TEMPERATURE CONTROL OF MULTI-PRODUCT SEMI-BATCH POLYMERIZATION REACTORS

TEMPERATURE CONTROL OF MULTI-PRODUCT SEMI-BATCH POLYMERIZATION REACTORS

BY

TRACY LEE CLARKE-PRINGLE, B. Sc. (Eng.)

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

McMaster University

MASTER OF ENGINEERING (1995) McMASTER UNIVERSITY
(Chemical Engineering) Hamilton, Ontario

TITLE: Temperature Control of Multi-Product Semi-batch Polymerization Reactors

AUTHOR: Tracy Lee Clarke-Pringle, B.Sc. (Queen's University)

SUPERVISOR: Professor J.F. MacGregor

NUMBER OF PAGES: xii, 155

ABSTRACT

The work in this thesis focuses on the temperature control of a semi-batch polymerization reactor. The system is published by Chylla and Haase (1993) as an Industrial Challenge and is typical of reactors at S.C. Johnson Wax. The challenge is to find a single controller that can adequately regulate reactor temperature despite changing process conditions. The multi-product nature of the system makes it a particularly interesting problem. Several different controllers are implemented and evaluated in this thesis. The controllers are in part chosen to quantify the amount process information (large *or* small) required in a controller structure in order to achieve satisfactory control. Two of the most promising controllers are a PID with feedforward compensation and a Nonlinear Adaptive algorithm. It is found that in many cases, there may be little incentive to go to a complex model based controller as the simpler feedback algorithm provides adequate control. However, the nonlinear adaptive controller is more easily extended to multi-batch or multi-product situations because of its more general nature. The PID with feedforward compensation requires retuning for each new situation in order to maintain satisfactory control.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. John MacGregor, for his eternal patience and invaluable guidance over the course of this work. I would also like to thank my husband Matt for his unconditional support and understanding. Finally, I would like to thank my parents, for doing so many little things that make such a big difference.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

LIST OF TABLES

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 6

CHAPTER 7

CHAPTER 1
INTRODUCTION

1.1 THE INDUSTRIAL CHALLENGE PROBLEM

The operation of polymerization processes presents a unique and challenging control problem. Many polymerization reactors exhibit nonlinear dynamics, and sometimes operation at open loop unstable operating points is required for polymer quality. Other issues include high viscosity, lack of on-line sensors and highly exothermic reactions. All these conditions can create a situation where traditional algorithms perform poorly or even go unstable.

The motivation for the thesis is an industrial challenge problem published by Chylla and Haase (1993). The 1993 'industrial challenge' model is based on the authors' experiences with semi-batch polymerization reactors. The challenge is to formulate a temperature controller that can effectively deal with the following process characteristics:

- Semi-batch operation;
- Production of multiple products in a single reactor;
- Changing heat transfer ability (from batch to batch, and during a batch run);
- Auto-acceleration of the rate of reaction;
- Limited process knowledge;

The controller must maintain the reactor temperature within one degree of setpoint under all conditions. Monomer feed is started and stopped according to set product recipes. Currently, the control system is cascaded, as in Figure 1.1. The master controller controls temperature by manipulating the setpoint of the inlet jacket temperature. A slave controller adjusts valve positions to control the inlet jacket temperature to the setpoint calculated by the master controller.

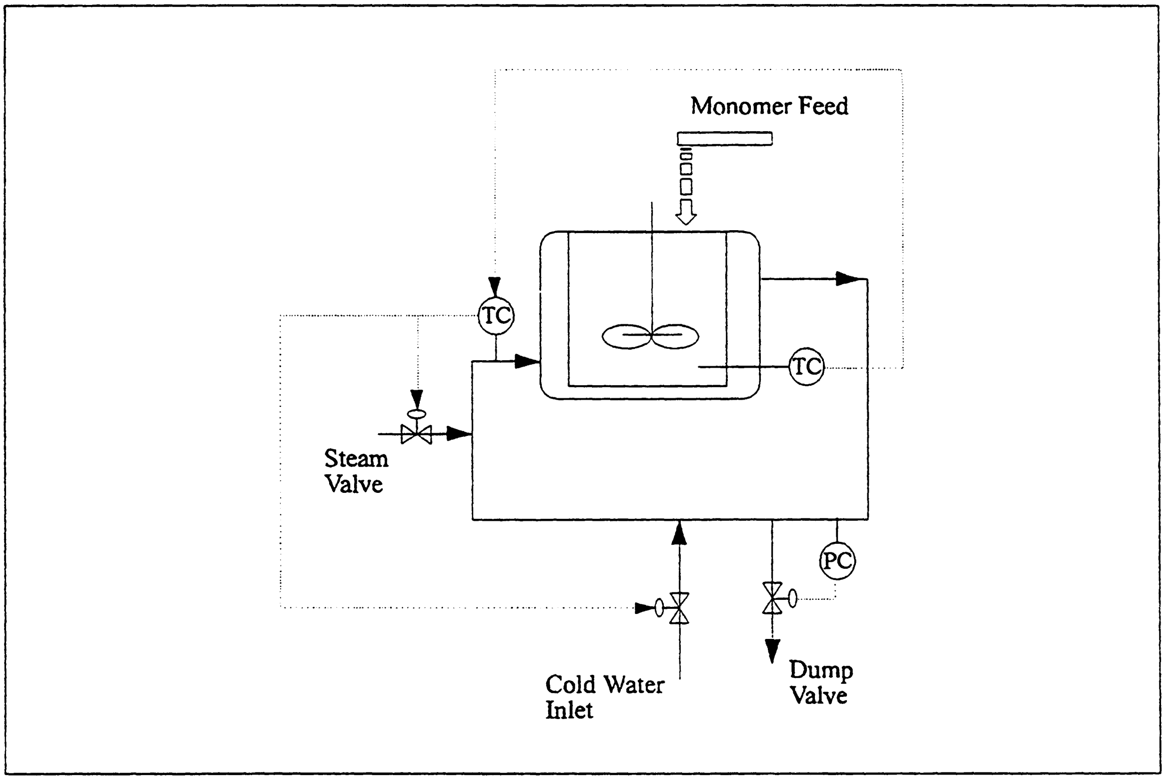


FIGURE 1.1: SCHEMATIC OF THE CHYLLA HAASE SEMI-BATCH REACTOR

It should be noted that the equipment of Figure 1.1 differs from the schematic published by Chylla and Haase (1993). A discussion is provided in Appendix A.

The model includes a random factor in the rate of reaction to simulate the effect of feed impurities from batch to batch. Summer and winter conditions are given for robustness tests. The authors encourage researchers to add noise to process measurements to more accurately represent the true environment.

The published model includes reactor and jacket energy balances and a material balance for the monomer. Empirical equations (based on actual plant data) describe the behavior of the heat transfer coefficient and viscosity. The authors also provide an equation to describe the dynamics of the coolant recirculation loop. For more specific details, the reader is referred to Chylla and Haase (1993) and updates. The reactor energy balance simulated in the thesis differs slightly from that of Chylla and Haase. An error in the equation derivation was discovered, therefore a correct reactor energy balance for a semi-batch reactor is derived and simulated instead. Please refer to Appendix A.

Table 1.1 summarizes key operating information:

Table 1.1: Operating Information for Products

	Product 1	Product 2
Setpoint	180 °F	176 °F

Recipe	1/ Charge reactor with water and pre-polymer as per the recipe;
	2/ Heat reactor to 179 °F then begin feed;
	3/ Feed monomer #1 at 1 lb/min for 70 minutes;
	4/ Hold reactor temperature to setpoint for 60 minutes;
		1/ Charge reactor as per the recipe;
		2/ Heat reactor to 175 °F then begin feed;
		3/ Feed monomer #2 at 0.8 lb/min for 60 minutes;
		4/ Hold reactor temperature to setpoint for 30 minutes;
		5/ Feed monomer #2 at 0.8 lb/min for 40 minutes;
		6/ Hold reactor temperature to setpoint for 45 minutes;

The recipe for each product is repeated for five batches. After the fifth batch, the reactor walls are cleaned to remove the polymer buildup (fouling). This buildup reduces the heat transfer ability of the system from batch to batch.

1.2 APPROACH

The goal of the thesis is to evaluate the problem of temperature control of semi-batch polymerization reactors. The multi-product, semi-batch nature of the Chylla Haase reactor makes it an interesting and challenging control problem. The effect of process knowledge on controller performance in these types of systems is highlighted in the thesis work. Several different controllers with varying degrees of process information are compared. Solutions evaluated include:

- well tuned PID
- adaptive PID
- PID with feedforward compensation
- nonlinear controller
- nonlinear adaptive controller

A portion of the thesis is dedicated to the PID controller and variations of this linear algorithm. It is felt that often this controller is overlooked in solutions because of its simplicity. Certainly, complex nonlinear model based controllers can provide more eloquent and impressive solutions. However, as the thesis work shows, surprisingly good control can be achieved by returning to the basics of PID control.

It should be noted that for the model based controllers, the focus is on solutions based on mechanistic (as opposed to empirical) process models. For a system with many products, empirical models must be identified for each product. Mechanistic models result in a more general control solution which is useful in multi-product systems. For this reason, controllers based on mechanistic models (as in nonlinear geometric control) are primarily considered for the Chylla Haase system.

All simulations were run in Matlab. The above controllers are implemented as the master controller in the cascaded loop. The inlet jacket temperature control can be satisfactorily accomplished using a PI controller as the slave. For all controllers, the slave is being executed every 0.6 seconds and the master every minute. The control interval for the slave was selected to reduce the impact this loop has on the performance of the master controller. The execution time for the master is selected as a reasonable value, based on examples in literature. It is expected that the control interval affects the performance of the algorithms. However, the thesis focuses primarily on comparing the controllers, therefore consistency between algorithms is far more important.

1.3 OVERVIEW OF THE THESIS

In this section, a brief overview of the thesis is provided. In chapters 3 through 6, each of the controllers is developed and tested on a Product One, batch one, summer simulation ('ideal conditions'). A comparison of the most promising controllers for nonideal situations is presented in Chapter 7. The details of each chapter are outlined below.

Before simulating the Chylla Haase system, the literature on the temperature control of polymerization reactors was thoroughly surveyed. Several of the most interesting papers are discussed in Chapter 2.

In Chapter 3, the time varying behavior of the Chylla Haase semi-batch reactor is quantified through an approximate linear analysis. Then, a PID and an adaptive PID controller are presented and the resulting control compared. The performance of the adaptive PID is comparable to the constant parameter PID when the former has full process knowledge. The adaptive PID performs considerably worse if limited process knowledge is available.

In Chapter 4, a second modification to the PID controller is presented. Feedforward compensation for the heat released and the feed stoppage is considered. With limited process knowledge, the PID with feedforward compensation outperforms the well tuned PID controller.

Chapter 5 introduces geometric nonlinear control. Nonlinear control theory is reviewed and a controller is derived for the semi-batch reactor. With exact knowledge of all process parameters, the nonlinear controller outperforms the PID algorithm. However, it is shown that the performance of the nonlinear controller is very sensitive to errors in parameter values. Using average values of some parameters in the algorithm, the nonlinear controller performs worse than a PID controller. This chapter highlights the need for an on-line parameter estimation scheme.

In Chapter 6, Kalman filtering is introduced to track model parameters. The set of parameters to be updated is chosen in part based on the results of Chapter 5. Problems associated with updating the parameters are outlined in the first half of the chapter. In the second half, the estimation scheme is combined with the nonlinear controller. The performance of the nonlinear adaptive controller is comparable to that of the nonlinear controller with all parameters known. A brief study shows that the nonlinear adaptive algorithm is robust to errors in the required initial parameter guesses.

