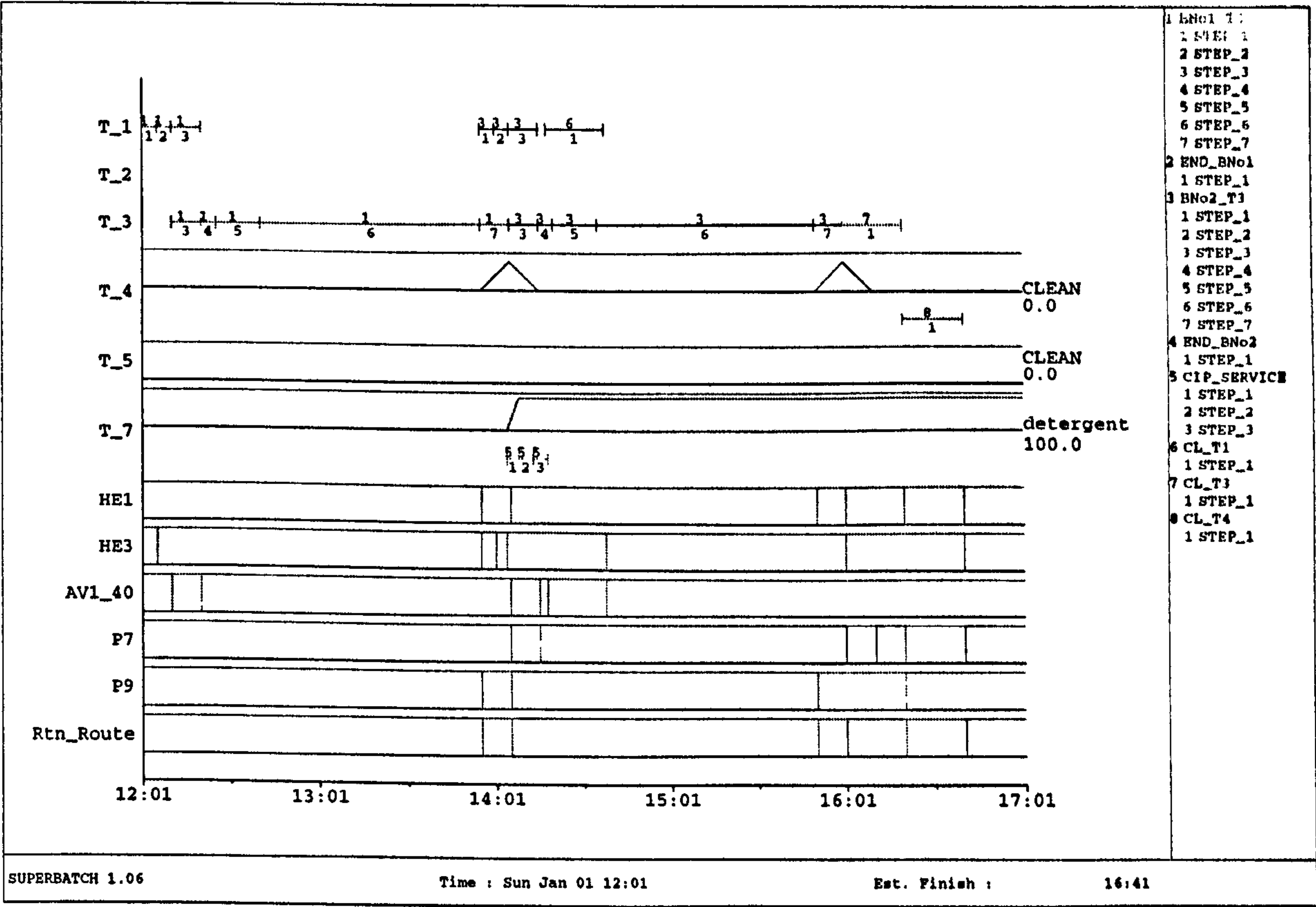


Figure 7.7 Case Ia - process schedule at time 12:00

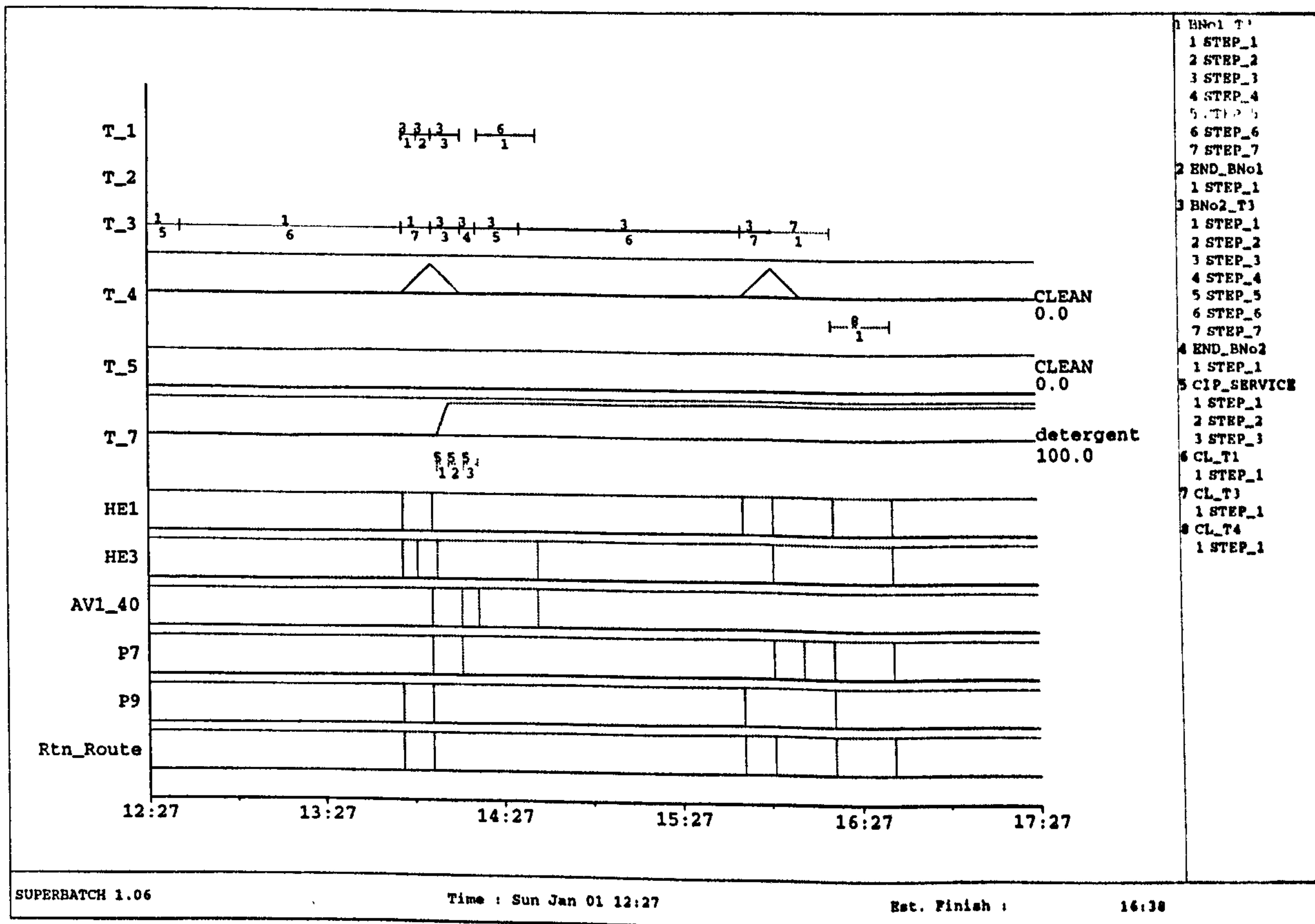


Figure 7.8 Case Ia - process schedule at 12:27

The entire production schedule was eventually successfully completed in 298 minutes at 16:58 for this case. This is 17 minutes later than the original planned completion time (see Figure 7.7). The variations were due to difference in the nominal data used in the original plan and actual times and rates which occurred in the simulation, *i.e.* flow rates for material transfers, the amount of time it took for operator/manual actions, and event-dependent operations. These factors also occur in the other cases.

A trace of some continuous holdup measurements over the entire operation is given in Figure 7.9, while Figure 7.10 shows the temperatures of the reactor and jacket under the cascade GMC control.

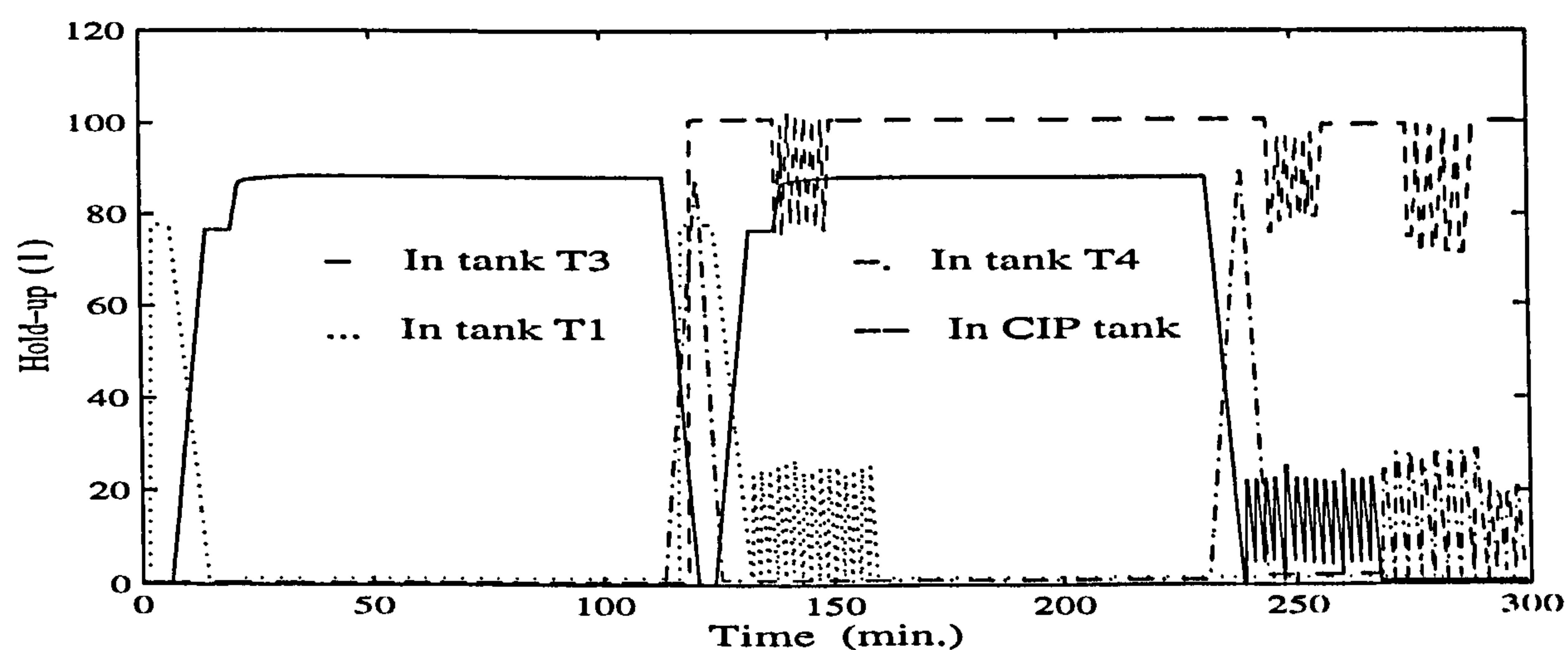


Figure 7.9 Case Ia - continuous holdup measurements over the entire operation.

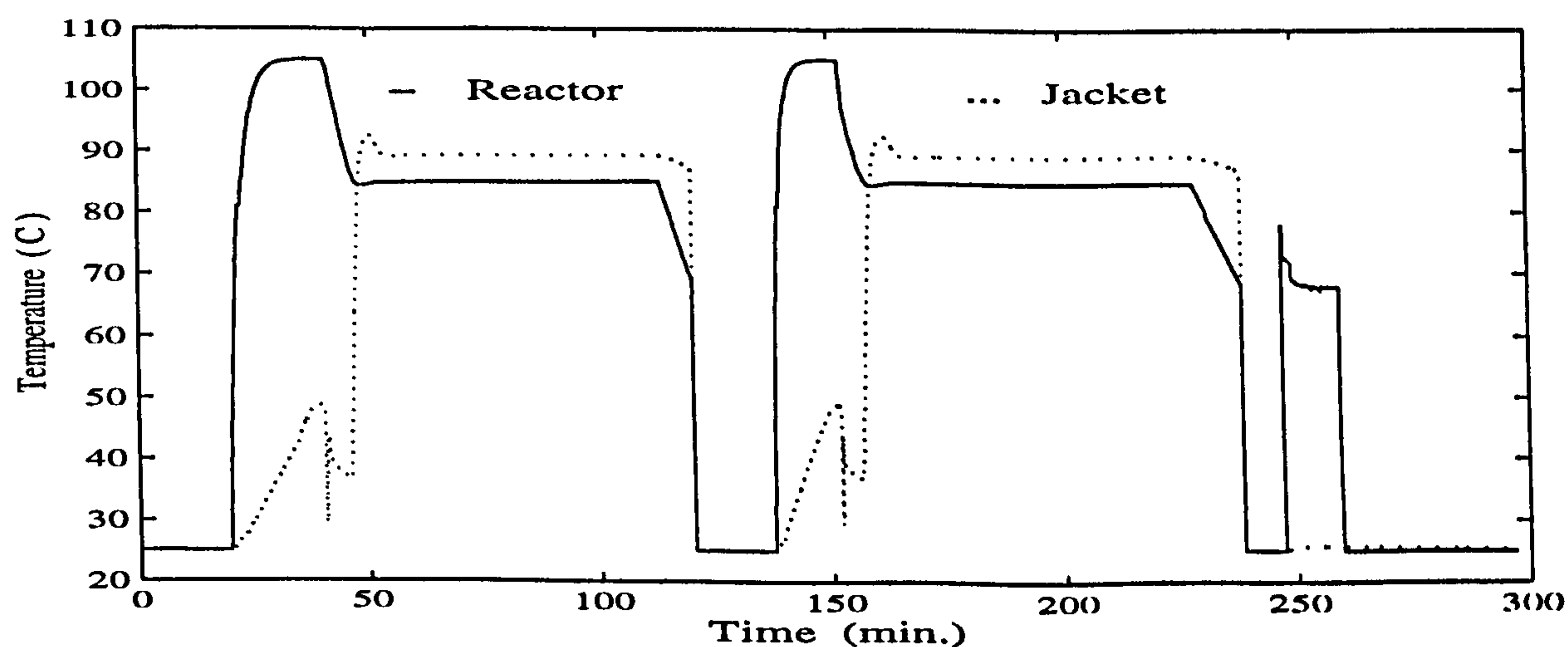


Figure 7.10 Case Ia - temperatures of the reactor and jacket over the entire operation.

SB001: BATCH SUMMARY			12:01 Sun 01 Jan 95		
Batch	Type	Size	Status	Start	End
1 BNo1_T3	PA_T1T3T4	89.00	EXECOK	001 12:01	001 14:08
2 END_BNo1	EMPTY_T4	89.00	WAITOK	001 14:08	001 14:20
3 BNo2_T3	PA_T1T3T4	89.00	WAITOK	001 13:54	001 16:01
4 END_BNo2	EMPTY_T4	89.00	WAITOK	001 16:01	001 16:14
5 CIP_SERVICE	SERVE_DET1	100.00	WAITOK	001 14:05	001 14:20
6 CL_T1	CLEAN_T1	20.00	WAITOK	001 14:23	001 14:43
7 CL_T3	CLEAN_T3	20.00	WAITOK	001 16:01	001 16:21
8 CL_T4	CLEAN_T4	20.00	WAITOK	001 16:21	001 16:41
SB002: BATCH INFORMATION			12:02 Sun 01 Jan 95		
Batch	BNo1_T3				
Type	PA_T1T3T4				
Size	89.00				
Status	EXECOK				
Allocations					
Resource	Type	Unit			
WATER_TANK	P	W_TANK			
	F	NOUNIT			
	R	NOUNIT			
TANK_1	P	T_1	Locked by batch		
TANK_3	P	T_3			
TANK_4	P	T_4			
	F	NOUNIT			
	R	NOUNIT			
Phases					
No. Name	Status	TF	Start	End	Units
1 START	FINIOK	1.00	001 12:01	001 12:01	
2 STEP_1	EXECOK	1.00	001 12:01	001 12:06	W_TANK T_1
3 STEP_2	WAITRESO	1.00	001 12:06	001 12:11	T_1
4 STEP_3	WAITRESO	1.00	001 12:11	001 12:21	T_1 T_3
5 STEP_4	WAITRESO	1.00	001 12:21	001 12:26	T_3
6 STEP_5	WAITRESO	1.00	001 12:26	001 12:41	W_TANK T_3
7 STEP_6	WAITRESO	1.00	001 12:41	001 13:55	T_3
8 STEP_7	WAITRESO	1.00	001 13:55	001 14:08	T_3 T_4
9 END	WAITRESO	1.00	001 14:08	001 14:08	

Table 7.5 Batch summary corresponding to Figure 7.7

SB001: BATCH SUMMARY

12:27 Sun 01 Jan 95

Batch	Type	Size	Status	Start	End
1 BNo1_T3	PA_T1T3T4	89.00	EXECOK	001 12:01	001 14:01
2 END_BNo1	EMPTY_T4	89.00	WAITOK	001 14:01	001 14:13
3 BNo2_T3	PA_T1T3T4	89.00	WAITOK	001 13:51	001 15:58
4 END_BNo2	EMPTY_T4	89.00	WAITOK	001 15:58	001 16:10
5 CIP_SERVICE	SER_DET1	100.00	WAITOK	001 14:05	001 14:20
6 CL_T1	CLEAN_T1	20.00	WAITOK	001 14:23	001 14:43
7 CL_T3	CLEAN_T3	20.00	WAITOK	001 15:58	001 16:18
8 CL_T4	CLEAN_T4	20.00	WAITOK	001 16:18	001 16:38

SB002: BATCH INFORMATION

12:28 Sun 01 Jan 95

Batch BNo1_T3
 Type PA_T1T3T4
 Size 89.00
 Status EXECOK

Allocations

Resource	Type	Unit
WATER_TANK	P	W_TANK
	F	NOUNIT
	R	NOUNIT
TANK_1	P	T_1
TANK_3	P	T_3 Locked by batch
TANK_4	P	T_4
	F	NOUNIT
	R	NOUNIT

Phases

No.	Name	Status	TF	Start	End	Units
1	START	FINIOK	1.00	001 12:01	001 12:01	
2	STEP_1	FINIOK	1.00	001 12:01	001 12:02	W_TANK T_1
3	STEP_2	FINIOK	1.00	001 12:02	001 12:06	T_1
4	STEP_3	FINIOK	1.00	001 12:06	001 12:14	T_1 T_3
5	STEP_4	FINIOK	1.00	001 12:14	001 12:19	T_3
6	STEP_5	EXECOK	1.00	001 12:19	001 12:34	W_TANK T_3
7	STEP_6	WAITRESO	1.00	001 12:34	001 13:48	T_3
8	STEP_7	WAITRESO	1.00	001 13:48	001 14:01	T_3 T_4
9	END	WAITRESO	1.00	001 14:01	001 14:01	

Table 7.6 Batch summary corresponding to Figure 7.8

Case Ib. Figure 7.11 shows a detailed schedule generated by SUPERBATCH for execution of the production plan for *Case Ib* at 12:00 on 1 January (start time). The expected finishing time is 11:50, 2 January. After five hour of operation, a failure was simulated to occur in vessel T1, which requires putting it out of service for eight hours. The supervisory controller readjusted the timing of all the batches including those allocated to T1 and for the rest of the operation (Figure 7.12). The dynamic rescheduling algorithm in SUPERBATCH (Cott, 1989) is based on a heuristic minimum time approach thus the schedule produced may not be optimal. For the case of equipment failure, the algorithm simply shifts backwards all the batches that are effected by the failure without changing the allocations. The execution of batch BNo3_T3 is not effected by the failure of vessel T1, so phase STEP_6 (corresponding to the reaction in T3 under the water jacket control) proceeds continuously.

The simple shift delay resulting from the rescheduling algorithm is rather pessimistic, since vessel T2 could also be used to carry out all the processing tasks that can be done in vessel T1. It is obvious desirable to reassign the processing tasks that are previously allocated to the vessel T1, rather than to delay all the rest batches and wait until T1 is back to the service. This might be performed by returning to the optimal production planning and scheduling level with the new conditions (Chua, 1995). However, the computation time for obtaining a feasible schedule for this case is too long (about five hours), thus effectively preventing any rescheduling to be carried out at this level. A reallocation of tasks and modification of the supervisory production schedule can be effectively done utilising the off-line portion of SUPERBATCH (Figure 7.6). This is easily initialised to reflect the current state of the plant/production. Various reallocation may be easily and quickly explored interactively. In this case, it is very obvious that re-allocating tasks previously scheduled in T1 to the vessel T2 is the only (and also optimal) solution. The SUPERBATCH production plan is then modified with new task allocation, verified for feasibility off-line first (with new initial conditions), then changes are passed to the on-line monitor. The new on-line process schedule at time 17:17 is shown in Figure 7.13, with new expected finishing time 12:02, 2 January.

Case II. Figure 7.14 shows a detailed SUPERBATCH schedule of the process at the initial time (12:00), with expected finishing time 17:10. By the time 13:08, the first five phases for BNo1_T3 have been completed while STEP_6 is being executed (Figure 7.15). BNo2_A has been completed and BNo2_B is being executed. This production was successfully completed at time 17:21 in this case.

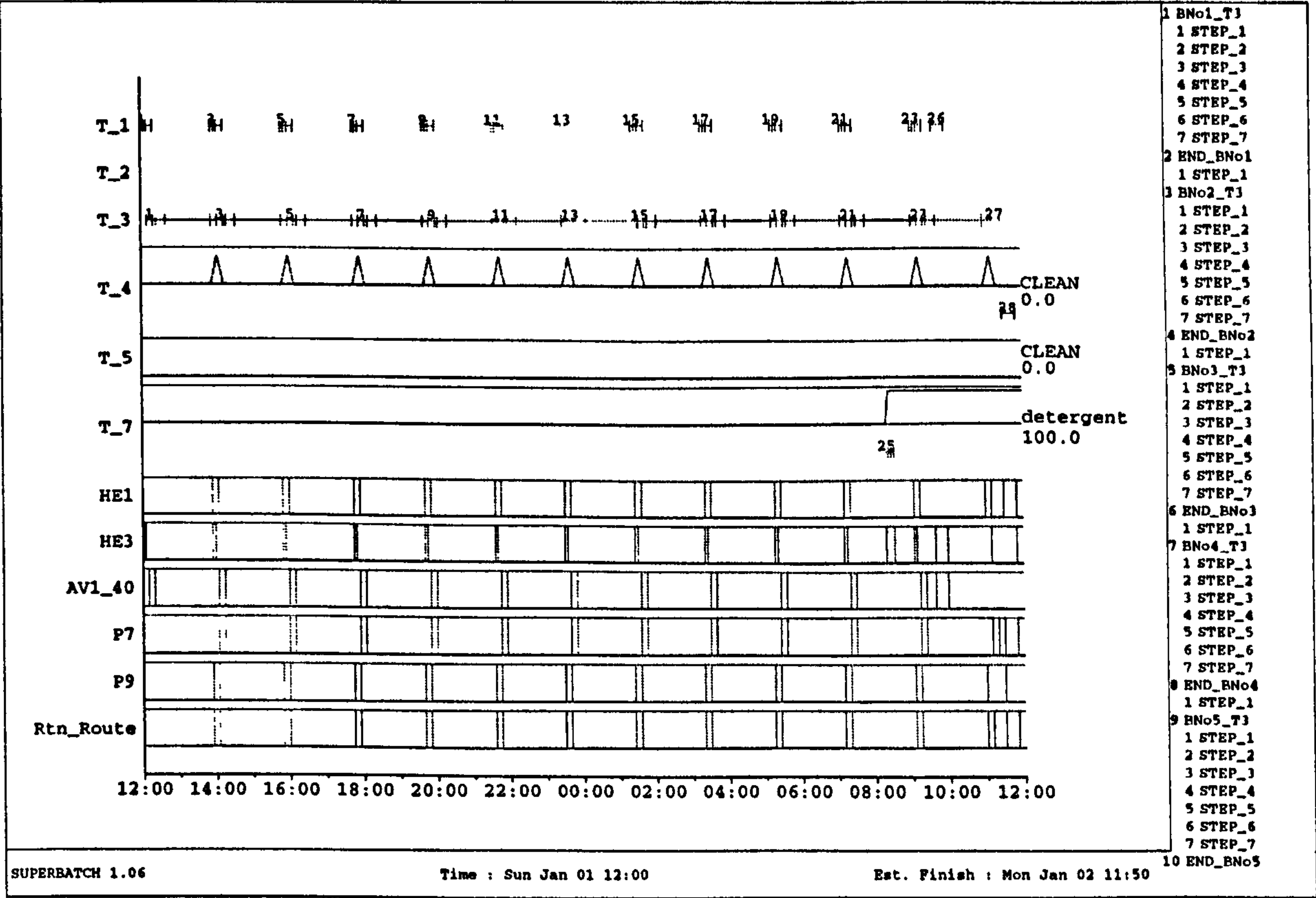


Figure 7.11 Case Ib - SUPERBATCH schedule at time 12:00

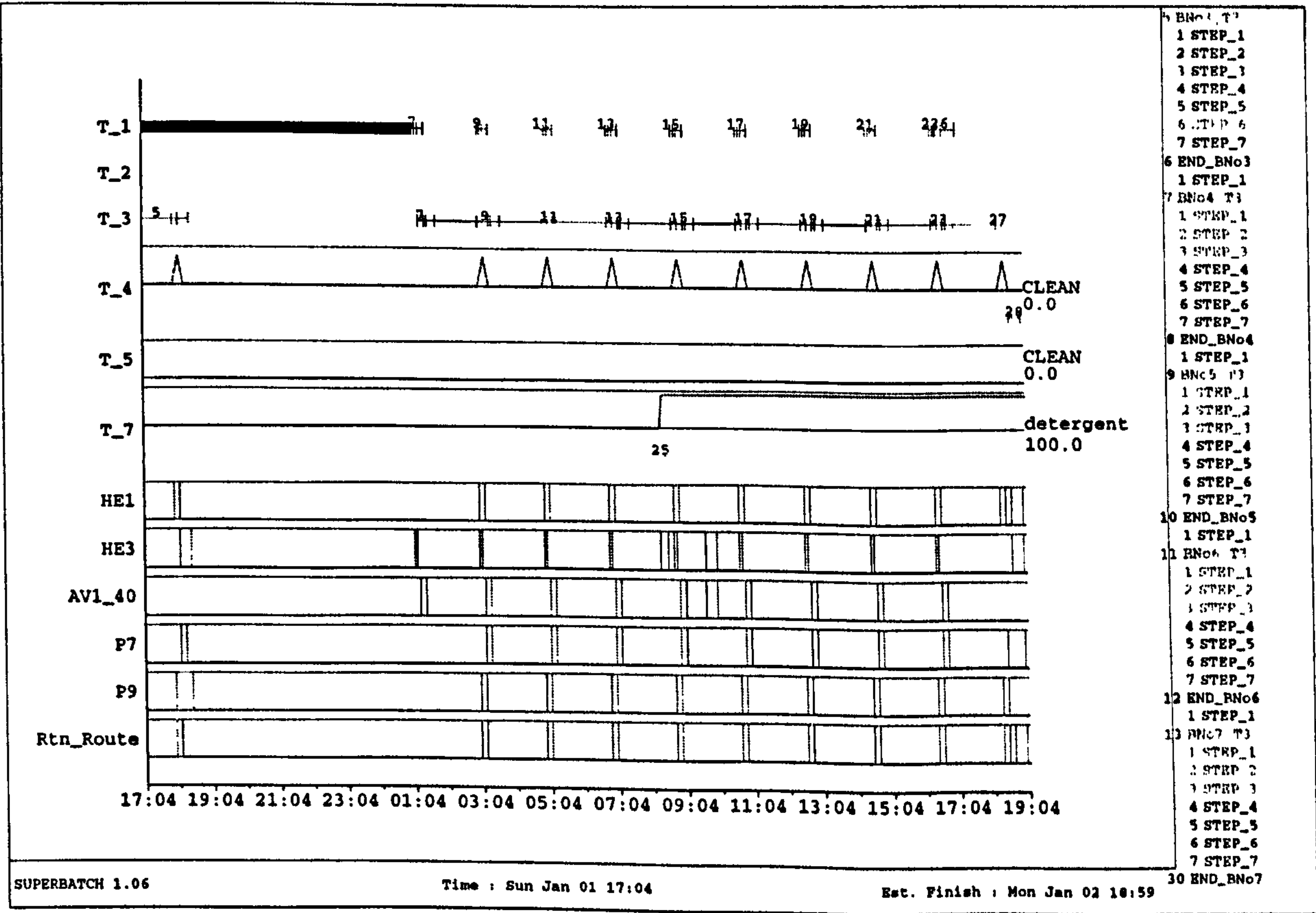


Figure 7.12 Case Ib - SUPERBATCH schedule at time 17:04, after vessel T1 failure.