Chapter 7 summarizes the work by comparing three controllers under non-ideal conditions. Noise is added to all simulations. The proportional integral derivative controller, the PID with feedforward compensation and the nonlinear adaptive controller are compared. The three controllers show a nice progression as process knowledge in the algorithms is increased. As expected, the nonlinear adaptive algorithm outperforms the PID class controllers. For the ideal simulation (summer, batch one), the improvement is not a substantial as one might expect. The nonlinear adaptive controller is more effectively extended to multiple batches and multiple products. This is a direct result of the mechanistic model on which the controller is based.

Chapter 8 summarizes the results of the thesis.

CHAPTER 2

TEMPERATURE CONTROL OF POLYMERIZATION REACTORS

2.1 INTRODUCTION

Before proceeding with the problem of controlling temperature in the Chylla Haase polymerization reactor, it is important to be familiar with work already done. For this reason, a comprehensive literature survey of the control of polymerization reactors was undertaken. Papers are organized according to algorithm and four major areas are covered: traditional approaches, model predictive control, adaptive control and nonlinear control. Only selected papers are discussed, however summary information on all papers is available in Table 2.1 at the end of Chapter 2.

A general point of interest is the tuning of 'comparison controllers'. Often, researchers tune the PID controllers from step/PRBS tests done on the reactor full of water/solvent. This can be very misleading and could result in poor models and tuning. Fine tuning (trial and error simulations or experiments) is required to obtain a well tuned PID. Comparing a new control strategy to a poorly tuned PID does not provide an accurate comparison.

2.2 Traditional Linear Control

2.2.1 Introduction

Before discussing advanced control methods applied to polymerization reactors, it is worthwhile evaluating papers that deal with traditional process control. Simple extensions of traditional control, such as feedforward control, are also considered here. This section is divided into two categories: feedback algorithms, and combined feedforward/feedback algorithms.

2.2.2 Feedback Algorithms

An interesting application is presented by Davidson (1987). Davidson (1987) introduces an 'intelligent' temperature controller for jacketed reactors. He points out the need for a simple but functional controller and combines an algorithm with logic. The general idea is that speed of response is improved by relying on Proportional action; integral action is implemented only when the temperature is close to its setpoint. The controller is coded as a series of 'if *this* then do *that*' statements, which is not particularly useful for general applications. While the controller is a good example of unique thinking, more simulations are required to be convincing.

Lie and Balchen (1992) evaluate four different structures for controlling a polymerization reactor. The paper attempts to highlight issues of multivariable control in an interactive, nonlinear system. The example system is a polypropene CSTR with the following controlled variables: level in reactor, level in the reflux accumulator and reactor pressure (conversion). The authors look at the existing multiloop PI structure, the current PI structure with better tuning, multiloop PI with different loop pairing, multivariable LQ (with input weighting) and a nonlinear decoupling controller. The controllers are compared in simulations for disturbance rejection, pressure setpoint tracking, and changing process conditions ('model errors'). Lie and Balchen (1992) conclude that the multivariable LQ is most promising overall. Unfortunately, no insight into the findings is provided. Given that the system is continuous, with no large operating changes, the nonlinearities of the system may not be significantly excited. The interactions are strong, hence a linear, multivariable controller is probably all that is needed. Such a discussion would have provided an excellent wrap up for an otherwise interesting paper.

Henderson and Cornejo (1989/1987) explore the effect of viscosity on temperature control. Three different control schemes are evaluated : heat removal by internal coils/jacket, by evaporation, and by external heat exhange. The mathematical treatment is not overly theoretical and the paper provides excellent insight into the behavoir of polymerization reactors. Summary information is available in Table 2.1.

Dougherty et al.(1988) are concerned with controlling temperature in an industrial semi-batch polymerization reactor. Prior to the work, the reactor was under manual control. By *correctly* identifying a stochastic model (transfer function and disturbance model) of the process, a well tuned PID results. Temperature control is significantly improved on the industrial reactor.

Chien and Penlidis (1994b) perform a very interesting study. It is the only study found that conducts simulations and experiments then compares the results. The control of conversion with initiator flow is considered in a continuous methyl methacrylate solution polymerization reactor. Control is tested in presence of unmeasured impurity disturbances. The authors compare a Smith Predictor, the Dahlin controller, a PID, a Minimum Variance, a Constrained Minimum Variance, and an LQ 1 step controller. Simulation and experimental runs are conducted for all controllers. While the simulations differed slightly from the experiments, the conclusions drawn from both were the same. In both experimental and simulated conditions, the stochastic controllers outperformed the PID-type controllers. The reasons for this were not discussed. However, the stochastic controllers do have more process information incorporated in them and this may be the source of the improved control.

2.2.3 FEEDFORWARD/FEEDBACK ALGORITHMS

Juba and Hamer (1986) published a very insightful paper on the control of batch and semi-batch reactors. They highlight the issues of stability of exothermic reactors; feedforward compensation for the heat released due to reaction; tracking optimal temperature policies (known and unknown kinetics) and property control. It is a comprehensive article that shows excellent understanding.

Congalidis et al.(1989) present a feedforward/feedback algorithm for a multivariable reactor with recycle. The control of rate, compostion, molecular weight and temperature is considered assuming accurate online measurements of these outputs. This is not necessarily valid for composition and molecular weight, and the results of the paper are misleading. Feedforward action is used to compensate for flow and composition disturbances in the recycle. Summary information is available in Table 2.1.

2.3 MODEL PREDICTIVE CONTROL

2.3.1 INTRODUCTION

In this section, model predictive applications to polymerization reactors will be discussed. Papers are grouped into two categories, polymer applications and control applications and is based on the overall perspective of the publication.

2.3.2 POLYMER APPLICATIONS

Inglis et al.(1991) examine the application of generalized predictive control in two parts. The control of conversion in a simulated MMA CSTR and temperature control of an experimental batch MMA are considered. The strength of the paper is that the authors are concerned with controlling the polymerization reactor and not with demonstrating an algorithm.

Dittmar et al.(1991) evaluate two predictive control algorithms: an adaptive Generalized Predictive Controller, and a new nonadaptive predictive controller developped by the authors. The motivation for the problem is the stabilization of an industrial reactor. Temperature control of solution styrene/acrylnitrile copolymerization is considered by manipulating the cooling oil flow. The controllers are tested on a realistic simulation model for disturbance rejection and setpoint tracking capabilities. The newly derived predictive controller depends on historical data instead of a process model for predicting the output and the authors show that it outperforms the adaptive-predictive controller and a PID.

Gattu and Zafirio (1991) propose a Nonlinear QDMC solution for the 'industrial challenge problem' that is the basis for this thesis work. They did not compare the controller to a PID; therefore it is impossible to tell if there is an improvement in control. The multi-product issue is not addressed in the solution. The authors are more concerned with demonstrating the NLQDMC algorithm that solving the actual control problem.

2.3.3 Control Applications

Hidalgo and Brosilow (1990) present a framework for a nonlinear model predictive controller. Issues addressed in paper are considerations for unstable processes and options for handling model mismatch and unmeasured disturbances.

Prasad et al.(1990) demonstrate the capabilities of a multivariable model based controller on a solution methyl methacrylate CSTR. The control of reactor temperature and conversion with jacket temperature and monomer concentration is presented. The apparent issues are nonlinearity and interaction. In general, the paper is quite weak. The authors obtain linear models for the process by doing tiny step changes. Therefore, they have very good linear approximations of the process around the operating point. In a real system, such small step tests are not possible. Simulations show that the controller has difficulties handling a time varying heat transfer coefficient because the linear models on which the controller is based are inaccurate over a wide range.

2.4 Adaptive Control

2.4.1 Introduction

Adaptive control has clearly been a productive area of research since the last major review on polymer reactor control by MacGregor et al. (1984). Most of the work was accomplished in the mid- to late- 1980s, however new papers on adaptive control still frequently appear. Adaptive control applications have been divided into the following two subsections: pole-placement adaptive control and robust adaptive control.

The adaptive control field is very algorithmic and the main criticism of this area is that the controllers are often applied with little insight. Adaptive control can provide excellent control when applied *correctly* to time varying systems. The papers summarized in this section represent some of the stronger publications.

2.4.2 Pole Placement Adaptive Control

Tzounas and Shah (1989) published an excellent paper to illustrate some important issues in adaptive control. An adaptive pole placement controller manipulates the desired reactor temperature in order to control conversion. An experimental batch MMA reactor is considered. The adaptive pole placement controller is compared to a generalized minimum variance controller (adaptive 1-step LQ) and a fixed gain PID. The issues highlighted in the paper are: the need for good initial parameter estimates (options are provided), global stability of the adaptive controller (mathematical treatment presented) and the need for on-line measurements. With reference to the last point, the adaptive algorithms only outperformed the PID when on-line measurements were available every minute (currently, sampling time for polymer quality is upward of 15 minutes). The authors conclude that the advantages of adaptive algorithms can only be realized with on-line sensors that have short sampling intervals. This is particular point is important for polymer quality control, where measurements are very infrequent.

Ham and Rhee (1994) present an interesting application of an adaptive pole placement controller. The authors apply the controller to a LDPE autoclave reactor. Significant effort went into modeling the system for simulations. Since this is a conference paper, it is expected that a publication with more details will follow.

2.4.3 Robust Adaptive Control

Mendoza-Bustos et al.(1990) evaluate a Stable Robust Adaptive Controller (SRAC). The system in question is a simulated MMA CSTR where the flow of initiator is manipulated to control conversion. The SRAC is an adaptive minimum variance controller with the robustness feature added to the estimator. The estimation algorithm uses a normalizing factor such that the estimator sees the unmodeled dynamics as bounded disturbances (that is, the estimator is not mislead by the 'disturbances'). The paper indicates that the controller shows promise. The paper itself is very insightful for polymerization applications.

Defaye et al.(1993) present an adaptive-predictive algorithm for improving temperature control in semi-batch reactors. In semi-batch systems, feed starting/stopping often introduces undesirable temperature disturbances. The authors treat an experimental semi-batch, copolymerization reactor (Vinyl acetate/2 ethyl-hexyl acrylate). The algorithm distinguishes itself from others by using filtered data in the estimation (adaptation) part to avoid misleading the estimator during feed starts/stops. The data filtering before estimation is credited with the improvement seen in the control.

Despite the many complicated algorithms tried and tested in literature, there are few reported applications in industrial reactors. An excellent example of adaptive control in an industrial setting was published by Whatley and Pott (1984). In the paper, an adaptive gain PI controller is applied to an industrial polymerization reactor. The simple adaptive algorithm is used to control the reactor temperature and maintain stability in the system. The controller results in higher profits because of improved operability.

There are many other adaptive algorithm applications and the remaining papers in this area are summarized in Table 2.1 at the end of chapter 2.

2.5 NONLINEAR CONTROL

2.5.1 INTRODUCTION

Papers in this area are classified into to sections according to their objectives. In the first section, Nonlinear Control Concepts, papers which discuss more global concerns of geometric nonlinear control are presented. In the second section, Nonlinear Control Algorithms, papers that primarily demonstrate specific nonlinear controllers are discussed. Several good review papers are available to understand the nonlinear control field. Bequette (1990) gives a broad overview of nonlinear control; this is a helpful paper for seeing what is available. McLellan et al.(1990) have an excellent paper for understanding the similarities between many nonlinear geometric control algorithms.

Most successful applications of nonlinear control have very complex process models in their controllers. While this results in impressive control, it presents a major barrier to implementation in industry. Clearly, this difference will need to be overcome before the nonlinear control algorithm can make the leap from academia to industry.

2.5.2 NONLINEAR CONTROL CONCEPTS

A large source of confusion in the field of geometric nonlinear control is that many very similar algorithms are called by different names. Despite the different derivations, most nonlinear controllers are based on differential geometric control theory, and differ only in how the researcher chooses to approach the problem.

Balchen et al.(1988) investigate a multivariable nonlinear controller in decoupling framework. The key issues are the effect of process/model mismatch on closed loop stability (analytical treatment); dealing with a plant in which there are more controlled variables than manipulated variables; and plants in which the manipulated variable does not appear explicitly in the controller equations (requires implicit solution). The paper concludes with an application of the nonlinear decoupling controller to a simple process simulation.

Alvarez and coworkers have a series of papers treating nonlinear control of polymerization reactors (Alvarez, Suarez and Sanchez (1990), Alvarez, Hernandez, Suarez (1988), Alvarez, Alvarez, Gonzalez (1989)). While the papers do not provide a lot of insight into polymerization concepts, they do treat the problem more mathematically than most.

2.5.3 NONLINEAR CONTROL ALGORITHMS

Cott and Macchietto (1989) demonstrate the capabilities of a nonlinear model based controller. They consider the control of reactor temperature by manipulating the jacket temperature setpoint using Lee and Sullivan's Generic Model Control (GMC). The generic batch exothermic reactor model in the controller allows the algorithm to be quite general; an on-line energy balance provides estimate of Q_r for the controller. Results show that the nonlinear GMC controller outperforms the dual mode controller for the cases chosen. However, the PID in the dual mode controller is poorly tuned and the extent of nonlinearity of the system is unclear. It appears the estimator, not the nonlinear controller, is providing the observed improvement in control.

Soroush and Kravaris (1992) present an experimental application of the Globally Linearizing Control algorithm. The system in question is a batch MMA reactor. The control of temperature and conversion is considered. The authors perform a very thorough robustness study; however, all errors are on the 'safe' side. As a result, the question of closed loop stability in the face of errors is unanswered. As well, the controller uses very complex process models; it is unlikely that these models would be available in industry. However, despite the weakness, this is a good article to understand the GLC algorithm, and the experimental nature of the study is very interesting. Kravaris and coworkers have several other papers published on this topic (Daoutidis, Soroush and Kravaris (1990), Soroush and Kravaris (1993)).

A very unique industrial application of nonlinear control is published by Singstad et al.(1992). A multivariable, nonlinear controller for an industrial LDPE plant is briefly presented. When the controller was implemented, improved temperature and conversion control resulted.

2.6 SUMMARY TABLE

Key details on all papers are summarized in Table 2.1.

Table 2.1: Summary Information on Temperature Control of Polymerization Reactors

Reference	Polymer	Simulation/ Experiment	Reactor Type	Controlled Variables	Manipulated Variables
Adebekun and Schork (1989)	MMA	sim	continuous	[M], [I], T, [solv]	$[M]_{feed}$, Tj, $[I]_{feed}$
Alvarez, Alvarez, Gonzalez (1989)	-	sim	continuous	T, [M]	Tj
Alvarez, Hernandez, Suarez (1988)	MMA	sim	continuous	T,[M], [I]	$[I]_{feed}$, T_j
Alvarez, Suarez, Sanchez (1990)	MMA	sim	continuous	T,x	initiator flow, T_j
Balchen et al.(1988)	-	sim	continuous	T	Q_j

Chien and Penlidis (1994a)	MMA	exp	continuous	X	initiator flow
Chien and Penlidis (1994b)	MMA	sim&exp	continuous	X	initiator flow
Cluett et al.(1985)	PVC	sim	batch	T or rate	steam flow
Congalidis et al.(1989)	VAC, MMA	sim	continuous	prodn. rate, MW, T, comp.	monomer, solvent, initiator & cta flows, Tj
Cott and Macchietto (1989)	-	sim	continuous	T	Tj
Daoutidis et al.(1990)	MMA	sim	continuous	T, MW	Fj, initiator flow
Davidson (1987)	-	sim	batch	T	Tj
Defaye et al.(1993)	VA/2-ethyl-hexyl acrylate	exp	semi-batch	T	heating (power)
Dittmar et al.(1991)	stryene/ acrylnitril	sim	continuous	T	Fj
Dougherty et al.(1988)	unknown	exp	semi-batch	T	monomer flow
Farber and Ydstie (1986)	styrene	sim	continuous	T	heat transfer coefficient
Gattu and Zafirio (1991)	unknown	sim	semi-batch	T	heating valve
Ham and Rhee (1994)	LDPE	sim	continuous	T	initiator flow
Henderson and Cornejo (1989)	styrene	sim	continuous	T	-
Hindalgo and Brosilow (1990)	styrene	sim	continuous	T	F_j, monomer flow
Houston and Schork (1987)	MMA	sim	semi-batch	T,X or X,MW	initiator flow, T_j
Inglis et al.(1991)	MMA	sim&exp	both	X or T	initiator flow or T_j
Jutan and Uppal (1984)	-	sim	batch	T	heating
Kwalik and Schork (1985)	MMA	sim	continuous	[M],T	$[I]_{feed}$, T_j
Lee et al.(1994)	ABS	industrial	batch	T	heating
Mendoza-Bustos et al. (1990)	MMA	sim	continuous	X	initiator flow
Merkle and Lee (1989)	-	exp	batch	T	steam flow
Papadoulis et al.(1987)	-	sim	continuous	[reactnt]	coolant flow
Peterson et al.(1992)	MMA	sim	semi-batch	T, MW	Tj, initiator flow
Prasad et al.(1990)	MMA	sim	continuous	T, [M]	Tj, $[I]_{feed}$
Singstad et al.(1992)	LDPE	ind	continuous	T	
Soroush and Kravaris (1992)	MMA	exp	batch	T	heating
Soroush and Kravaris (1993)	MMA	exp	continuous	T,X	heating, initiator flow

Tzouanaz and Shah (1989)	MMA	sim&exp	batch	X or MW	T_{sp}
Wang and Lin (1991)	PVC	sim	batch	T	steam flow
Wang et al.(1994)		sim	continuous	X, T	heating valve opening, $[I]_{feed}$
Whatley and Pott (1984)	unknown	industrial	continuous	T and $T_{oil,sp}$	$T_{oil,sp}$ and $F_{oil,in}$

CHAPTER 3

LINEAR CONTROL

3.1 INTRODUCTION

The purpose of this chapter is to explore the issues surrounding linear control. First, the time varying, nonlinear nature of the reactor is quantified in order to provide some insight into controller performance. A Proportional Integral Derivative Controller is implemented and will serve as a basis of comparison for future solutions. An adaptive PID (integral time constant and controller gain adapted) is also implemented. The performance of the adaptive PID is comparable to the constant parameter PID controller.

3.2 TIME VARYING NATURE OF THE REACTOR

3.2.1 DERIVATION OF TIME CONSTANT AND PROCESS GAIN

Before implementing and comparing controllers, it is important to understand the time varying and nonlinear behavior of the system. An approximate linear analysis is helpful in this aspect. The procedure in this section may be visualized as freezing the system at an operating point X_{op}, and performing a linear analysis of the system at that point. Repeating this for several operating points gives a snap shot view of the reactor at selected points in the batch.

An energy balance around the cooling jacket and the reactor provides the equations for deriving the linearized time constant and process gain. The energy balance for the reactor may be written as:

$$\sum(m_i C_{pi})\frac{dT}{dt} = \dot{m}_m C_{pm}(T_{feed} - T) - UA(T - T_{j,out}) + Q_r \tag{3.1}$$

Notation used here and throughout the thesis may be found in Appendix B, and is consistent with that used in Chylla and Haase (1993).

In the absence of any knowledge of reactor dynamics, this equation provides a solid starting point. Equation (3.1) assumes the jacket is well mixed (i.e. jacket temperature may be represented as T_{jout}). As well, it is assumed for equation (3.1) that no information about the heat loss to the surroundings is available. If kinetic information is available, the term for the heat release, Q_r, may be expanded:

$$\begin{aligned} Q_r &= R_p(-\Delta H_r) \\ &= k_0 \mu^{0.4} n_m e^{-\alpha/T}(-\Delta H_r) \end{aligned} \tag{3.2}$$

It should be noted that the viscosity term in the rate of reaction (R_p) is included in Chylla and Haase (1993) to model the auto-acceleration of the reaction with increased viscosity. For the initial analysis, full knowledge of R_p is assumed in order to illustrate the variations in the semi-batch reactor time constant and process gain. In section 3.2.2, the implication of *not* knowing the kinetics is discussed.

Equation (3.1) requires linearization for the linear analysis. An obvious nonlinearity is the Arrhenius dependence of reaction rate. The heat transfer coefficient and reactor viscosity are also typically treated as nonlinear functions of temperature. However, consider the following logic with respect to the heat transfer: as the reactor is being heated up, the temperature changes over 80 °F. The corresponding change in the heat transfer coefficient is about 4 Btu/(ft^2 hr °F). While the monomer is being fed, reactor temperature is held constant at its setpoint. During the feed period, the heat transfer coefficient drops over 90 Btu/(ft^2 hr °F). This parameter is clearly a much stronger function of some other variable (fraction solids for example). Therefore, it may be assumed that U is a time varying parameter, independent of temperature. The same logic also applies to viscosity. As a result, only the Arrhenius expression requires linearization with respect to temperature.

Taylor series expansion is used to linearize equation (3.2) about some operating state, denoted by T$_{op}$. Expressed in deviation variables, the linearized form of equation (3.1) is:

$$\sum_{i=1}^{3}(m_i C_{pi})\frac{d\tilde{T}}{dt} = \dot{m}_m C_{pm}(\tilde{T}_{feed} - \tilde{T}) - UA(\tilde{T} - \tilde{T}_{j,out}) + k_0 \mu^{0.4} n_m(-\Delta H_p)(e^{-\alpha/Top}\frac{\alpha}{T_{op}^{\ 2}}\tilde{T}) \tag{3.3}$$

where

$$\tilde{T} = T - T_{op}$$

Rearranging:

$$\sum_{i=1}^{3}(m_i C_{pi})\frac{d\tilde{T}}{dt} + (\dot{m}_m C_{pm} + UA - k_0 \mu^{0.4} n_m(-\Delta H_p)(e^{-\alpha/Top}\frac{\alpha}{T_{op}^{\ 2}}))\tilde{T} = \dot{m}_m C_{pm}\tilde{T}_{feed} + UA\tilde{T}_{j,out} \tag{3.4}$$

Let:

$$\tau_{reactor} = \frac{\sum_{i=1}^{3}(m_i C_{pi})}{(\dot{m}_m C_{pm} + UA - k_0 \mu^{0.4} n_m(-\Delta H_p)(e^{-\alpha/Top}\frac{\alpha}{T_{op}^{\ 2}}))} \tag{3.5}$$

$$K_{r1} = \frac{\dot{m}_m C_{pm}}{(\dot{m}_m C_{pm} + UA - k_0 \mu^{0.4} n_m (-\Delta H_p)(e^{-\alpha/Top} \frac{\alpha}{T_{op}^{\ 2}}))} \tag{3.6}$$

$$K_{r2} = \frac{UA}{(\dot{m}_m C_{pm} + UA - k_0 \mu^{0.4} n_m (-\Delta H_p)(e^{-\alpha/Top} \frac{\alpha}{T_{op}^{\ 2}}))} \tag{3.7}$$