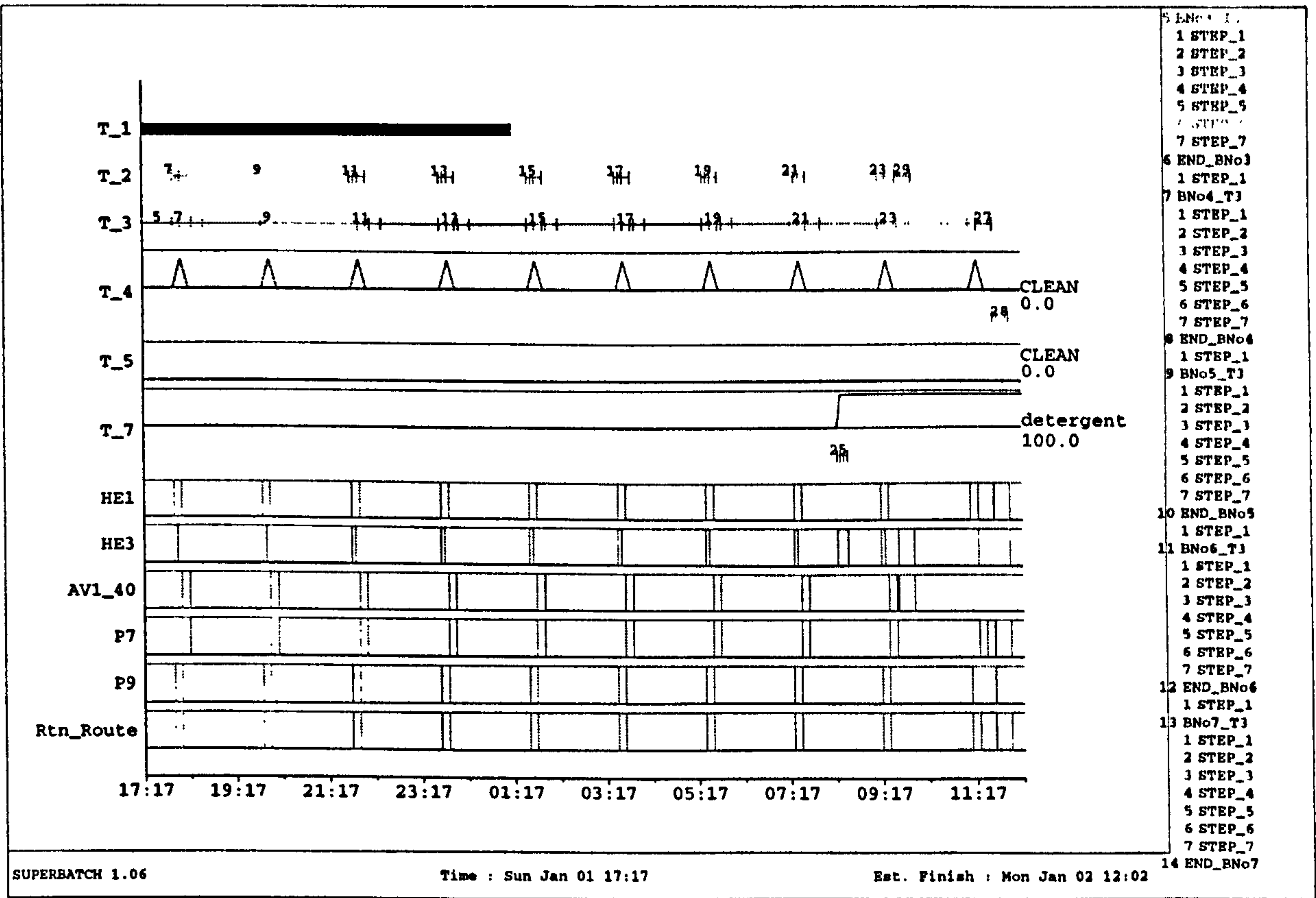


Figure 7.13 Case Ib - SUPERBATCH schedule at time 17:17 after reallocation of same tasks.

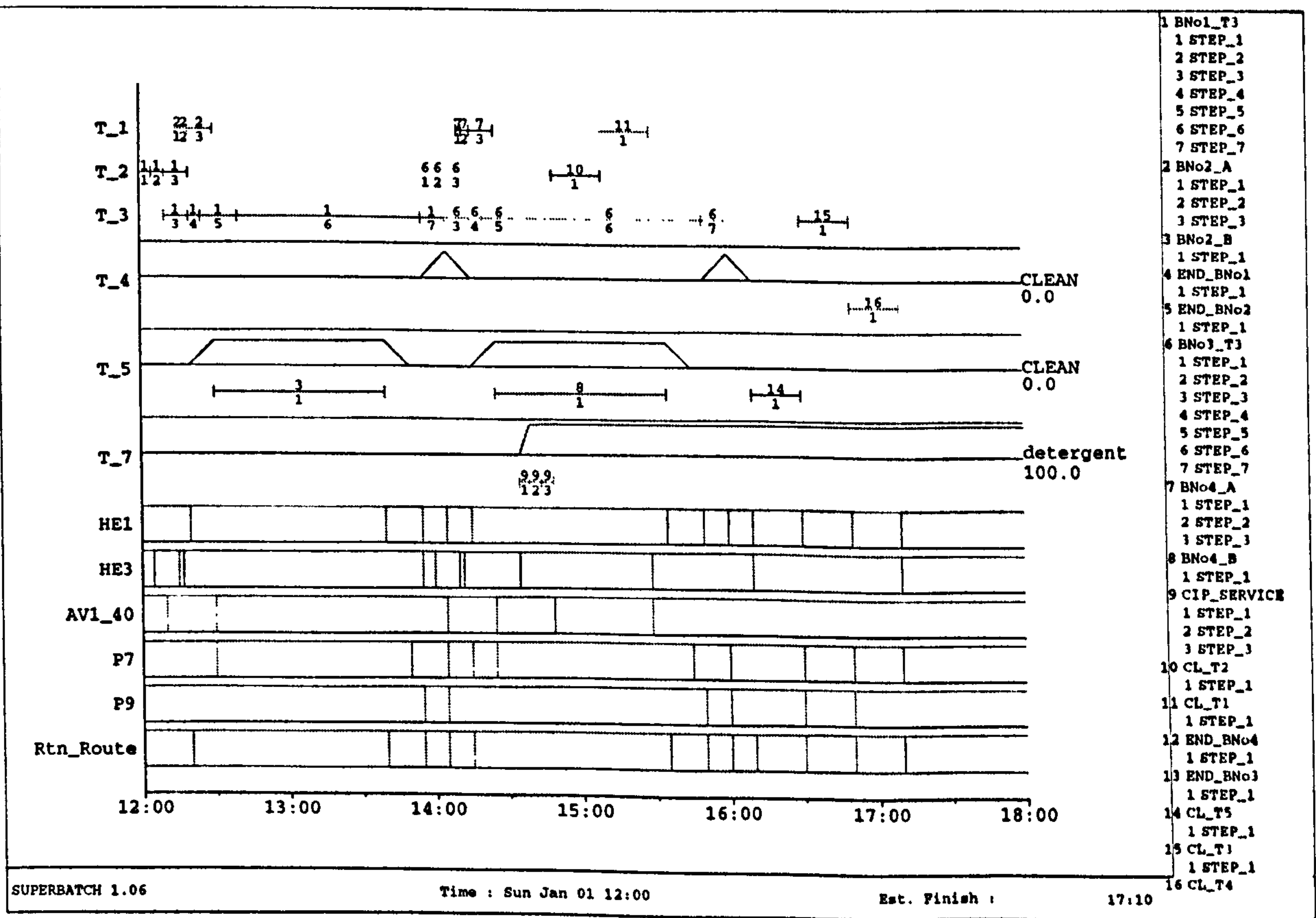


Figure 7.14 Case II - two products, SUPERBATCH schedule of at time 12:00.

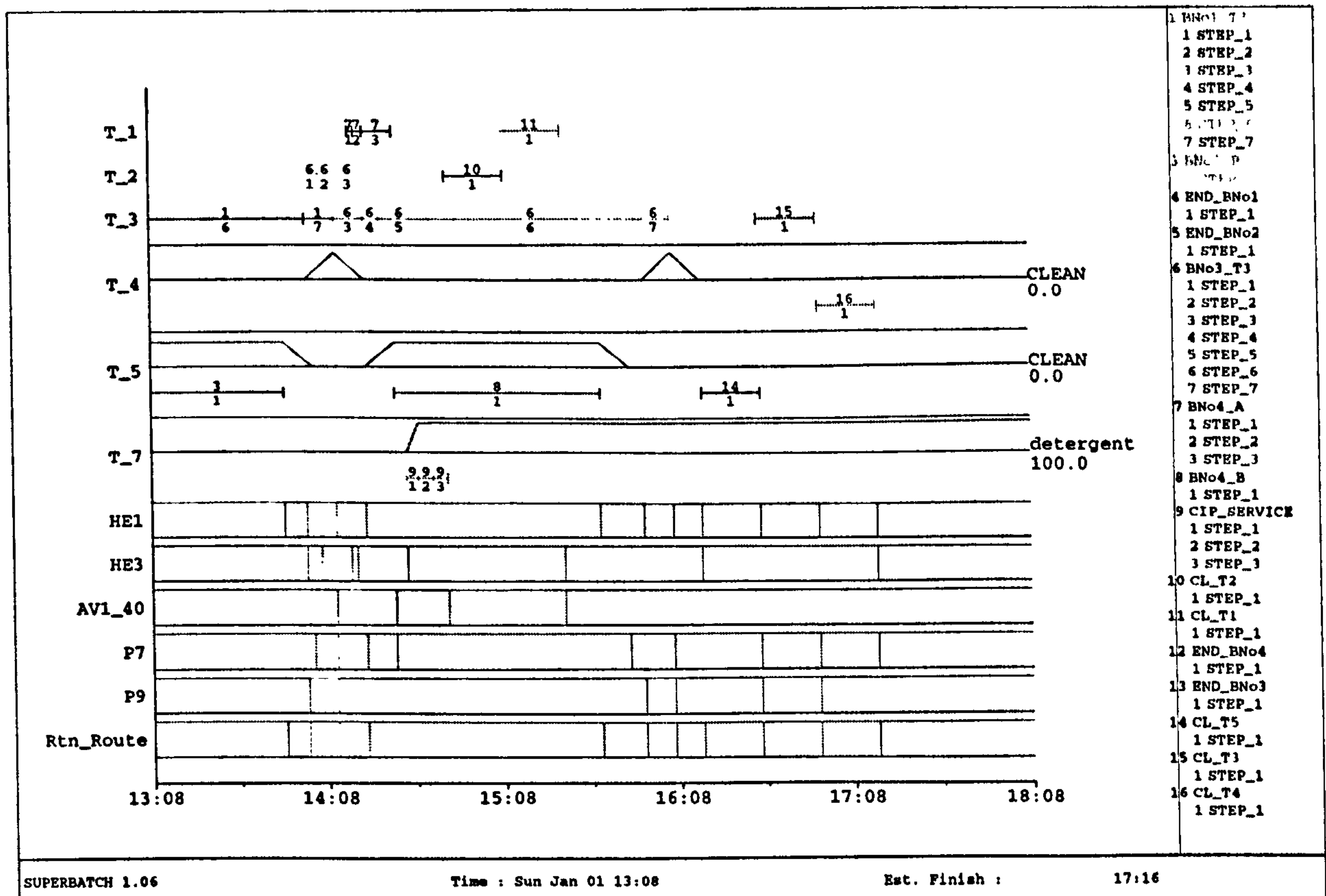


Figure 7.15 Case II - two products, SUPERBATCH schedule at start time 13:08.

7.3 Alternative Operating Procedure

The simulation of process schedules in the previous section indicate that the "Reaction_A" task which can only be performed in the reactor T3 (according to the initial process recipe) places the major bottleneck for the single product production cases. While the reactor T3 is busy performing the "Reaction_A" task, the vessels T4 and T5 are idle most of the time. Here we wish to show how the study of a process modification to remove the bottlenecks from the plant can be effectively undertaken utilising the tools and operations support environment discussed so far.

In the aggregated STN model (Chapter 4), the pH control, gelatinization stage in the aggregated "Reaction_A" task can only be carried out in the reactor T3 due to equipment and operation constraints. A pH probe is only fitted on T3, and the high temperature requirement for gelatinization can only be achieved in T3 by sparge heating. In the initial stage of the starch degradation step, viscosity of the starch gel is so high that the mixture cannot be

transferred into other vessels (that is, the reaction must be continued in T3). However, experimental results have shown that the viscosity decreases sharply after the initial stage of degradation, and the reactant can be transferred to other vessels approximately 15 minutes (a conservative value) after the degradation has started. Moreover, the reactor temperature becomes the only controlled variable in the degradation phase and the desired operating temperature is in a range of 78 - 85 °C. This offers the possibility of splitting the degradation phase into two subphases: *initial degradation* (high viscosity) in the reactor T3 and *further degradation* (low viscosity). The latter could be carried out either in T3 or in one of the product tanks T4/T5 in which the temperature control function may be achieved by recirculating the reactant through the external heat exchanger HE1. The original "Reaction_A" task shown in Figure 4.4.a is therefore decomposed into the "Reaction_S1" and "Reaction_S2" subtasks, with a low dextrins state produced by the first subtask (Figure 7.16). We may note that "Reaction_S2" can be performed in T3, T4 or T5.

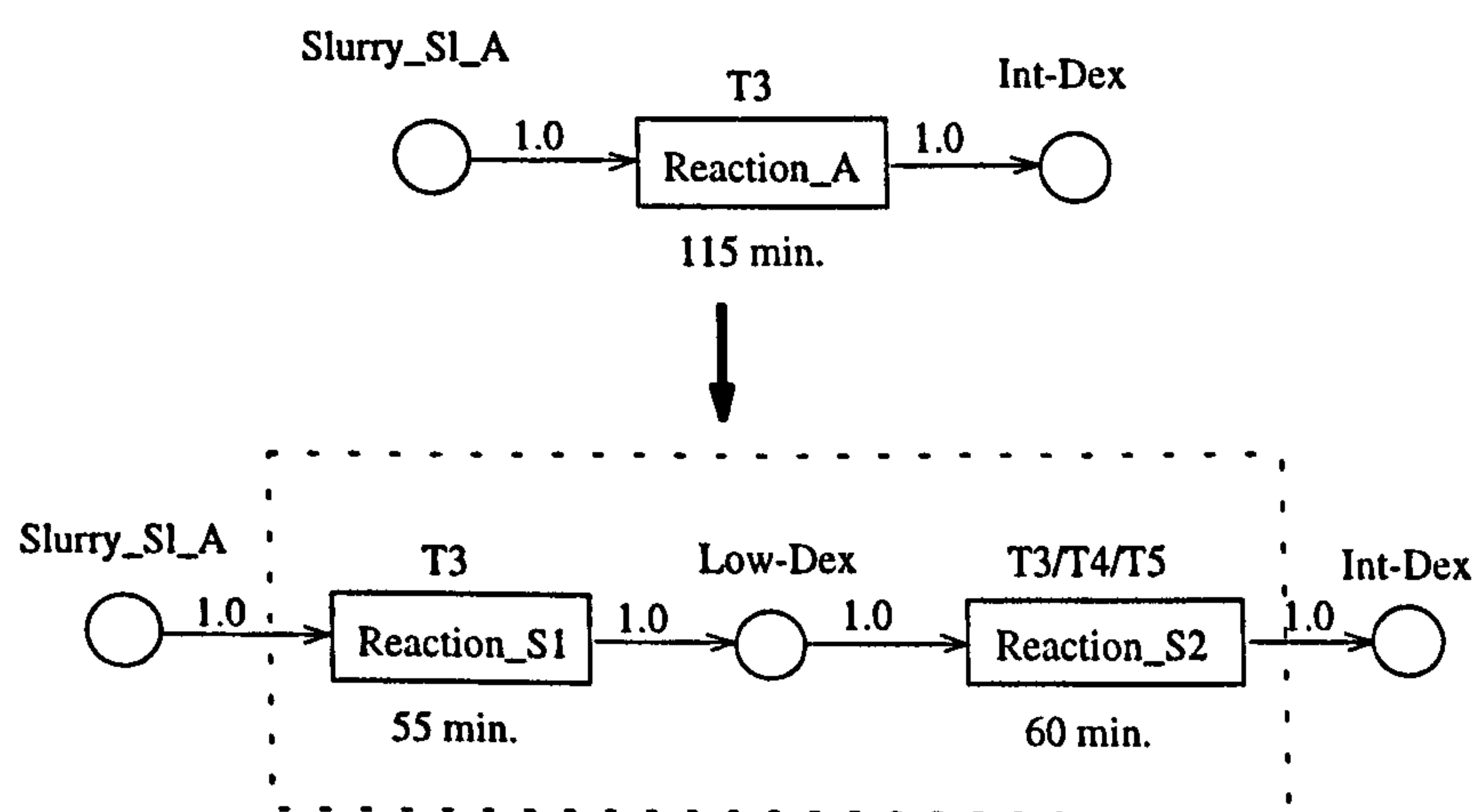


Figure 7.16 Decomposition of the reaction task into two subtasks

Case Ic. An optimal production schedule with the new product recipe and the same production requirements as *Case Ia* is shown in Figure 7.17. The corresponding utility profile is given in Figure 7.18. We may observe that the entire production requirement may now be scheduled in 205 minutes (as opposed to 270 minutes for the original case shown in Figure 7.2). Detailed SUPERBATCH schedules for this operations at the initial time (12:00) and as updated after 30 minutes of execution (at 12:30) are given in Figure 7.19 and 7.20, respectively. A trace of some continuous holdup measurements over the entire production time is shown in Figure 7.21. It may be noted that the same model based temperature controllers were used here as in all previous cases, with no retuning or other adjustments.