Therefore, the linearized energy balance for the reactor may be represented in the Laplace domain as:

$$\widetilde{T}(s) = \frac{K_{r1}}{\tau_{reactor} s + 1} \widetilde{T}_{feed}(s) + \frac{K_{r2}}{\tau_{reactor} s + 1} \widetilde{T}_{j,out}(s) \tag{3.8}$$

In the semi-batch system, the manipulated variable is the setpoint for the inlet jacket temperature. An energy balance for the jacket is required to obtain T_{jout} (and hence T) in terms of T_{jin}. Assuming a well mixed cooling jacket:

$$m_c C_{pc} \frac{dT_{j,out}}{dt} = \dot{m}_c C_{pc}(T_{j,in} - T_{j,out}) + UA(T - T_{j,out}) \tag{3.9}$$

Equation (3.9) is a first order linear differential equation. After applying a Laplace transform, equation (3.9) is expressed as:

$$\widetilde{T}_{j,out}(s) = \frac{K_{j1}}{\tau_j s + 1} \widetilde{T}_{j,in}(s) + \frac{K_{j2}}{\tau_j s + 1} \widetilde{T}(s) \tag{3.10}$$

where

$$K_{j1} = \frac{\dot{m}_c C_{pc}}{\dot{m}_c C_{pc} + UA} \tag{3.11}$$

$$K_{j2} = \frac{UA}{\dot{m}_c C_{pc} + UA} \tag{3.12}$$

$$\tau_j = \frac{m_c C_{pc}}{\dot{m}_c C_{pc} + UA} \tag{3.13}$$

Replacing T_{jout} in equation (3.8) with equation (3.10):

$$\widetilde{T} = \frac{K_{r1}}{\tau_{reactor} s + 1} \widetilde{T}_{feed} + \frac{K_{r2}}{\tau_{reactor} s + 1} \left(\frac{K_{j1}}{\tau_j s + 1} \widetilde{T}_{j,in} + \frac{K_{j2}}{\tau_j s + 1} \widetilde{T} \right) \tag{3.14}$$

Simple algebraic manipulation results in equation (3.15):

$$\begin{aligned} \widetilde{T} = {} & \frac{K_{r1}(\tau_j s + 1)}{(\tau_{reactor} s + 1)(\tau_j s + 1) - K_{r2} K_{j2}} \widetilde{T}_{feed} \\ & + \frac{K_{r2} K_{j1}}{(\tau_{reactor} s + 1)(\tau_j s + 1) - K_{r2} K_{j2}} \widetilde{T}_{j,in} \end{aligned} \tag{3.15}$$

Rearranging:

$$\left(\frac{\tau_{reactor} \tau_j}{1 - K_{r2} K_{j2}} s^2 + \frac{(\tau_{reactor} + \tau_j)}{1 - K_{r2} K_{j2}} s + 1 \right) \widetilde{T} = K_{r1} (\tau_j s + 1) \widetilde{T}_{feed} + \frac{K_{r2} K_{j1}}{1 - K_{r2} K_{j2}} \widetilde{T}_{j,in} \tag{3.16}$$

Equation (3.16) shows that at any operating point, a second order relationship exists between the inlet jacket temperature and the reactor temperature. To obtain a more convenient expression, a further simplification may be made. Compared to the time constant between T_{jout} and T ($\tau_{reactor} > 7$ minutes), the dynamics between T_{jin} and T_{jout} are very fast ($\tau_j < 0.3$ minutes). Therefore, neglecting the jacket dynamics introduces only a small error into the approximate linear analysis. Equation (3.10) becomes:

$$\widetilde{T}_{j,out} = K_{j1}\widetilde{T}_{j,in} + K_{j2}\widetilde{T} \tag{3.17}$$

Substituting equation (3.17) into equation (3.8) and rearranging gives:

$$\widetilde{T} = \frac{K_{r2}K_{j1}/(1-K_{r2}K_{j2})}{\dfrac{\tau_{reactor}}{1-K_{r2}K_{j2}}s+1}\widetilde{T}_{j,in} + \frac{K_{r1}(\tau_j s+1)}{\dfrac{\tau_{reactor}}{1-K_{r2}K_{j2}}s+1}\widetilde{T}_{feed} \tag{3.18}$$

Equation (3.18) can be equivalently derived by setting τ_j to zero in equation (3.16). Expressions for the instantaneous, linearized system time constant and gain of the process are:

$$K_{process} = \frac{K_{r2}K_{j1}}{1-K_{r2}K_{j2}} \tag{3.19}$$

$$\tau_{process} = \frac{\tau_{reactor}}{1-K_{r2}K_{j2}} \tag{3.20}$$

It should be noted that the material balance for the monomer is not included in the analysis. By neglecting the material balance, the system order is reduced. This introduces some error into the linear analysis presented above; not all of the process dynamics are captured with a first order model. The mismatch between the 'true' second order process and the approximate first order process is most substantial immediately after the feed starts. At this time, the poles of the characteristic equation are equal in magnitude and contain small imaginary parts (indicating underdamped behavior is possible). After this initial transient, the pole associated with the reactor energy balance dominates and contains no imaginary part. Despite the errors introduced by approximating the higher order process with a lower order model, the analysis still provides some insight into the semi-batch reactor.

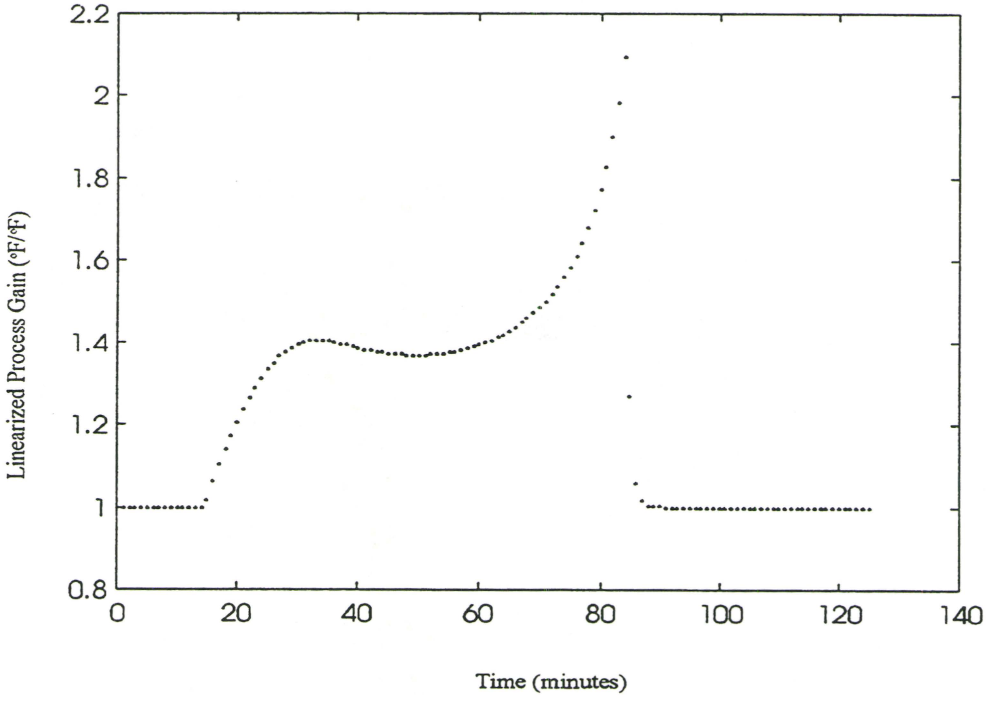


FIGURE 3.1: LINEARIZED PROCESS GAIN (PRODUCT ONE, BATCH ONE, SUMMER)

Equations (3.19) and (3.20) are functions of process variables (e.g. viscosity, heat transfer coefficient) and may be evaluated periodically during a batch run. Figures 3.1 and 3.2 show how the linearized gain and time constant typically vary for a Product One, batch one simulation. Exact knowledge of all the process variables (e.g. heat transfer, viscosity) is assumed.

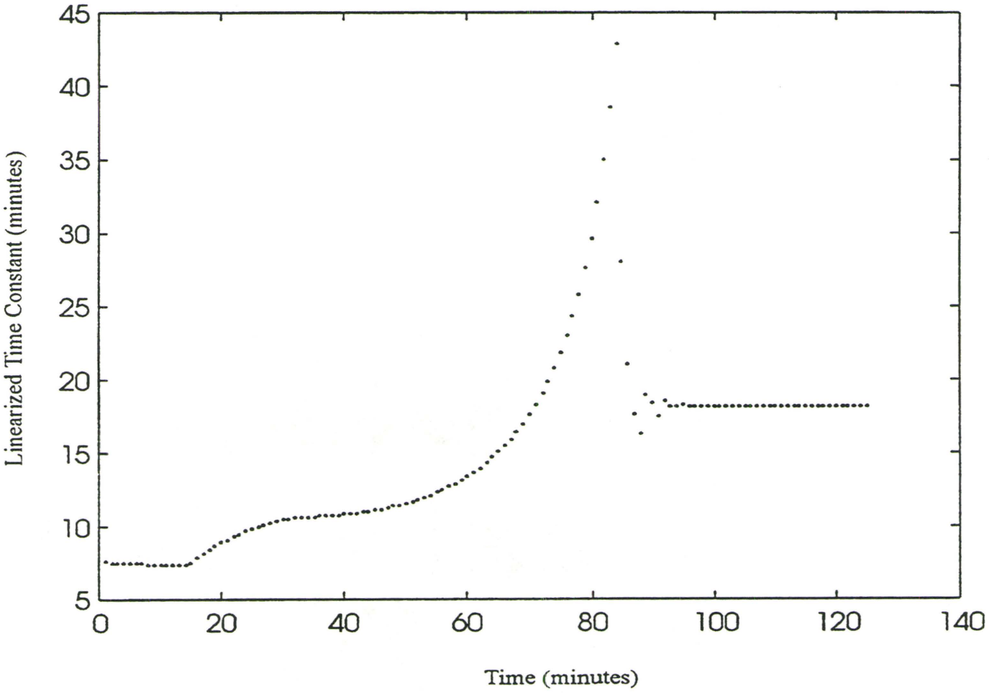


FIGURE 3.2: LINEARIZED TIME CONSTANT (PRODUCT ONE, BATCH ONE, SUMMER)

In this section, the linearized time constant and process gain for the semi-batch reactor have been derived. The expressions for the linearized parameters were evaluated periodically during a product one, batch one run and presented in figures.

3.2.2 ANALYSIS OF TIME CONSTANT AND PROCESS GAIN

Expressions for the instantaneous linearized time constant and process gain were developed in section 3.2.1. In this section, the individual contributions of select process parameters on the behavior of the linearized time constant and gain are evaluated. Only

Product One is considered here; Product Two has unique problems that will be discussed in section 3.2.3.

The linearized time constant and process gain from section 3.2.1 can be re-evaluated during a batch for three different situations:

CASE 1: Effect of the heat released during the reaction. In many situations, an expression for R_p is unavailable, and Q_r must be treated as a lumped, unknown disturbance acting on temperature. Therefore, equations (3.5) through (3.7) are re-evaluated without the expression $\{k_0\mu^{0.4}n_m(-\Delta H_p)e^{-\alpha/T_{op}}(\alpha/T_{op}^{2})\}$, and used in equations (3.19) and (3.20).

CASE 2: Effect of the viscosity term in the rate of reaction (auto-acceleration). In evaluating equations (3.19) and (3.20), assume a value of one for the viscosity multiplying factor ($\mu^{0.4}$). The true value of the variable changes from one and nine during a batch.

CASE 3: Effect of falling heat transfer coefficient. A constant value of U equal to its initial value is used in evaluating equations (3.19) and (3.20).

The three cases and the original plots from Figures 3.1 and 3.2 are shown in Figures 3.3 (process gain) and 3.4 (time constant). The figures demonstrate how each of the components contribute to the time varying behavior of the linearized time constant and process gain. It is clear that the heat transfer coefficient (Case 3) has a large effect on the linearized time constant and a moderate effect on the linearized gain. The kinetic terms in Q_r (Case 1) and the viscosity multiplying factor (Case 2) both have a large impact on the linearized process gain and show a lesser influence on changes in the time constant.

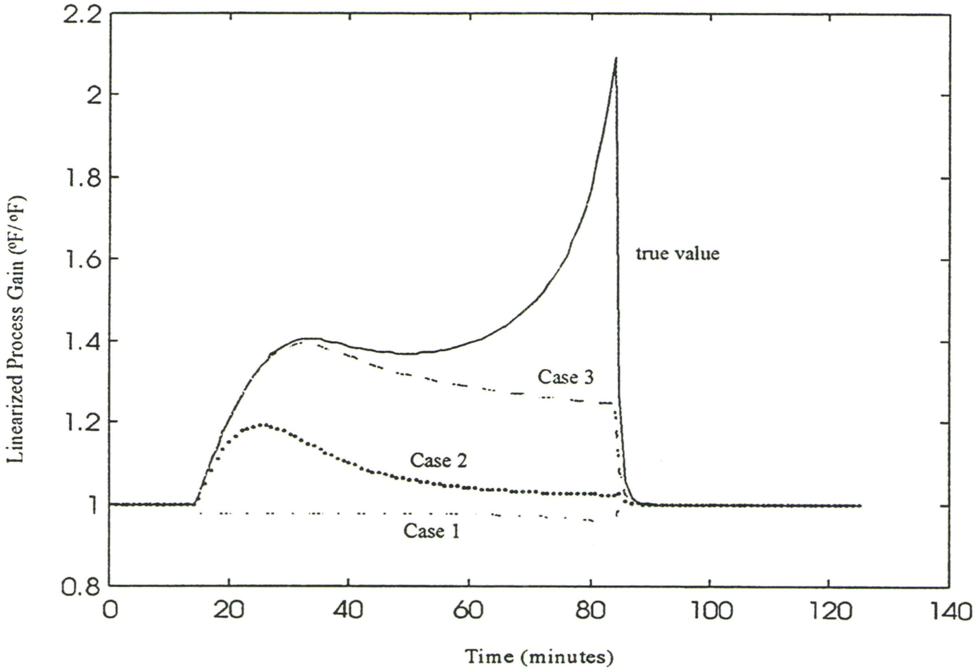


FIGURE 3.3: CONTRIBUTIONS OF PARAMETERS TO LINEARIZED PROCESS GAIN

While it is interesting to evaluate the effect each individual component has on the linearized parameters, there is a more important interpretation. In most realistic systems, one has limited process knowledge. Each of the above three cases describes a situation where a piece of process information is missing. From the plots, it can be interpreted that limited process knowledge significantly alters the view of the system. With no knowledge of fouling (constant U value), the estimated time constant and process gain differ significantly from the 'true' linearized values. Not knowing the reaction kinetics or having no quantitative auto-acceleration information similarly affects the behavior of the estimates.

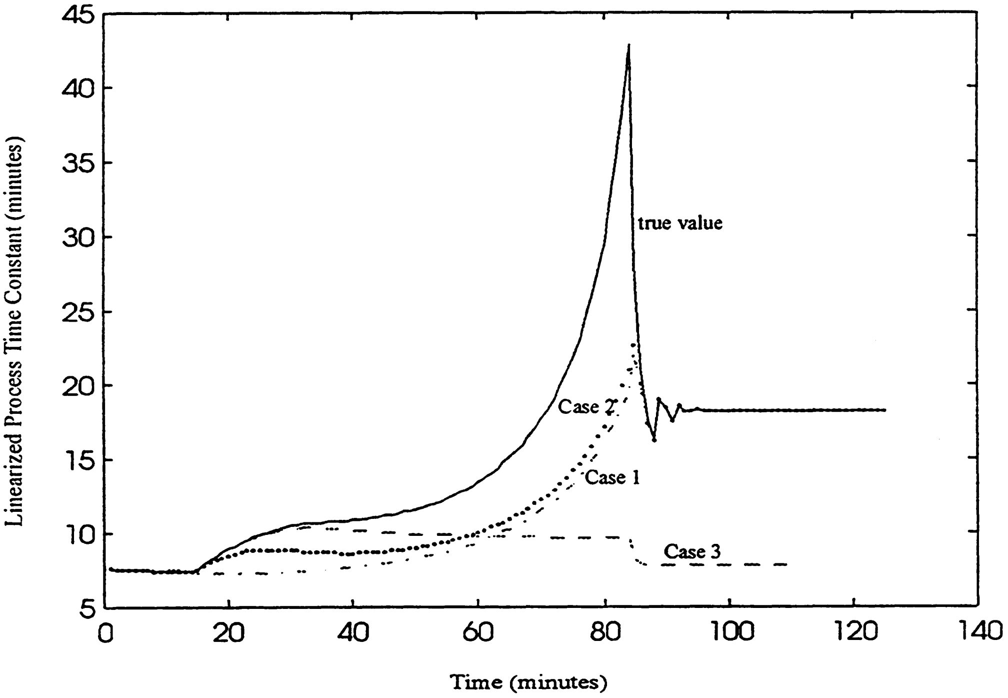


FIGURE 3.4: CONTRIBUTIONS OF PARAMETERS TO LINEARIZED TIME CONSTANT

In summary, the individual contributions to the linearized time constant and gain have been examined. From the analysis, it is evident that limited process knowledge has a significant impact on the estimates of the linearized parameters.

3.2.3 PRODUCT TWO CONSIDERATIONS

Since the Chylla Haase semi-batch reactor has two products, a similar analysis can be performed for the second product. For the first feed period, the results are consistent with findings in section 3.2.2. However, early in the second feed period, the heat transfer coefficient drops to zero due to extensive fouling. Beyond that point, the linearized time constant and gain acquire non-real values as there is no longer a relationship between the controlled and manipulated variables.

A suggested remedy (Chylla, 1994) for the general problem of extensive fouling is to *increase* the jacket temperature in order to *cool* the reactor. This counter-intuitive measure is intended to increase the wall temperature, decrease the viscosity of the polymer at the wall and increase the heat transfer coefficient. Therefore, although the jacket temperature is increasing (thereby reducing the heat transfer driving force), the reactor will experience more cooling due to an increased heat transfer coefficient. Simulations using this counter-intuitive control action proved unsuccessful. Figure 3.5 explains why:

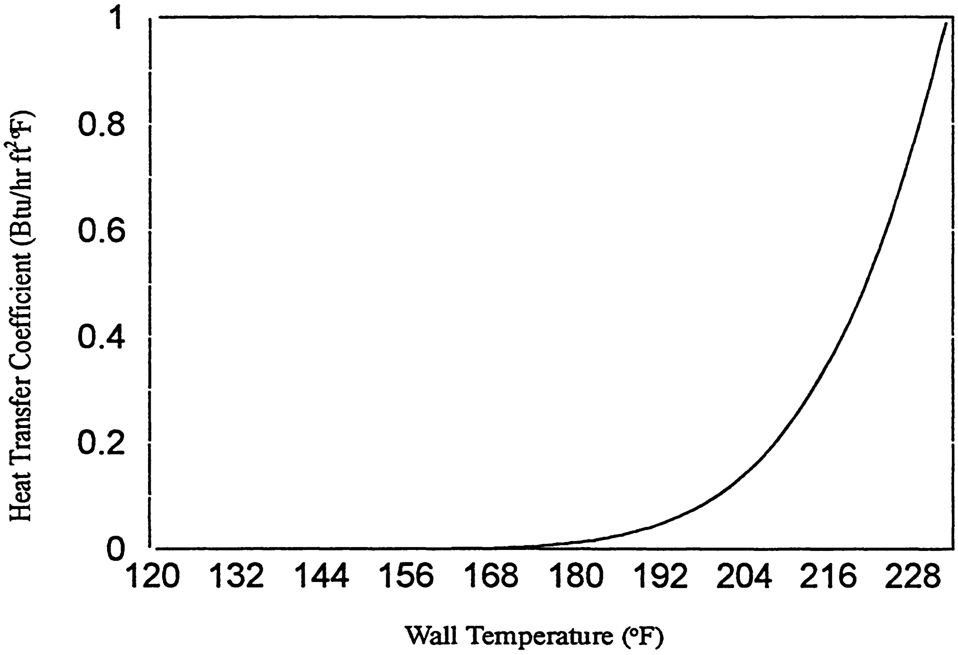


FIGURE 3.5: EFFECT OF WALL TEMPERATURE ON HEAT TRANSFER COEFFICIENT

Figure 3.5 shows the effect of wall temperature at a constant value of the fraction solids (set to its value at the beginning of the second feed). In order to obtain a value of the heat transfer coefficient of only 1 Btu/(hr ft^2 °F), a wall temperature of 228 °F is needed. In other words, the jacket temperature will need to be well above the reactor setpoint of 176 °F in order to get measurable heat transfer. At that point, the increased heat transfer will heat, not cool, the reactor.

From this brief study, it is clear that the problems with Product Two (during the second feed) cannot be solved with a different control algorithm. The situation is a process design problem, and will require an appropriate design solution.

3.3 Proportional Integral Derivative Control

3.3.1 Tuning Rules

In order to reduce the time and expense of tuning a PID controller on-line, tuning rules are helpful in establishing initial tuning parameter estimates. Tuning rules from Smith and Corripio, 1985 (ITAE, ISE, IAE) and Marlin, 1995 (Ciancone) are discussed. Only PID tuning will be considered as the controller with derivative action outperforms the PI controller. A brief discussion on each of the rules is included in this section.

The rules in Smith and Corripio (1985) were developed by Lopez and coworkers and provide the tuning given different error functions to minimize. Inherent in the rules are the following assumptions:

- first order plus deadtime model between the manipulated and controlled variable;
- disturbance model is the same as the process model;
- fraction deadtime (θ/τ) is between 0.1 and 1;

The Integral of the Absolute Error (IAE) tuning places equal weight on all errors while the Integral of the Squared Error (ISE) weights larger errors more heavily. The Integral of the Time-Weighted Absolute Error (ITAE) includes a penalty for the elapsed time, to reduce the oscillatory behavior often noted with the ISE tuning. The ISE criteria of penalizing the largest errors is most compatible with the semi-batch performance requirement (±1 °F). Of these three, the ISE rules provide the tightest tuning and the ITAE rules give the loosest tuning.

The Ciancone correlations (Marlin, 1995) also assume the process is first order with deadtime and the disturbance transfer function is identical to the process transfer function. However, the tuning parameters are derived to provide the minimum IAE given:

- errors of 25% in the model parameters
- limits on the manipulated variable moves
- noise on the controlled variable

Because of the addition three criteria, the Ciancone correlations result in tuning that is less tight than the others. However, the Ciancone tuning tends to be much more realistic since few processes meet the rigid assumptions of the Lopez rules.

3.3.2 Product Specific Tuning Parameters

For good performance, each different product requires unique tuning parameters. Both products are tuned for ideal conditions (batch one, summer). The robustness of the PID controller for winter and batch five runs is evaluated in Chapter 7. Noise is also added to the temperature measurements in Chapter 7.