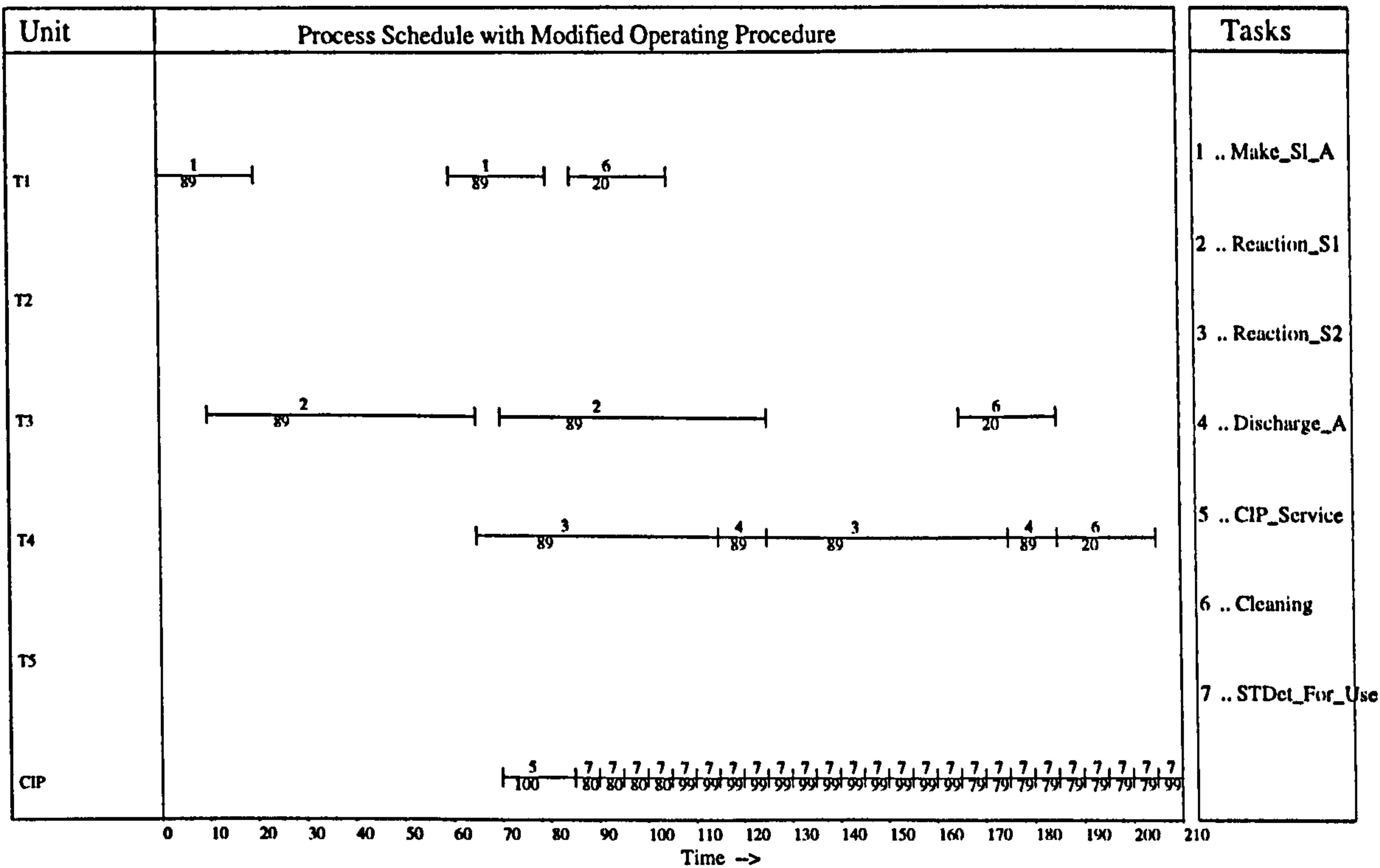


Figure 7.17 Case 1c - off-line optimal schedule with the new operating procedure

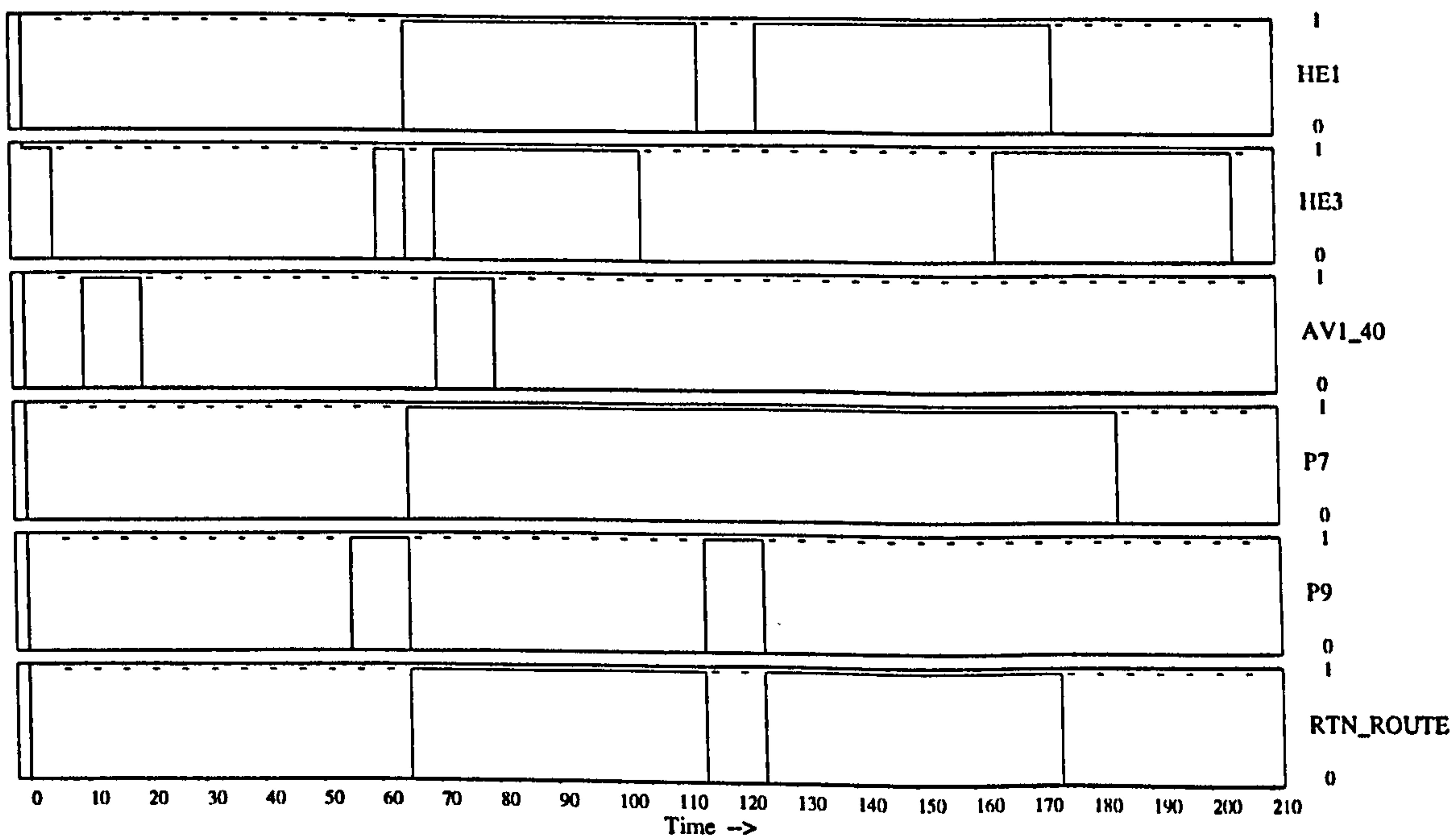


Figure 7.18 Utility profile with the new operating procedure

The execution of the production schedule for this case is finished successfully at time 15:42 (a total of 222 minutes). The actual execution of the corresponding operation with the original procedure required 298 minutes. Clearly, the modification of the procedure is advantageous in terms of production scheduling since it permits increasing the productivity of the plant (in this case by 26%) at zero investment cost. This could be very easily and quantitatively established using the operations support environment provided. Of course, a further experimental check ought to be performed in practice with regards to any effects of the new procedure on the quality of the products. These are not so easily simulated.

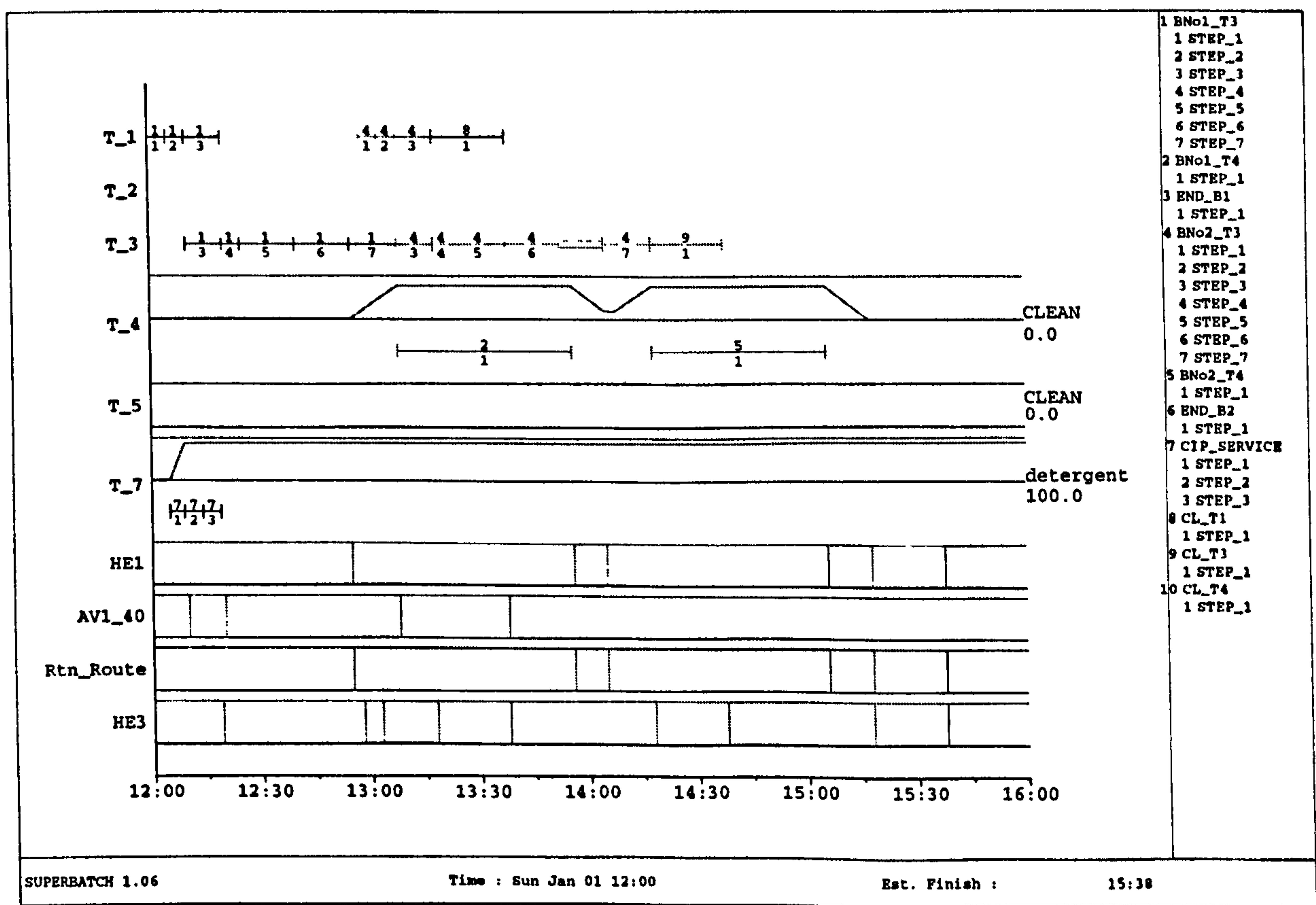


Figure 7.19 Case Ic - SUPERBATCH schedule with the new operating procedure at time 12:00