Initial tuning parameters are based on average values of the linearized time constant and process gain from section 3.2. Alternatively, a step test could be performed with the reactor filled with the initial loading contents. An estimate of the process deadtime is also required. For sampled data systems, as a rule of thumb the 'deadtime' is taken as the actual process deadtime plus one half the sampling interval. Since the semi-batch reactor has only a small amount of deadtime, a value of fraction deadtime (θ/τ) of 0.1 is assumed to apply the rules. Table 3.1 summarizes the initial tuning values from each rule, and the final tuning for a well tuned PID controller for Product One. Due to the known model error, significant effort is directed towards fine tuning the PID. The final tuning of the PID is the result of numerous trial and error simulations.

Table 3.1: Tuning for Product One(τ_p = 12 min, K_p = 1.5, θ/τ = 0.1)

Tuning Rule	K_c	τ_i	τ_d
IAE (Lopez)	7.9	2.44	0.42
ISE (Lopez)	8.8	1.84	0.0552
ITAE (Lopez)	8.0	2.61	0.462
Ciancone	0.93	2.16	0
By Trial and Error	4	8	0.3

As Table 3.1 shows, the Lopez tuning rules predict much tighter tuning than is actually implemented. Conversely, Ciancone predicted a smaller gain, but more integral action than the actual best tuning.

The PID algorithm is implemented in velocity form with anti-reset windup precautions (equation 3.21):

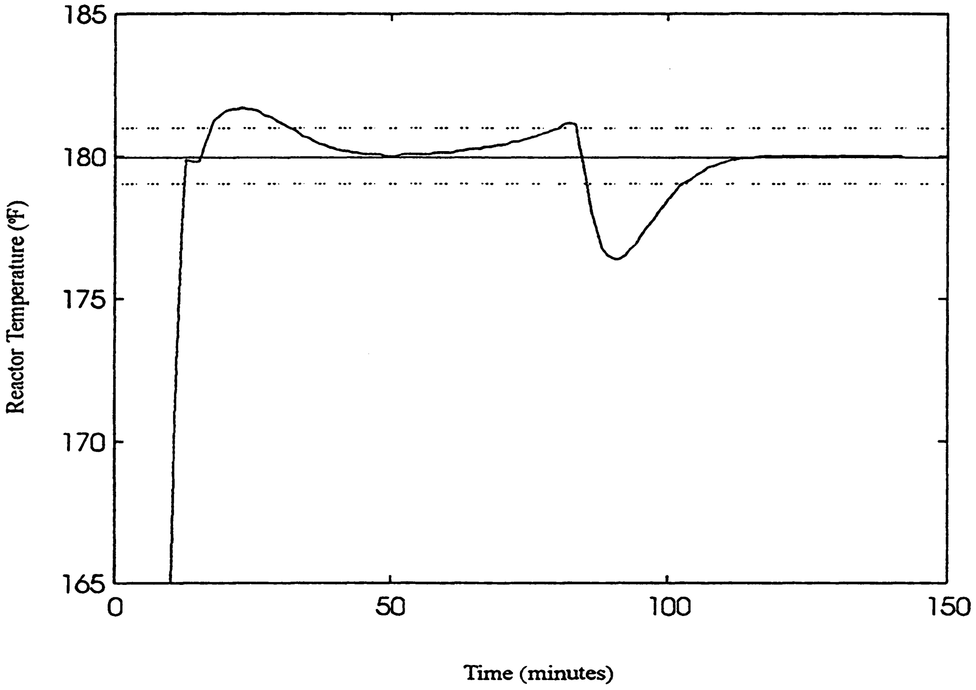
$$\Delta T_{jinsp} = K_c\left(E_n - E_{n-1} + \frac{\Delta t E_n}{\tau_I} - \frac{\tau_d}{\Delta t}(T_n - 2T_{n-1} + T_{n-2})\right)$$
$$T_{jinsp}(n) = T_{jinsp}(n-1) + \Delta T_{jinsp} \tag{3.21}$$

The implemented value of T_{jinsp} is retained for use at the next execution interval in order to avoid windup. At the beginning of the batch, T_{jinsp} is initialized to the ambient temperature.

Figure 3.6 shows the performance of the well tuned PID controller for Product One, batch one, summer conditions (both the controlled and manipulated variables are shown). The setpoint for Product One is 180 °F. The limits of good control, ± 1°F, are also plotted.

The recipe for Product One calls for one monomer feed period. The reactor is initially charged with water and some polymer and the reactor contents heated. The monomer feed begins when the temperature is within one degree of setpoint and is fed at a rate of one pound per minute. The feed is stopped exactly seventy minutes from the time that it is started. As Figure 3.6 shows, an unacceptably large temperature deviation results when the feed is stopped.

A few comments will be made here about convention. The top plot shows the reactor temperature. It should be noted that the reactor contents are initially at the ambient temperature (90 °F for summer), however only the upper portion of the temperature trajectory is shown. Temperature figures will be presented in this manner throughout the thesis. In Figure 3.6, the setpoint is plotted, however it will not be shown on future temperature figures in order to simplify the presentation. On the bottom plot, the manipulated variable (inlet jacket temperature setpoint) is at its upper bound of 350 °F for the first few minutes of the batch. The line is plotted but is being masked by the plot borders (this is seen in many future plots of the manipulated variable).



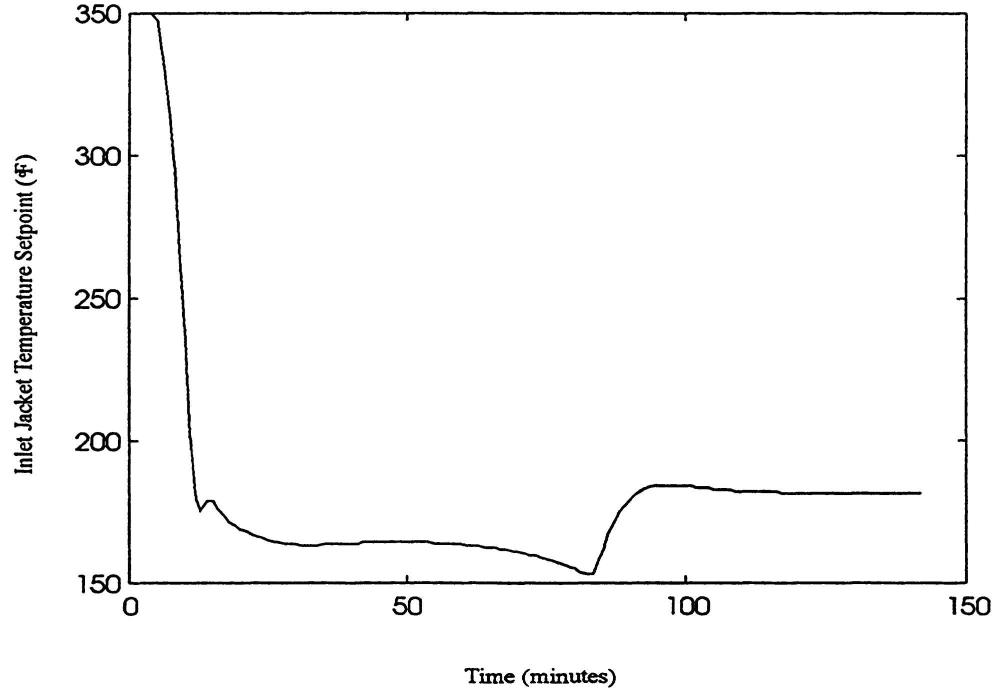
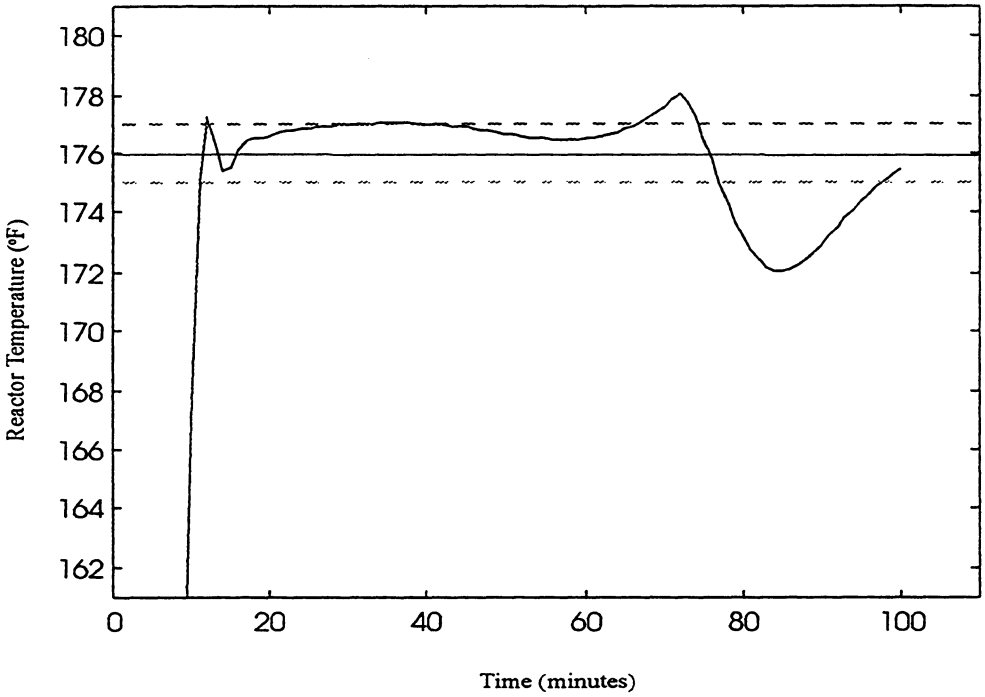


FIGURE 3.6: PROPORTIONAL INTEGRAL DERIVATIVE CONTROL (PRODUCT ONE)

Table 3.2 and Figure 3.7 show the tuning for Product Two and the controller performance, respectively. The setpoint for product two is 176 °F and is shown on the figure. Note that only the first feed period is considered for Product Two, due to considerations discussed in section 3.2.3. As well, only the upper portion of the reactor temperature behavior is shown, however the controller is heating the contents from ambient temperature. The initial value of the inlet jacket temperature setpoint is masked by the plot border. Once again, stopping the monomer feed causes a large temperature disturbance.

Table 3.2: Tuning for Product Two ($\tau_p = 20$, $K_p = 2.1$, $\theta/\tau = 0.1$)

Tuning Rule	K_c	τ_i	τ_d
IAE (Lopez)	5.9	4.0	0.70
ISE (Lopez)	6.6	3.0	1.1
ITAE (Lopez)	6.0	4.3	7.7
Ciancone	0.7	3.6	0
By Trial and Error	5	6	0.3



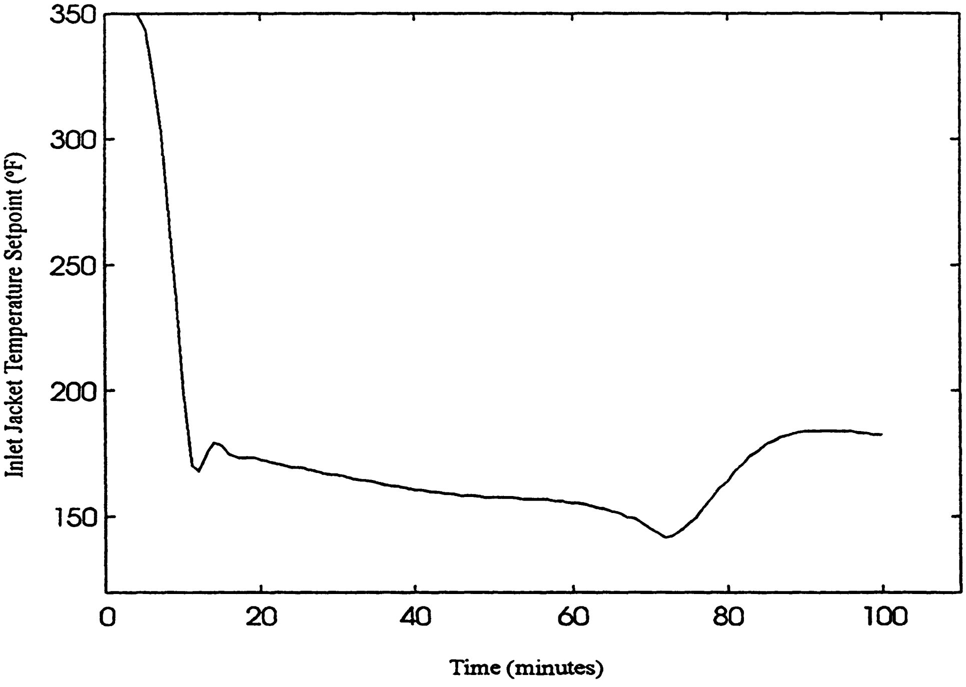


FIGURE 3.7: PROPORTIONAL INTEGRAL DERIVATIVE CONTROL (PRODUCT TWO)

In this section, the tuning for the PID controllers has been outlined. Simulations indicate that a well tuned PID controller cannot maintain the temperature within the bounds for good control.