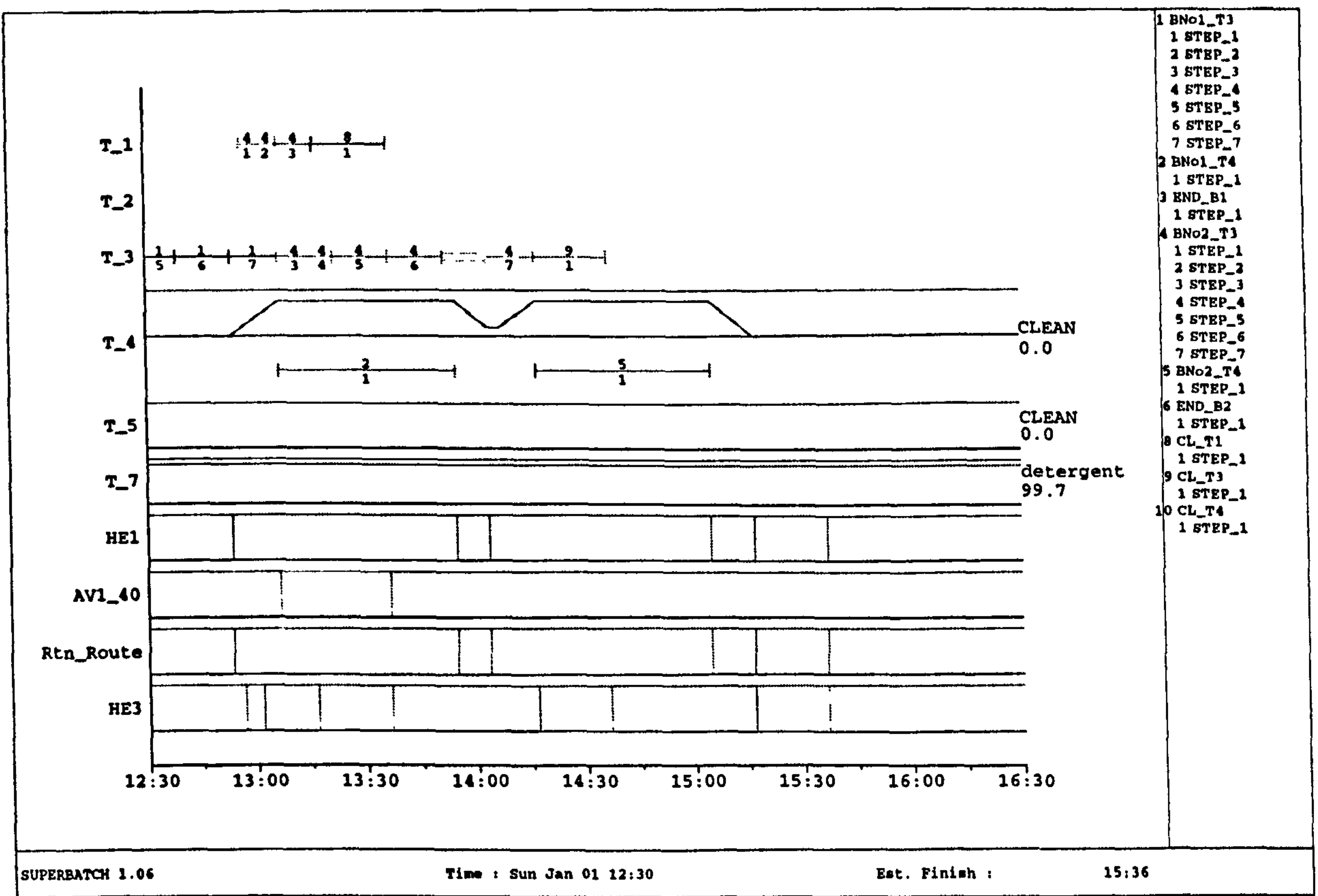


Figure 7.20 Case Ic - SUPERBATCH schedule with the new operating procedure at time 12:30

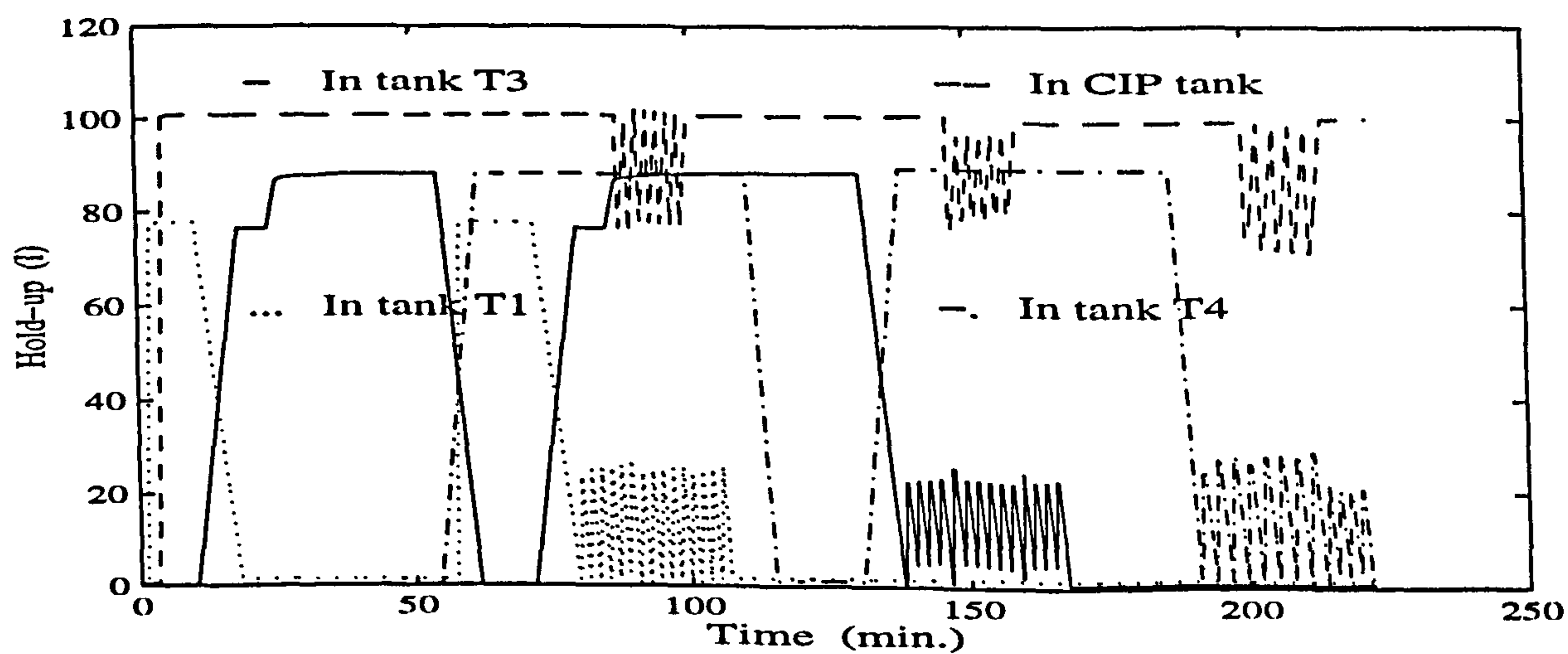


Figure 7.21 Case Ic - continuous holdup measurements over the entire production with the new operating procedures

Case III. A new production goal is set to deliver 5.0 tons of dextrans and 4.0 tons of cyclodextrins within one week with the new product recipes as discussed in the previous section. Again, intermediate cleaning is required only if a unit (or transfer route) is used to consecutively perform different tasks, and the plant must be cleaned at the end of production. In this case, no optimal solution of the MILP scheduling problem was found within 2880 minutes of CPU time. A detailed process schedule was therefore generated interactively using the SUPERBATCH system in which 57 batches of dextrans and 50 batches of cyclodextrins are scheduled by alternating individual batches of the two products. All dextrans batches are assigned to T_2, T_3 and T_4 while all cyclodextrins batches use T_1 and T_5. For this allocations and order, cleaning of all items is only required once, at the end of production. Figure 7.22 shows the schedule at 12:01 on 1 January (start time) for producing the first 7 batches of each product. The expected finishing time is 9:16, 8 January.

Figure 7.22 also shows that there is a high demands on shared resources (in particular, heat exchanger HE1 and the return route Rtn_Route), which represent the bottleneck of the operation.

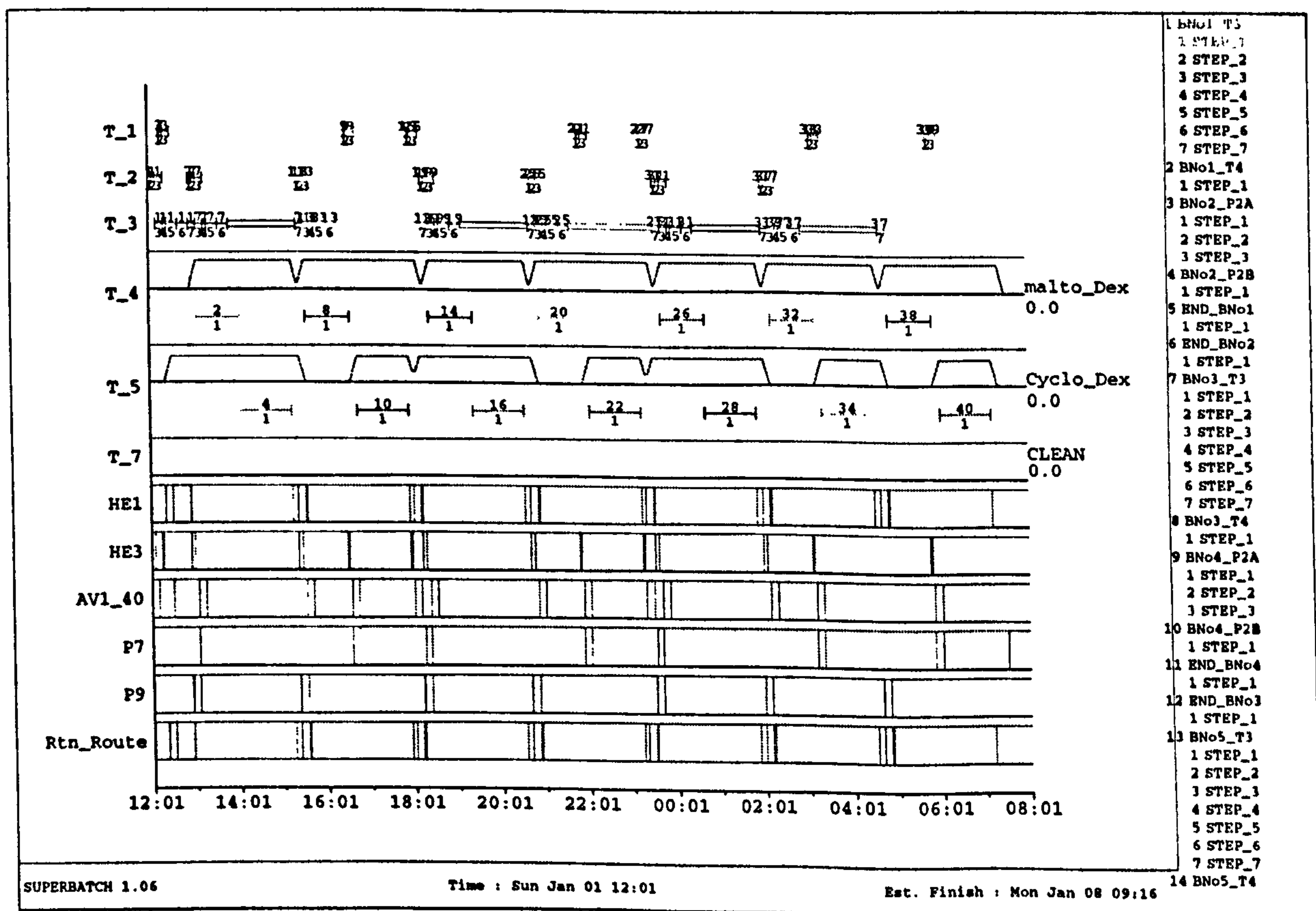


Figure 7.22 Case III - SUPERBATCH schedule at time 12:01 with new product recipe.

7.4 Summary

Chapter 7 has described the implementation and application of off-line optimal scheduling and the supervisory control for the operation of the batch pilot plant. The example in section 7.1 shows that even for a process with a small number of process steps, operation under operator control could be quite labour intensive and error-prone since there are many degrees of freedom.

Although continuous advances in computer hardware and in mathematical programming algorithms are enhancing the applicability of the optimal planning and scheduling systems, the size of the problems resulting from mathematical programming approaches is often still too large to be solvable at the present time. This supports the hierarchical design structure presented in Chapter 3 which includes a supervisory control system capable of operating autonomously, but also of accommodating "best" planning and scheduling decisions from various systems, even from operator inputs.

Conversely, the use of supervisory control enables the sequence level control to be carried out in a closed-loop manner. Apart from ensuring operational feasibility at all time, the supervisory controller leads to a considerable reduction of operator load. Another major advantage is that the integrity of the operation and the repeatability of the operation are assured by executing all processing procedures in the correct order, precisely and correctly at all time.

The production schedules in section 7.2 also indicated that the plant resources are poorly utilised for the single production case (Case I) under the original operating procedure. The new operating procedure, with the reaction phase being split into two sub-phases and using product vessels T4/T5 to perform part of the second sub-phase, led to substantial reduction of total processing time, more flexible operation and more effective utilisation of the plant. With the production goal of delivering 170 Kg of dextrins, the total time with the new procedure is 26% less than the reaction in T3 only, with a further reduction of up to 50% when producing more batches. Further experimental investigation of this new operating procedure on the product quality is needed which is considered as a future work.

The case studies demonstrated that the support environment developed can be used to quickly and reliably test and validate new manufacturing procedures.

SUPERBATCH is capable of handling most of the supervisory control problems in industry and is being developed as commercial software. Nevertheless, certain limitations still exist. The study in this chapter has led to the following suggestions which might enhance its power:

1. *Introduce interlocking facility among phases belong to different master procedures.* In the current version of SUPERBATCH, only precedence relationships between control phases within a master procedure can be specified. However, in some cases, control phases in different master procedures must be interlocked, either due to safety reason or restrictions of the plant configuration.
2. *Enable one or more control phases to be performed in an unit simultaneously.* The current version of SUPERBATCH does not allow two or more control phases being imposed on the same unit simultaneously which might be required by some processes (*e.g.* it would be desirable to carry out the detergent dosing and heating phases simultaneously in the CIP service operation). This may introduce extra time in the execution of production schedules.
3. *Distinguish master procedures for auxiliary operations (*e.g.* cleaning and service) from those for productions.* Some of the master procedures are not for producing products, rather for altering unit status (*e.g.* cleaning). Enforcing material balance constraints on such kind of operations is not only difficult, but also unnecessary since the purpose of such operations is not producing material products in required quantities. It is desirable to introduce another type of master procedure model in which mass balance constraints are released, however, with additional constraints on the conditions of execution (*e.g.* a master procedure for cleaning an unit should not be scheduled if the unit is clean already).
4. *Introduce precedence relationship among master procedures.* No precedence relationship between master procedures may be specified in the current version of SUPERBATCH. Therefore, a desired batch order in the production schedule may not be properly enforced. For example, the CL_T1 procedure in Figure 7.7 may be executed between the BNo1_T3 and BNo2_T3 in the production schedule. In this study, an additional constraint on the earliest start time was imposed on CL_T1 to ensure the cleaning operation is only performed after both BNo1_T3 and BNo2_T3 are finished (cleaning is only to be applied after the production).

Chapter 8

Conclusions and Future Work

In this thesis, the design, implementation and operation of an advanced process manufacturing system for a flexible batch pilot plant has been presented. The inherent variability and flexibility of batch processes increases the operational complexity. This underlines the need for the development of an operations support system for both on-line control implementation and off-line development support.

A hierarchical and model-based approach was considered in the design of the system. The system supports optimal production planning (off-line only), supervisory management and control, sequence control and advanced model based control either individually or in an integrated manner. High levels of aggregation of equipment and process models were utilised for optimal planning and scheduling so as to reduce the size of the scheduling problems, while a detailed dynamic model was developed and utilised to capture the more detailed phenomena of the plant. To ensure the feasible execution of any scheduled activities and production plans, and to ensure that local control decisions are properly propagated, a coordination of operational models at various levels of resolution was established. The development of flexible control sequences was also addressed. Better control of unit operations becomes more important in the integrated operation environments for the local suppression of disturbances and to provide a robust performance in response to process variations.

A partitioning-space design approach has been developed for the control of a class of nonlinear processes. The approach allows various controller design techniques to be used for nonlinear systems and results in a more flexible control system structure. The approach may be applied to partially known systems. In this case it makes it possible to make the best use of the

well known part of the system for the controller design. The effectiveness of the approach was demonstrated by simulation and experiment for the temperature control of a multipurpose batch reactor.

Dynamic models and simulation were used to assist in the development of the functional specification, implementation and verification of typical control sequences. Simulation was also used to identify potential improvements in existing control sequences and to develop new control procedures. A number of control sequences have been improved and new control sequences have been developed which led to reduced total processing times and improved operability.

Two levels of decision making were considered for production planning and control. Optimal production planning and scheduling based on MILP techniques was used for generating initial schedules *off-line*, while a minimum makespan algorithm was used for *on-line* production control and supervision. Operational feasibility and linking with the low level control system were ensured at the supervisory control level. The capability of the supervisory system to handle minor variations (*e.g.* in processing times) and major "disturbances" (*e.g.* equipment break downs) was demonstrated in the case study.

The batch pilot plant studies show that even for this small plant and well understood process, the potential exists for large improvements in productivity by refinement of the operation at all levels. The application of the advanced support system has led to improved utilization of plant resources, reduced total batch cycle time by 50%, and improved batch consistency and product quality through nonlinear model based control. In relation to the second objective (*i.e.* support of the development of new manufacturing procedures), the proposed system eases development, testing and implementation. Many new control procedures and controllers were tested and tuned firstly by simulation, leading to a safe and easy on-line commissioning and implementation.

Finally, the system structure proposed in this work has proved very effective for on-line implementation of the control procedures and control algorithms. It allows new controllers and control procedures to be evaluated by simulation which are easily implemented on the plant. As control activities at each level are directly coupled with all others, consideration of better controls at all levels was an important step towards a proper development of an integrated operations support system. Furthermore, experimental studies were essential to provide an early estimate of the potential economic impact of advanced operations support systems. This demonstrates the large opportunities for such systems in providing early feedback in the plant and process development cycles.

8.1 Future Work

Several potential areas for future work have been identified for the improvement of the operations support system described in this thesis.

The system presented is hierarchical and modular in structure. New control algorithms and control sequences can be added to the system without changing or modifying other parts of the system. Hence, it can be easily used for testing and validating new control procedures. However, the system should be extended to include monitoring and fault diagnosis functions. These could provide additional feedback to the supervisory control system, for example, so that better re-scheduling can be performed.

The partitioning-space approach developed here is only a first step. The key idea is to develop a control system structure which utilises the well known part of a system in the controller design. The approach can only be used for systems which satisfy the partitioning conditions. Many aspects are yet to be addressed or improved upon. For example, the current method relies on nonlinear control techniques used for each subsystem to handle constraints and lacks a generic way of handling constraints imposed on the state variables.

Dynamic simulation has proved useful for testing and validation of control sequences. However, modelling control sequences is labour intensive. It would be desirable to be able to more automatically develop detailed control sequences which are able to drive both the simulation and the plant control system. Some progress in this direction has been made (Sanchez, 1994).

Finally, in this work the optimal production plans were manually translated into SUPERBATCH production schedules and optimal on-line rescheduling was not considered due to practical reasons. Some preliminary results have been obtained in closing the gap between optimal planning and supervisory control levels (Crooks, 1992; Chua, 1995). However, these approaches require that the MILP models be sufficiently detailed (*e.g.* there must be broadly a one-to-one mapping between the STN tasks and the supervisory control phases). This often leads to large size scheduling problems, which makes optimisation difficult for on-line rescheduling at present. The need for improvement in the MILP solution times is recognised.

References

- ACCOS 30 *Traning Manual*. APV Baker, Crawley, U.K.
- Balchen, J. G., B. Lie and I. Solberg (1988). Internal decoupling in nonlinear process control. *Modelling, Ident. Control*, **9**, pp. 137-148.
- Barbosa-Povoa, A. and S. Macchietto (1994). Detailed design of multipurpose batch plants. *Computers and Chemical Engineering*, **18** (11/12), pp. 1013-1042.
- Barbosa-Povoa, A. (1994). Detailed design and retrofit of multipurpose batch plants. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Barolo, M., G. B. Guarise, S. Rienzi and A. Trotta (1992). GMC for startup control of distillation column. In: *Proceeding of ESCAPE-2*. France.
- Barolo, M. (1994). On the equivalence between the GMC and the GLC controllers. *Computers and Chemical Engineering*, **18** (8), pp. 769-772.
- Barton, P. and C. C. Pantelides (1991). The modelling and simulation of combined discrete/continuous processes. In: *Proceedings of PSE'91*, Vol. I, pp. 20.1-20.14. Montebello, Canada.
- Bartusiak, R. D., C. Georgakis and M. J. Reilly (1989). Nonlinear feedforward/feedback control structures designed by reference system synthesis. *Chemical Engineering Science*, **44** (9), pp. 1837-1851.
- Bassett, M. H, F. J. Doyle, G. K. Kudva, J. F. Pekny, G. V. Reklaitis, S. Subrahmanyam, M. G. Zentner and D. L. Miller (1994). Perspectives on model-based integration of process operations. In: *Proceeding of PSE'94*. Kyongju, Korea.
- Bequette, B. W. (1991). Nonlinear control of chemical processes: A Review. *Ind. Eng. Chem. Res.* **30**, pp. 1391-1413.
- Bhat, J., M. Chidambaram and K. Madhavan (1990). Robust control of batch reactors. *Chemical Engineering Communications*, **87**, pp. 195-204.

- Brown, M. W., P. L. Lee, G. R. Sullivan and W. Zhou (1990). A constrained nonlinear multivariable control algorithm. *Trans. Inst. Chem. Eng.* 68 (A), pp. 464-476.
- Chua, E. S. (1995). Integrated management system for multi-purpose batch chemical plants. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Clark, S. M. and K. Kuriyan (1989). BATCHES - Simulation software for managing semicontinuous and batch processes. *AIChE National Meeting*, No. 32f. Houston.
- Clark, S. M. and G. S. Joglekar (1992). General and special purpose simulation software for batch process engineering. In: *Nato Advanced Study Institute on Batch Processing Systems Engineering*. Antalya, Turkey, May 29-June 7.
- Clarke, E. M., E. A. Emerson and A. P. Sistla (1986). Automatic verification of finite state concurrent systems using temporal logic specifications. *ACM transactions on Programming Languages and Systems*, 8(2).
- Costello, D. (1992). GMC control of a strong acid-base neutralisation system. *Technical Report*. Chemical Engineering Department, University of Edinburgh, UK.
- Cott, B. J. and S. Macchietto (1989). Temperature control of exothermic batch reactors using Generic Model Control. *Ind. Eng. Chem. Res.*, 28, pp. 1177-1184.
- Cott, B. J., B. L. Ramaker, P. L. Lee and G. R. Sullivan (1989). Process model-based engineering. *Computers and Chemical Engineering*, 13 (9), pp. 973-984.
- Cott, B. J. (1989). An integrated computer-aided production management system for batch chemical processes. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Crooks, C. A., K. Kuriyan and S. Macchietto (1992). Integration of batch plant design, automation, and operation software tools. *Computers and Chemical Engineering*, 16S, pp. S289-S296.
- Crooks, C. A. (1992). Synthesis of operating procedures for chemical plants. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Cutler, C. R. and B. L. Ramaker (1979). Dynamic matrix control - a computer control algorithm. *AIChE National Meeting*. Houston, Texas.

- Czulek, A. J. (1988). An experimental simulator for batch chemical processes. *Computers and Chemical Engineering*, **12**, pp. 253-259.
- Daoutidis, P. and C. Kravaris (1989). Synthesis of feedback/state feedback controllers for nonlinear processes. *AIChE Journal*, **35**, pp. 1602-1616.
- Deguchi, P., Y. Yamasaki and H. Nishitani (1994). Operation support system using dynamic simulation for a combined batch/continuous plant. In: *Proceeding of ADCHEM'94*. Kyoto, Japan.
- Djavdan, P. (1992). Design of an on-line scheduling strategy for a combined batch/continuous plant using simulation. *Computers and Chemical Engineering*, **16**, pp. S281-288.
- Dore, S. D. (1993). Application of geometric nonlinear control in the process industries - A case study. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Egli, U. M. and D. W. T. Rippin (1986). Short-term Scheduling for Multiproduct Batch Plants. *Computers and Chemical Engineering*, **10**, pp. 303-325.
- Engell, S. and K. Wollhaf (1994). Dynamic simulation for improved operation of flexible batch plants. In: *Proceedings of Rutgers' Conference on Computer Integrated Manufacturing in the Process Industries*. Brunswick Hilton, East Brunswick, New Jersey, USA. April 25-26.
- Feyo De Azevedo, S., M. J. Goncalves, V. Ruiz and K. Najim (1989). Generalized predictive control of a batch evaporative crystallizer. In: *Proceeding of DYCORD'89* (J. E. Rijnsdorp, J. F. MacGregor, B. D. Tyreus, and T. T. Takamatsu, (Ed.)), pp. 93-98. Maastricht, The Netherlands.
- Fisher, D. G. (1991). Process control: An overview and personal perspective. *Can. J. Chem. Eng.*, **69**, pp. 5-26.
- Foss, B. A., T. A. Johansen and A. V. Sorensen (1994). Nonlinear predictive control using local models - Applied to a batch process. In: *Proceeding of ADCHEM'94*. Kyoto, Japan. X
- Fujii, M., S. Murakami, Y. Yamada, T. Ona and T. Nakamara (1981). A kinetic equation for hydrolysis of polysaccharides by mixed exo-and endoenzymes system. *Biotechnology and Bioengineering*, **23**, pp. 1393-1398.

- Garcia, C. E. and M. Morari (1982). Internal Model Control - a unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.*, **21**, pp. 308-323.
- Garcia, C. E. (1984). Quadratic dynamic matrix control of nonlinear processes. An application to a batch reactor process. *AIChE Annual Meeting*. San Francisco, CA.
- Garcia, C. E. and D. M. Prett (1987). Design methodology based on the fundamental control problem formulation. In: *Shell Process Control Workshop* (D. M. Prett, M. Morari, (Ed.)), pp. 225-250. Butterworth Publishers, Stoneham, MA.
- Garcia, C. E., D. M. Prett and M. Morari (1989). Model predictive control: theory and practice - a survey. *Automatica*, **25** (3), pp. 335-348.
- Grosdidier, P. (1987). Setpoint Inc., Houston. +
- Halasz, L., M. Hofmeister and D. W. T. Rippin (1992). GanttKit - an interactive scheduling tool. In: *Nato Advanced Study Institute on Batch Processing Systems Engineering*. Antalya, Turkey, May 29-June 7.
- Henson, M. A. and D. E. Seborg (1990). Input-output linearization of general nonlinear processes. *AIChE Journal*, **36** (11), pp. 1753-1757.
- Henson, M. A. and D. E. Seborg (1991). Critique of exact linearization strategies for process control. *Journal of Process Control*, **1**, May, pp. 122-139.
- Hofmeister, M., L. Halasz and D. W. T. Rippin (1989). Knowledge-based tools for batch processing systems. *Computers and Chemical Engineering*, **13** (11/12), pp. 1255-1261.
- Isidori, A. (1989). *Nonlinear Control Systems: an introduction*, 2nd Edt. Springer-Verlag, Berlin.
- Ishii, N. and M. Muraki (1993). A process-variability-based online scheduling system in multiproduct batch process. Personal communications.
- ISA SP88 (1994). Batch Control Systems - Models and Terminology. Draft 11. Instrument Society of America.

- Juba, M. R. and J. W. Hamer (1986). Progress and challenges in batch process control. In: *Proceedings of the Third International Conference on Chemical Process Control* (M. Morari and T. McAvoy, (Ed.)). Elsevier Science Publishers, New York.
- Kantor, J. C. (1986). An overview of nonlinear geometrical methods for process control. In: *Shell Process Control Workshop* (D. M. Prett, M. Morari, (Ed.)), pp. 225-250. Butterworth Publishers, Stoneham, MA.
- Kershenbaum, L. S. and P. Kittisupakorn (1994). An experimental testing of model predictive control algorithms using a partially simulated exothermic (PARSEX) reactor. *AIChE 1994 Annual Meeting*. San Francisco, California, November.
- Kondili, E. (1988). Optimal scheduling of batch chemical processes. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Kondili, E., C. C. Pantelides and R. W. H. Sargent (1993). A General Algorithm for Scheduling Batch Operations. *Computers and Chemical Engineering*, **17** (2), pp. 211-227.
- Kravaris, C. and C. B. Chung (1987). Nonlinear state feedback synthesis by global input/output linearization. *AIChE Journal*, **33** (4), pp. 592-603.
- Kravaris, C. and S. Palanki (1988). Robust nonlinear state feedback under structured uncertainty. *AIChE Journal*, **34**, pp. 1119-1127.
- Kravaris, C., R. A. Wright and J. F. Carrier (1989). Nonlinear controllers for trajectory tracking in batch processes. *Computers and Chemical Engineering*, **13**, pp. 73-82.
- Kuriyan, K., G. E. Rotstein, A. Sanchez, C. A. Crooks and S. Macchietto (1994). Integrated operation of batch chemical processes, *IMSE'94*. Grenoble, France, December.
- Le Lann, M. V., M. Cabassud and G. Casamatta (1992). Adaptive model predictive control. In: *NATO Control On Batch Processing Systems Engineering*. Antalya, Turkey.
- Lee, P. L. and G. R. Sullivan (1988). Generic Model Control. *Computers and Chemical Engineering*, **12** (6), pp. 573-589.
- Lee, P. L., R. B. Newell and G. R. Sullivan (1989a). Generic model control - A case study. *Can. J. Chem. Eng.* **67**, pp. 478-484.