3.4 ADAPTIVE PROPORTIONAL INTEGRAL DERIVATIVE CONTROL

The time varying behavior of the semi-batch reactor makes it an ideal candidate for adaptive control. The PID controller in section 3.3 is tuned for average performance over a range of process conditions. In this section, an adaptive PID controller is developed to account for the time varying behavior. Its performance is compared to that of the well tuned PID controller.

There are many different formulations for an adaptive PID controller. Building on the analysis in the previous sections, the linearized time constant and gain are chosen as measures of the time varying behavior of the reactor. It should be emphasized that many formulations of the adaptive PID are possible, however by using the linearized parameters, it is possible to take advantage of the PID tuning rules discussed in section 3.3. Recall that the time constant increases during the feed period. Therefore it is expected that the tuning will be tightest at the beginning of the batch. Detuning will be required throughout the batch as the process becomes more difficult to control. From this, one may predict that the benefits of adaptive control will primarily be seen in a reduction of the initial overshoot.

The Lopez ITAE rules are used as a starting point for the adaptive controller tuning. The rules are in the form:

$$K_c = \frac{a_1}{K_{process}}\left(\frac{\theta}{\tau_{process}}\right)^{b1} \tag{3.22}$$

$$\tau_I = \frac{\tau_{process}}{a_2}\left(\frac{\theta}{\tau_{process}}\right)^{b3} \tag{3.23}$$

$$\tau_d = a_3\tau_{process}\left(\frac{\theta}{\tau_{process}}\right)^{b3} \tag{3.24}$$

The fraction deadtime is set at the minimum value allowable ($\theta/\tau=0.1$). Estimates of the linearized time constant and process gain to be used in equations (3.22) through (3.24) are provided by equations (3.19) and (3.20). Initially, it is assumed that all process parameters (e.g. U, viscosity) are measured and available to the adaptive controller at each control step.

The Lopez ITAE tuning gives very oscillatory closed loop behavior, particularly when the monomer feed is started. Therefore, the rules are 'detuned' until good performance is achieved. Table 3.3 summarizes the adaptive PID controller tuning for Product One. Again, significant effort was directed at fine tuning the adaptive PID controller, due to the known model error. Figure 3.8 shows how the controller gain, integral time constant and derivative time constant are adapted based on the trial and error tuning in Table 3.3.

Table 3.3: Tuning for Adaptive PID Control (Product One)

	K_c	τ_i	τ_d
ITAE tuning	$12/K_p$	0.217τ	0.0385τ
Trial and Error tuning	$6/K_p$	0.7τ	0.0385τ

Figure 3.9 shows the performance of the adaptive controller when all process parameters are assumed known in evaluating the linearized time constant and gain. A Product One, summer, batch one simulation is plotted. The corresponding PID with constant parameters is shown as a dotted line for comparison purposes.

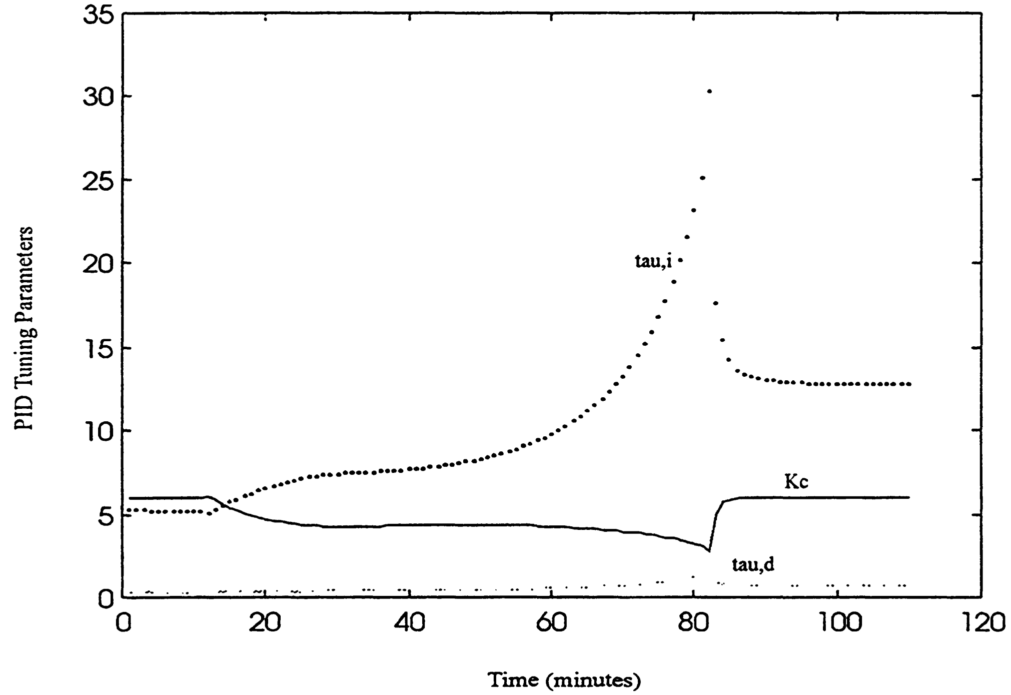


FIGURE 3.8: ADAPTATION OF PID TUNING PARAMETERS

The performance of the adaptive controller is only marginally better than that of the PID controller, and is slightly oscillatory when the feed starts. Note that Figure 3.9 demonstrates an ideal situation for the adaptive controller: all time varying process parameters are assumed known. However, realistically, not all of the process variables such as viscosity and heat transfer coefficient will be available to the adaptive controller. From the short study in section 3.2.2, it is known that limited process knowledge significantly affects one's perception of the process. It is important to see how the adaptive controller performs with limited process knowledge.

Unless extensive modeling effort is initiated, only rough estimates of the heat transfer coefficient and the auto-acceleration effect will be available (at best). If reasonable estimates are available from process experience, they may be used. Otherwise, a procedure to obtain an average heat transfer value using on-line energy balances is described in Chapter 4. To evaluate the performance of the adaptive controller, the value of U from a batch three, summer simulation has been parameterized off-line as a function of the mass of monomer fed. The parameterized value of U is used in the calculation of the linearized time constant and process gain. It is also assumed that the viscosity multiplying factor on rate of reaction (auto-acceleration) is unknown (value is assumed to be one).

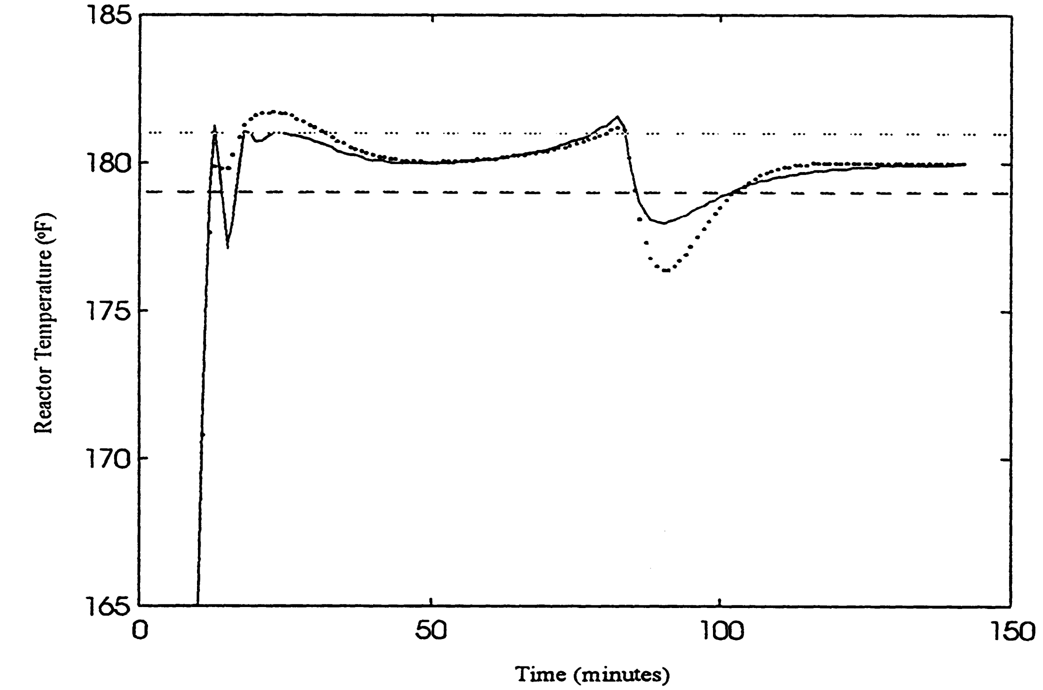


FIGURE 3.9: ADAPTIVE PID CONTROL (PRODUCT ONE)

The performance of the adaptive controller with the above assumptions is shown in Figure 3.10. Also shown on the plot is the adaptive controller of Figure 3.9. There is a degradation of control, particularly at the beginning of the batch.

It was noted at the beginning of this section that the benefits of the adaptive control will most likely be seen at the beginning of the batch. Trial and error simulations on Product One show that the reactor is susceptible to oscillations when the feed starts.

Clearly, starting the feed introduces a severe disturbance into the system. As a result, the adaptive controller must be detuned to accommodate the nature of the disturbance at the beginning. However, the increasing time constant of the process indicate that the process becomes more difficult to control. Therefore, the adaptation detunes the controller, from the initial values, over the course of the batch. The combination of severe disturbance at the onset and more difficult process to control at the end results in a situation where this adaptive PID does not provide any benefit over the constant parameter PID.