- Lee, P. L., G. R. Sullivan and W. Zhou (1989b). Process/model mismatch compensation for model-based controllers. *Chemical Engineering Communications*, 80, pp. 33-51.
- Love, J. (1991). Automatic generation of application software. In: *Proceedings of PSE'91*, Vol. I, 15.1-15.8. Montebello, Canada.
- Love, J., B. Sheil and N. Drakos (1992). Sequence synthesis for batch process control, *Journal A, Benelux Quarterly Journal on Automatic Control*, 1, pp. 42-51.
- Macchietto, S., C. A. Crooks and K. Kuriyan (1992). An integrated system for batch processing. In: *Nato Advanced Study Institute on Batch Processing Systems Engineering*. Antalya, Turkey, May 29-June 7.
- Macchietto, S. (1992). Automation research on a food processing pilot plant. *ICHEME Symposium Series*, 126, pp. 179-189.
- Macchietto, S. (1993). Bring the Gap - Integration of design, operation scheduling and control, In: *FOCAPO II*, Colorado, July.
- Marino, R., W. Respondek and van der Shaft (1989). Almost disturbance decoupling for single-input single-output nonlinear systems. *IEEE Transactions on Automatic Control*, 34 (9), pp. 1013-1018.
- Marroquin, G. and W. L. Luyben (1972). Experimental evaluation of nonlinear cascade controllers for batch reactors. *Chemical Engineering Science*, 28, pp. 993-1003.
- Mather, T. W. and H. E. Garner (1989). Implementing CIM in an existing facility. *Hydrocarbon Processing*, pp. 43-49.
- Moon, I., G. Powers, J. R. Burch and E. M. Clarke (1992). Automatic verification of sequential control systems using Temporal Logic. *AIChE Journal*, 38 (1), pp. 67-75.
- Moon, I. and S. Macchietto (1994). Formal verification of batch processing control procedures. In: *Proceeding of PSE'94*, pp. 469-475. Kyongju, Korea.
- Moore, R. (1993). The Gensym dynamic scheduling package. In: *Industrial Consortium Meeting*, Centre for Process Systems Engineering, Imperial College, London.

- Nakamoto, K. and N. Watanabe (1991). Multivariable control experiments of non-linear chemical processes using non-linear feedback transformation. *Journal of Process Control*, **1**, pp. 140-145, May.
- Oliveira, S. and M. Morari (1994). Robust model predictive control for nonlinear systems with constraints. In: *Proceeding of ADCHEM'94*, pp. 295-300. Kyoto, Japan.
- Pantelides, C. C. (1993). Unified framework for optimal process planning and scheduling. In: *Third Intl. Conf. on Foundations of Computer-Aided Proce. Des.* Snowmass, Colorado.
- Pekny, J., V. Venkatasubramanian and G. V. Reklaitis (1991). Prospects for computer-aided process operations in the process industries. In: *Proceeding of COPE'91*. Barcelona, Spain.
- Peterson, T., E. Hernandez, Y. Arkun and F. J. Schork (1992). A nonlinear DMC algorithm and its application to a semi-batch polymerization reactor. *Chemical Engineering Science*, **47** (4), pp. 737-753.
- Ranky, P. G. (1986). *Computer Integrated Manufacturing: an introduction with case studies*. Englewood Cliffs: Prentice-Hall.
- Rawlings, J. B., N. F. Jerome, J. W. Hamer and T. M. Bruemmer (1989). End-point control of semi-batch chemical reactors. In: *Proceeding of DYCORDER'89*, pp. 339-344. Maastricht, The Netherlands.
- Rawlings, J. B., E. S. Meadows and K. R. Muske (1994). Nonlinear model predictive control: a tutorial and survey. In: *Proceeding of ADCHEM'94*, pp. 203-214. Kyoto, Japan.
- Reklaitis, G. V. (1992). Overview of scheduling and planning of batch process operations. In: *Nato Advanced Study Institute on Batch Processing Systems Engineering*. Antalya, Turkey, May.
- Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, **29** (5), pp. 1251-1274.
- Richalet, J. (1994). Model based predictive control. In: *Automatic Control of Food and Biochemical Processes* (J. J. Bimbenet, E. Dumoulin and G. Trystram, (Ed.)), pp. 253-264. Elsevier Science B. V., Paris, France.

- Ricker, N. L. (1991). Model-predictive control: state of the art. In: *Proceedings of the Fourth International Conference on Chemical Process Control* (Y. Arkun and W. H. Ray, (Ed.)). Padre Island, Texas.
- Rippin, D. W. T. (1989). Control of batch processes. In: *Proceeding of DYCORD'89*, pp. 131 - 141. Maastricht, The Netherlands.
- Rippin, D. W. T. (1991). Batch process planning. *Chemical Engng.*, **98** (5), pp. 100-107.
- Rippin, D. W. T. (1993). Batch process systems engineering: a retrospective and prospective review. *Computers and Chemical Engineering*, **S17**, pp. s1-s13.
- Rosenof, H. P. and A. Ghosh (1987). *Batch process automation - theory and practice*. Van Nostrand Reinhold Company Inc. New York.
- Rotstein, G. E., A. Sanchez, N. Alsop, L. Solfaroli Camillocci and S. Macchietto (1995). Synthesis of procedural controllers and the automatic generation of sequential control code. In: *Workshop on Analysis and Design of Event-Driven Operations*. Imperial College, London. April.
- Rouhani, R. and R. K. Mehra (1972). Model algorithmic control: basic theoretic perspectives. *Automatica*, **18**, pp. 401-414.
- Sanchez, A. (1994). Formal specification and synthesis of sequential/logic controllers for process systems. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Sawyer, P. (1993). *Computer-Controlled Batch Processing*. IChemE, Rugby, UK.
- Schuler, H., F. Allgowerm and E. D. Gilles (1991). Chemical process control: present status and future needs - The view from European industry. In: *Chemical Process Control - CPC IV* (Y. Arkun and W. H. Ray, (Ed.)).
- Shah, N. (1992). Efficient scheduling, planning and design of multipurpose batch plants. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London.
- Singh, H. and E. G. Mahler (1994). Integrating decision support systems with legacy systems. In: *Proceedings of Rutgers' Conference on Computer Integrated Manufacturing in the Process Industries*. Brunswick Hilton, East Brunswick, New Jersey, USA. April 25-26.

- Slotine, J. E. and W. P. Li (1991). *Applied nonlinear control*. Prentice-Hall, Englewood Cliffs.
- Sorosh, M. and C. Kravaris (1992). Nonlinear control of a batch polymerization reactor: an experimental study. *AIChE Journal*, 38 (9), pp. 1429 - 1448.
- Stephanopoulos, G. (1984). *Chemical Process Control*. Prentice Hall, Englewood Cliffs, NJ.
- SUPERBATCH User's Guide V1.06 (1994). Imperial College, UK.
- Terpstra, V. J., R. M. Bruijkere and H. B. Verbruggen (1994a). On-line schedule optimization for mixed batch/continuous plants. In: *Proceeding of ADCHEM'94*. Kyoto, Japan.
- Terpstra, V. J., R. M. Bruijkere and H. B. Verbruggen (1994b). Robustness in reactive batch scheduling. In: *Proceedings of PSE'94*, pp. 503-506. Kyongju, Korea.
- Williams, T. J. (1987). The integrated approach to industrial control. In: *Proceedings of the 10th World Congress of the International Federation of Automatic Control*, pp. 208-216. Munich, Germany.
- Valliere, P. De and D. Bonvin (1989). Application of estimation techniques to batch reactors - II. Experimental studies in state and parameter estimation. *Computers and Chemical Engineering*, 13 (1/2), pp. 11-20.
- Venkatasubramanian, V. and G. M. Stanley (1993). Integration of process monitoring, diagnosis and control: issues and emerging trends. In: *Proceeding of Foundations of Computer Aided Process Operations (FOCAPO'93)*. Colorado, July.
- Yamalidou, E. C., E. P. Patsidou and J. C. Kantor (1990). Modelling discrete-event dynamical systems for chemical process control - A survey of several new techniques. *Computers and Chemical Engineering*, 14 (3), pp. 281-299.
- Yamamoto, S. and I. Hashimoto (1991). Present status and future needs: the view from Japanese industry. In: *Proceeding of the 1991 American Control Conference*, pp. 372-376.
- Zentner, M. G. and G. V. Reklaitis (1992). An interval-based mathematical model for scheduling of resource-constrained batch chemical processes. In: *Nato Advanced Study Institute on Batch Processing Systems Engineering*. Antalya, Turkey, May 29-June 7.
- Zentner, M. G., J. F. Pekny, D. L. Miller and G. V. Reklaitis (1994). RCSP++: a scheduling system for the chemical process industry. In: *Proceedings of PSE'94*. Kyongju, Korea. May.

- Xia, Q. S. and S. Macchietto (1994). Routing, scheduling and product mix optimisation by Minmax Algebra models. In: *Proceeding of ESCAPE-4*. Ireland.
- Zhang, X. and R. W. H. Sargent (1994). The optimal operation of mixed production facilities - a general formulation and some approaches for the solution. In: *Proceedings of PSE'94*, pp. 171-177. Kyongju, Korea.

Appendices

A. Implementation of Nonlinear Controllers

In order to enable mathematical manipulation of controller outputs or the replacement of a controller output by model based controllers, analogue loop outputs from conventional PID controllers are not sent directly to the output devices (control valves, secondary loop setpoint, etc.) but read into registers by control sequences (Seq 1.117 in sparge heating control, Seq 1.104 and Seq 1.118 in the water jacket control) through *control output ghost channels* (analogue inputs). In all configurations discussed below, bold solid lines with arrows show the control configuration using the model-based controllers, while the hair solid lines with arrows are the conventional configuration using PID controllers.

In all experiments, a sampling time of 10 seconds is used. In each sampling interval, measurements (*e.g.* reactor and jacket temperature, recirculation flow rate) are obtained by the model based controllers (through GET command in the gPROMS simulator) from the plant. The model (controller) is then integrated forward 10 seconds in simulation time (CONTINUE). The synchronisation between the simulation time and the real time is done by a PAUSE which is triggered by a real system timer and reset/restarted when a sampling interval has expired. The control actions are sent to the control system using a SEND command. A complete sampling and control cycle is shown in Figure A.1.

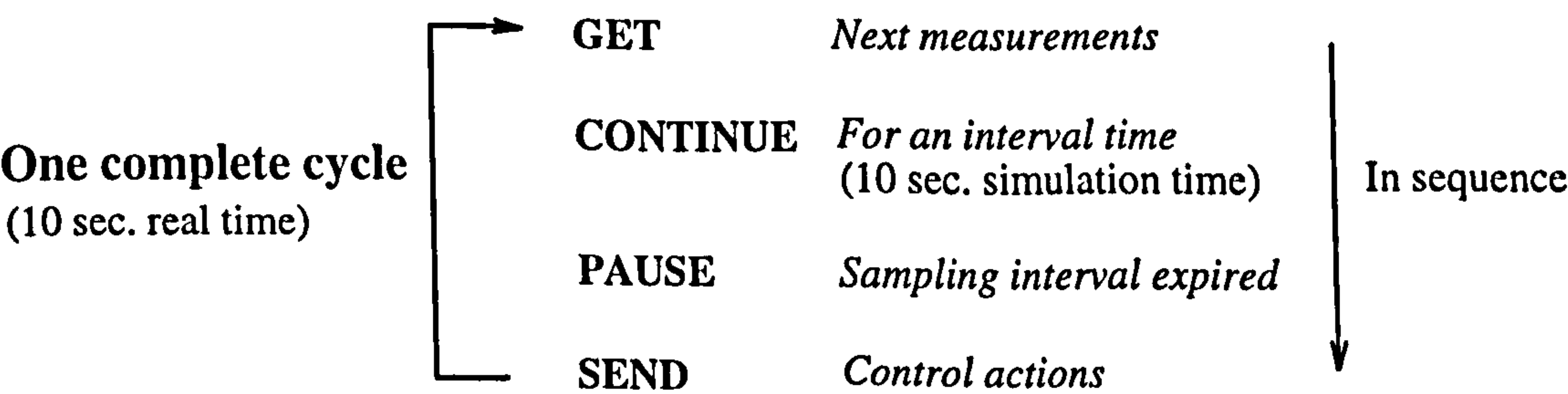


Figure A.1 A complete sampling cycle

A.1 Model-based Controllers for Sparge Heating

The loop configuration for the implementation of GMC/GLC controllers for the sparge heating control is shown in Figure A.2. The original configuration is such that the analog control loop with a PID controller is enabled/disabled by the control sequences (Seq 1.78 and Seq 1.103) when the sparge heating control is started. The output of the PID controller (channel 1.11.0) is not sent directly to the output device (control valve, CV1-3). Instead, the signal from a *Ghost Channel* (channel 1.12.1, the output of which is equal to the value in channel 1.11.0) is read into a register by Seq 1.117 before being sent to the control valve. With model based nonlinear controllers, the desired control valve position (0-100%) is calculated by the nonlinear controllers and stored in a floating point register F2.8. The value is then directed to the control valve CV1-3 through an analog output card (channel 1.3.0) by sequence Seq 1.117. The data (*e.g.* measurements, control actions) between the controllers and control system are transferred through the Control Link server as discussed in Chapter3.

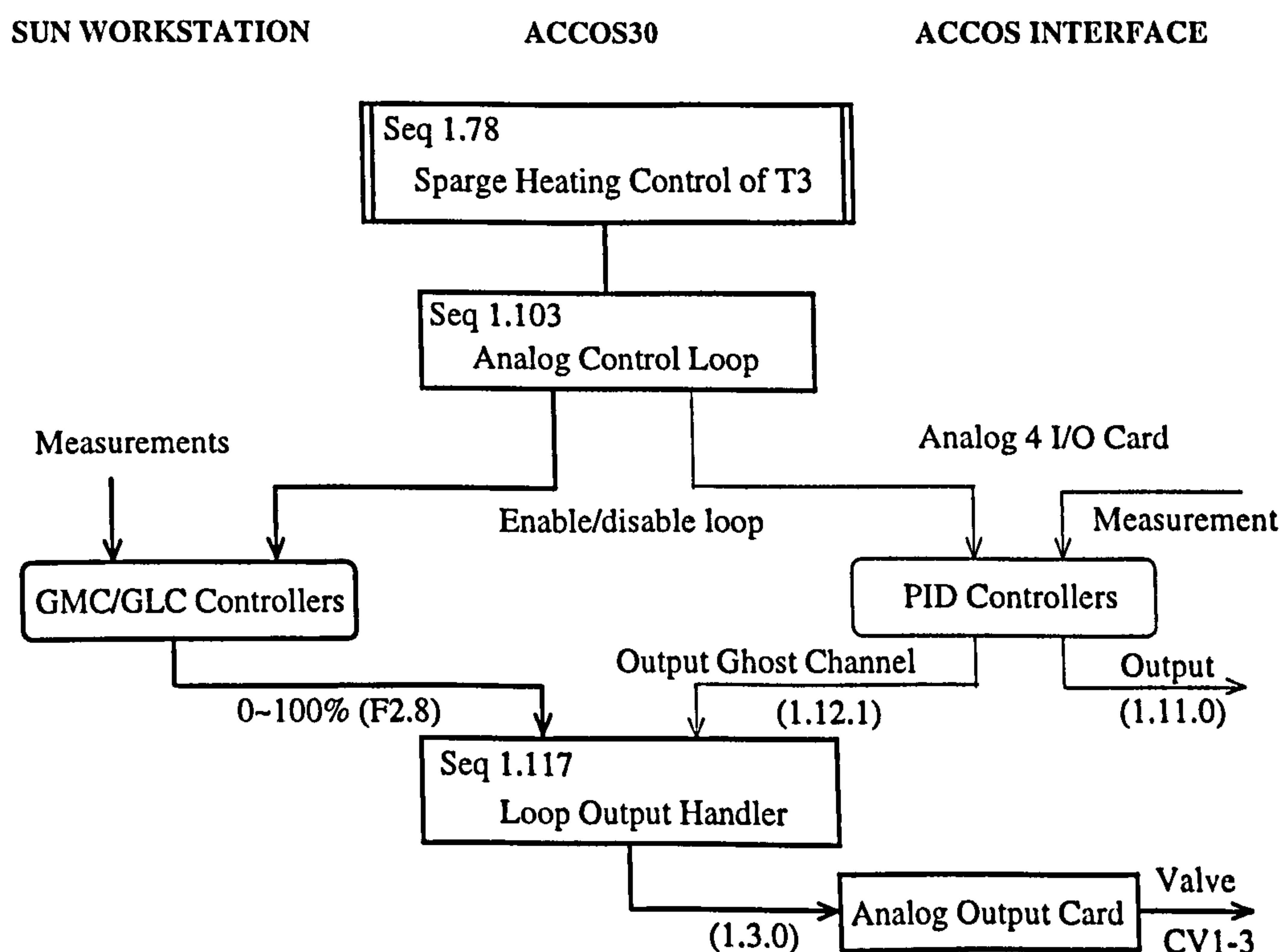


Figure A.2 Implementation of nonlinear controllers for sparge heating control

A.2 Cascade GMC/GLC and PI Controllers

A loop configuration for the implementation of the cascade GMC/GLC and PI controllers in the water jacket control is shown in Figure A.3. The GMC/GLC controller is enabled when the sequence Seq 1.80 (Water Jacket Control of T3) is started. The output of the GMC/GLC controller is a desired jacket temperature setpoint for the secondary loop PI controller. This value is read into a register (F2.43), then sent to the conventional PI controller by the sequence Seq 1.104 through the channel 1.11.3. The output of the PI controller (ghost channel 1.12.3) is directed to the cooling water (CV1-7) and steam (CV1-4) valves (set in a split range manner, see section 4.5) through sequence Seq 1.118 and an analog output card (channel 1.3.1).

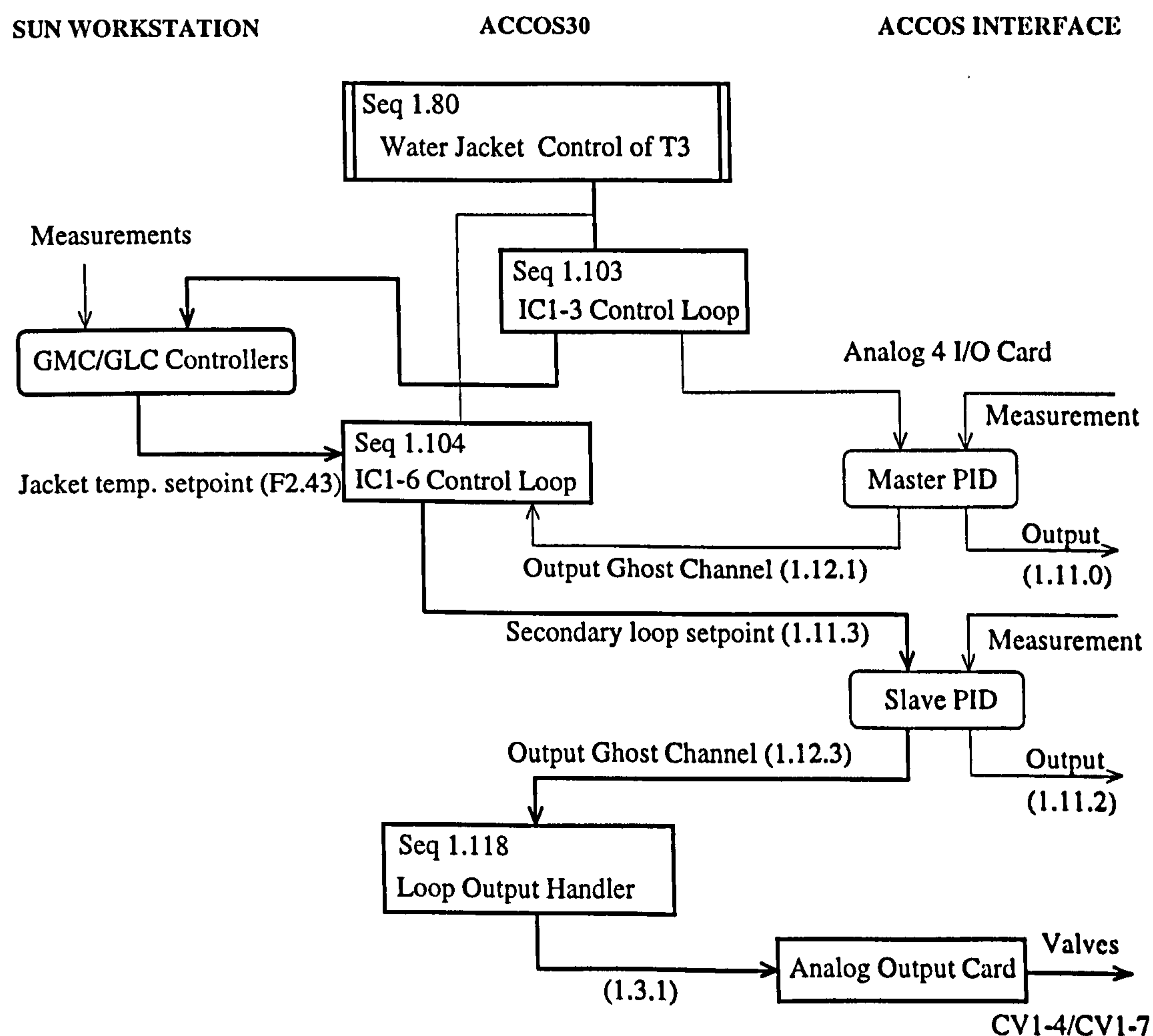


Figure A.3 Implementation of cascaded GMC/GLC and PI controller for water jacket control

A.3 Cascade GMC Controllers

Figure A.4 shows the loop configuration for the implementation of the cascade GMC controllers for water jacket control.

The arrangement is similar to that in Figure A.3, however, the output of the primary GMC controller (jacket temperature setpoint) is sent to a secondary GMC controller. The output of the secondary GMC controller is the control valve position (0-100%) whose value is read into a register (F2.44), then directed to the control valves CV1-4/CV1-7 through sequence Seq 1.118 and an analog output card (channel 1.3.1).

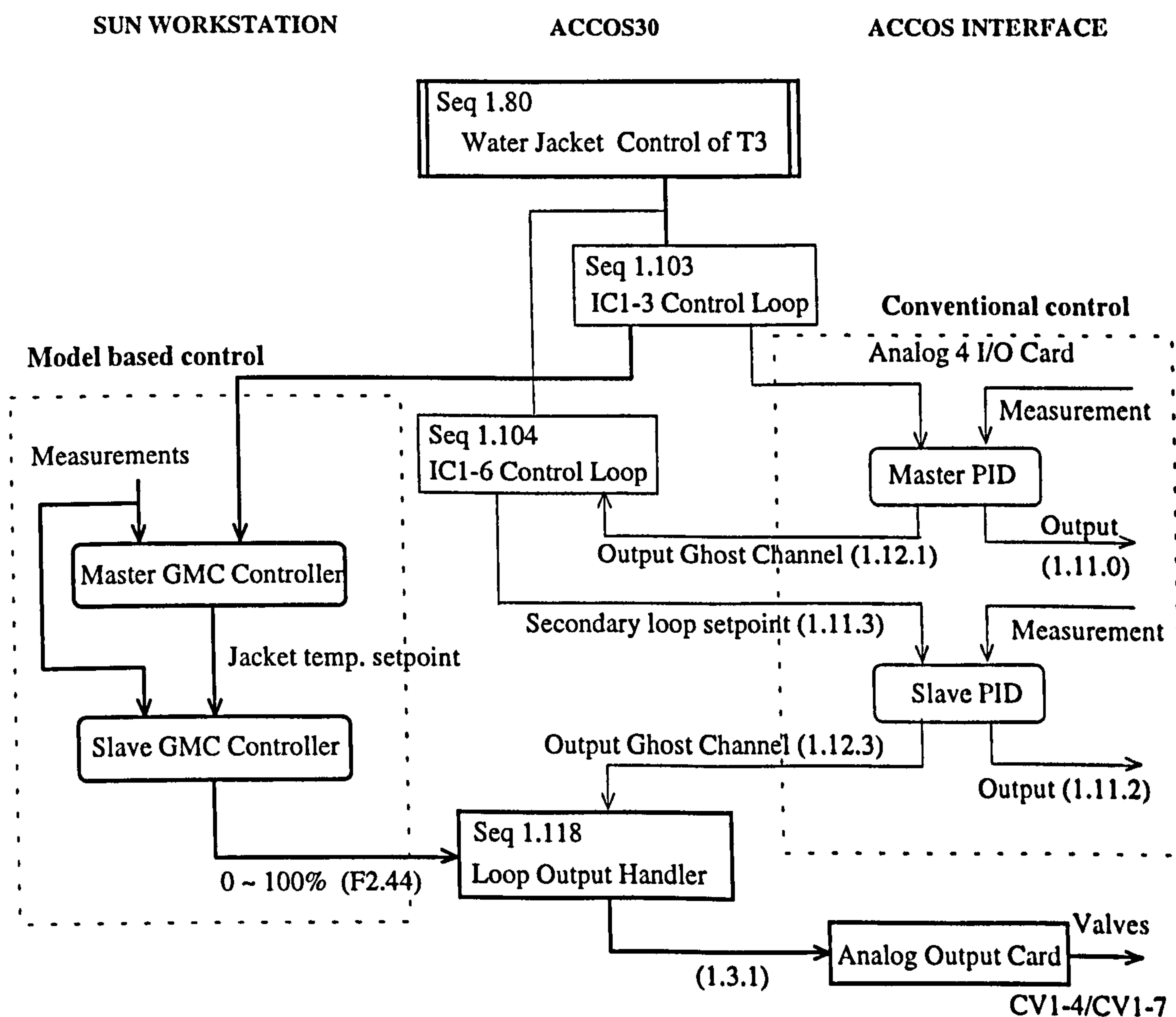


Figure A.4 Implementation of cascaded GMC controllers for the water jacket control

B. Implementation of Supervisory Control

The on-line scheduler (SUPERBATCH) is connected either to the simulator (gPROMS) or to the control system (ACCOS) using the Control Link software. To connect to the simulator, SUPERBATCH and the simulator open network connections to a Control Link server (configured for data storage) running on a Sun Workstation. Data are then passed between the two applications upon a request (*e.g.* using GET/SEND command) to the server. To connect to the plant, SUPERBATCH opens a connection to the Control Link server which is itself connected to ACCOS 300 and the plant. The connection/configuration to the simulator or the ACCOS control system is done through two files (ControlLink.dat and CL_tag_map.dat) as described in Chapter 3.

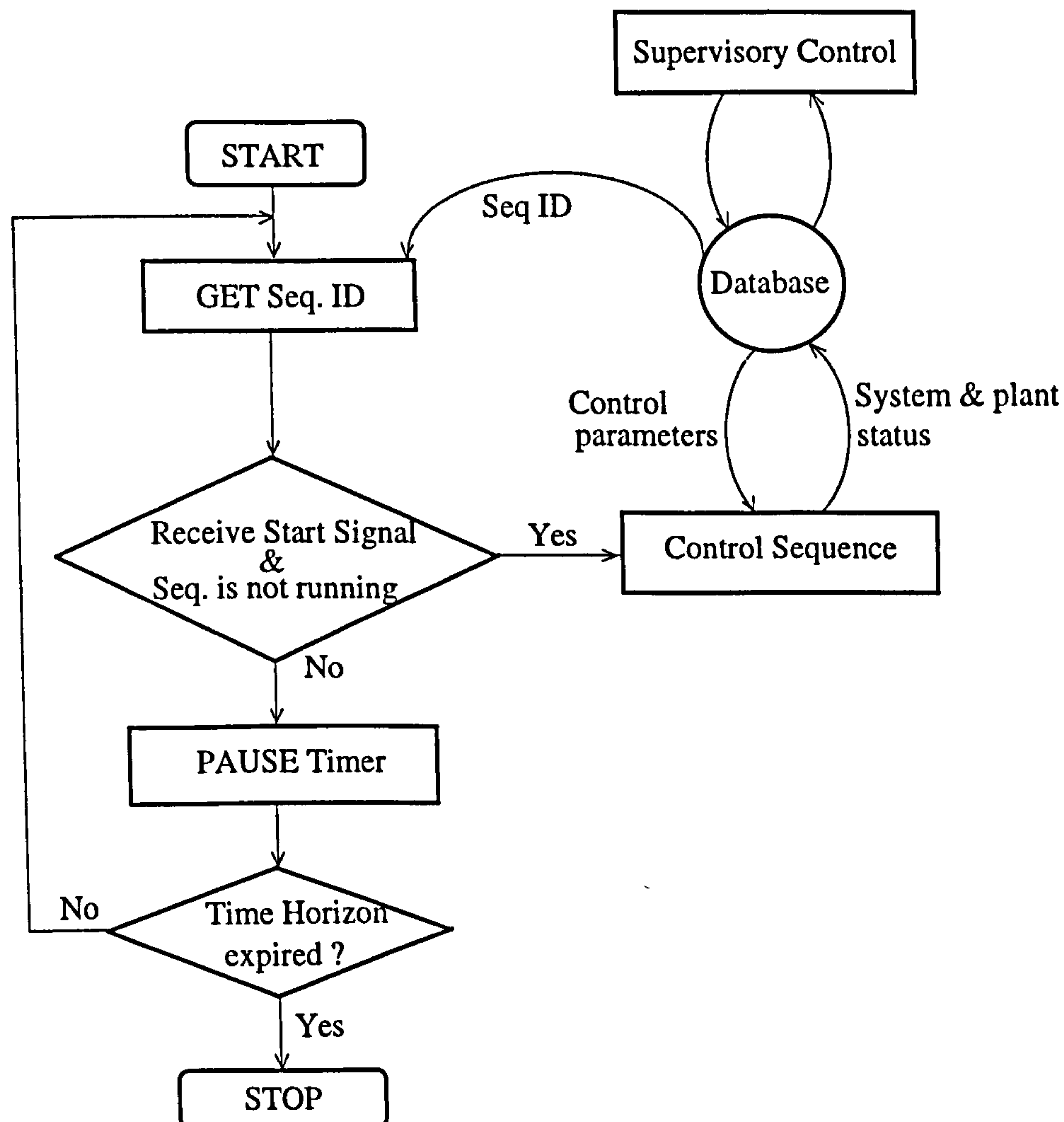


Figure B.1 Synchronisation between control phases and sequences

Synchronisation between the SUPERBATCH control phases and control tasks/sequences is achieved through a *Master task* in the simulator or an *Operator Entry Code* (OEC) handler sequence in the ACCOS control system (Figure B.1). This master task/OEC handler sequence monitors the information (start a task/sequence) passed to the Control Link server by SUPERBATCH, then passes the information to the appropriate task/sequence.

Detailed control parameters (*e.g.* temperature setpoints) are also passed to the appropriate sequences through the Control Link server. The plant data (inventory levels, actual batch sizes, etc.) and the status of the control sequences are fed back to the supervisory control level where on-line modification of current schedule is made.