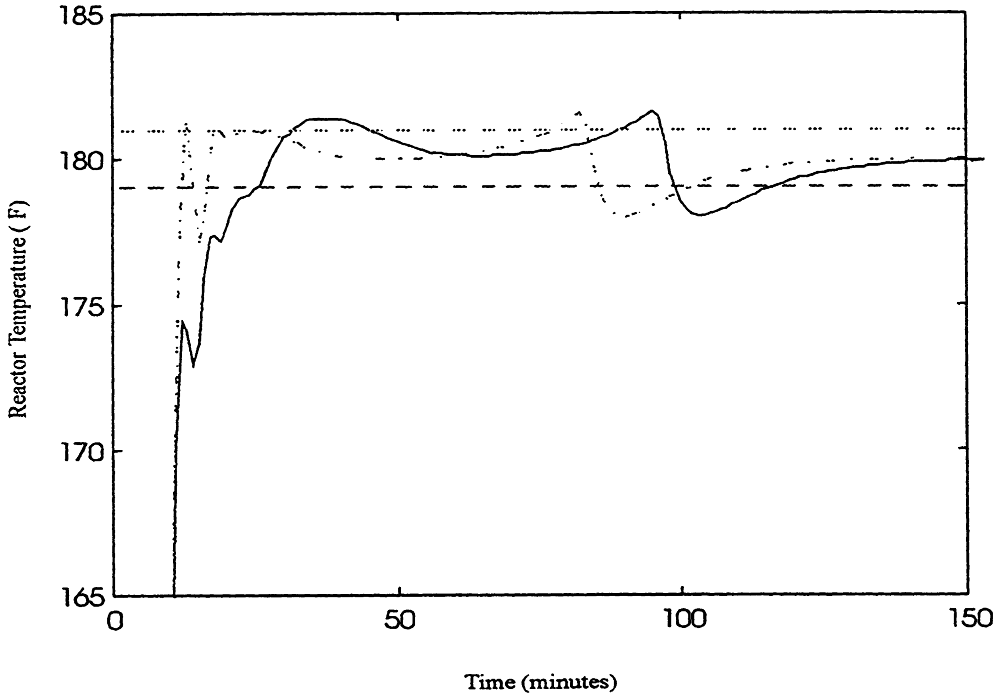


FIGURE 3.10: EFFECT OF LIMITED PROCESS KNOWLEDGE ON ADAPTIVE PID CONTROL

It should be re-iterated that there are many different formulations of an adaptive PID controller, and only one is presented here. Other options include adapting the tuning parameters based on a different measure of process changes. It is possible that with enough effort, an adaptive controller that outperforms the well tuned PID can be derived. However, it was decided not to pursue this option any further because of the multi-batch, multi-product nature of the system. The empirical nature of the adaptive PID controller tuning makes it less flexible than a controller based on mechanistic models.

In summary, for the semi-batch reactor considered, the adaptive PID control performs only slightly better than the PID controller. It is unable to maintain the temperature within one degree of setpoint for Product One, batch one, summer conditions. Based on the findings of this initial study and considering other process characteristics, the adaptive PID controller will not be pursued further.

CHAPTER 4

FEEDFORWARD COMPENSATION

4.1 INTRODUCTION

In this chapter, a second method of adding information to the PID algorithm is explored. The batch runs of the well tuned PID for Product One show that the temperature exceeds the ± 1 °F limits twice. Initially, the temperature overshoots the setpoint due to the heat released by reaction. It then exhibits a sharp drop when the feed stops. In this chapter, feedforward compensation is added to the well tuned PID to counter the effect of these disturbances. The controller is discussed for Product One, batch one, summer conditions only; the results are extended to other batches and products in Chapter 7. It is seen that the combined feedforward/feedback controller is able to maintain the temperature within one degree of setpoint for a Product One, batch one, summer simulation.

4.2 FEEDFORWARD FOR THE FEED STOPPAGE

4.2.1 THEORY

The feed stoppage clearly causes the temperature to deviate the most from setpoint. Therefore, feedforward for this disturbance will be considered first. Simple linear theory provides a platform for the discussion and is reviewed in this section.

The semi-batch reactor may be generally represented by:

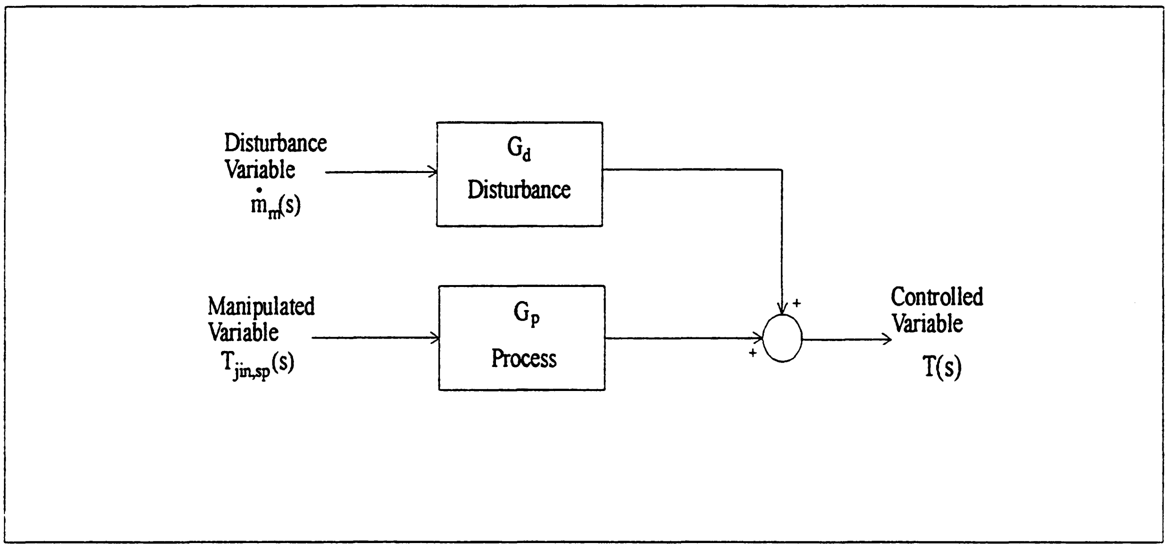


FIGURE 4.1: BLOCK DIAGRAM OF A PROCESS AND DISTURBANCE

For the effects of the feed disturbance to be exactly canceled:

$$\dot{m}_m(s)G_d(s) + T_{j,in,ff}(s)G_p(s) = 0 \tag{4.1}$$

Therefore:

$$T_{j,in,ff}(s) = \frac{-G_d(s)}{G_p(s)}\dot{m}_m(s) \tag{4.2}$$

If expressions for G_d and G_p are available, an approximate expression for the feedforward action can be formulated. Models of G_d and G_p are required for each product. Values are given for Product One, however the results for Product Two are similar.

For Product One, the relationship between the inlet jacket temperature and reactor temperature at the time of the disturbance can be approximated by a first order transfer function with a gain of one and a time constant of eighteen minutes (section 3.2). A similar result could also be obtained by doing an open loop step test on the batch at the end of the run.

An expression for the disturbance transfer function is obtained with a step test. On a batch one, summer simulation, the master temperature controller is turned off when the feed is stopped. The ensuing response may be approximated by a first order transfer function with gain of minus twenty-nine °F/lb/min and time constant of twenty minutes.

To derive the feedforward controller, G_d and G_p in equation (4.2) are replaced with their respective transfer functions:

$$T_{jin,ff}(s) = -\frac{\left(\dfrac{29}{20s+1}\right)}{\left(\dfrac{1}{18s+1}\right)}\frac{\Delta \dot{m}_m}{s}$$

$$= -29\frac{18s+1}{20s+1}\frac{\Delta \dot{m}_m}{s} \tag{4.3}$$

For a step in the feed from 1 lb/min to 0 lb/min (Product One recipe), equation (4.3) in the time domain (Marlin, 1995) is:

$$T_{jin,ff}(t) = -\frac{K_d}{K_p}\left(1+\left(\frac{\tau_p-\tau_d}{\tau_d}\right)e^{-t/\tau d}\right)\Delta \dot{m}_m = 29\left(1+\frac{18-20}{20}e^{-t/20}\right) \tag{4.4}$$

Equation (4.4) provides compensation for a specific disturbance (the feed stoppage). Feedback is still required for other unmeasured or unknown disturbances, and to account for any error in the feedforward compensation. The setpoint calculated by the combined feedforward/feedback algorithm is the sum of the individual components:

$$T_{jin,sp}(t) = T_{jin,sp}(fb) + T_{jin,sp}(ff) \tag{4.5}$$

Equations (4.3) and (4.4) hold several pieces of information. First, equation (4.3) is a lead-lag transfer function. As shown in equation (4.4), the time domain interpretation of a lead-lag (for a step input) is an exponential relationship. Second, the disturbance and the process time constants are very close in value. Therefore, one might expect that a steady state feedforward algorithm (that is, a constant value) would provide good compensation as well. Performance of both dynamic and steady state compensation is discussed in the next section.

4.2.2 Steady State and Dynamic Feedforward Compensation

In this section, the performance of the two types of feedforward control is shown for a Product One, batch one, summer simulation. First, the steady state (constant value) compensation is discussed. The dynamic (exponential) compensation is then compared to the steady state compensation.

From section 4.2.1, the expression for steady state feedforward compensation (Product One value) is:

$$T_{j,in,sp,ff}(t) = \frac{-K_d}{K_p}\Delta \dot{m}_m = -29\Delta \dot{m}_m \tag{4.6}$$

Due to model error, the feedforward gain in equation (4.6) does not give the best compensation possible. Trial and error tuning of the feedforward gain shows that a value of -23 °F/lb/min results in the best control. Figure 4.2 shows the temperature control of the combined feedforward/feedback algorithm for a Product One, batch one, summer simulation. The same PID tuning constants are used in the feedback portion and a constant gain of -23 °F/lb/min is implemented in the feedforward portion. The feedback algorithm alone is shown as a dotted line for comparison.

Clearly, the feedforward action has improved control. The temperature is maintained above 179 °F when the feed stops. The difference in the manipulated variable behavior is also seen in Figure 4.2.

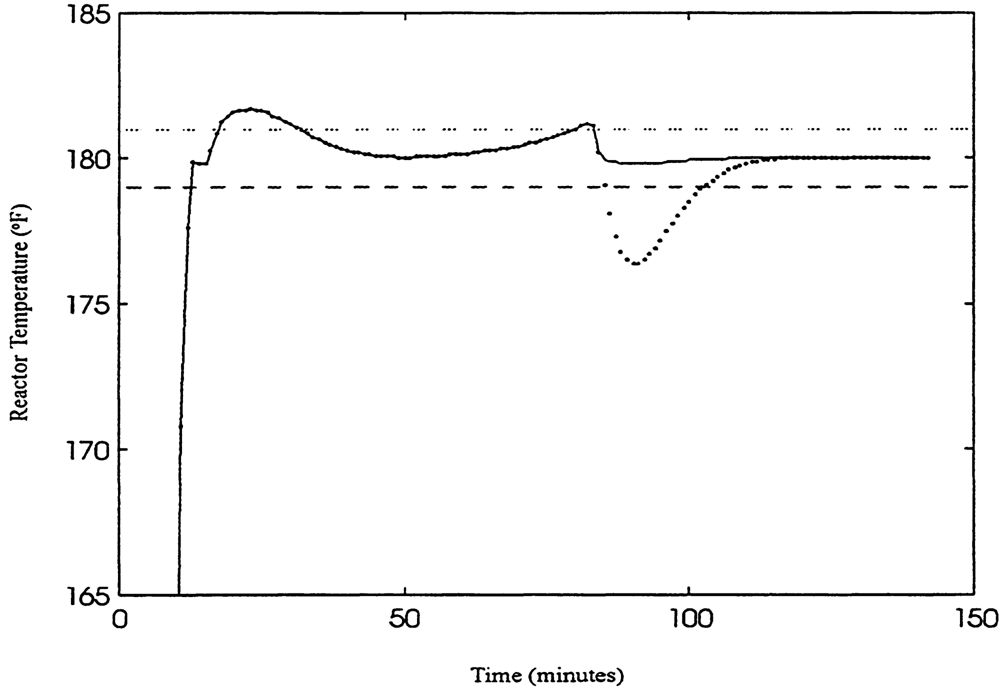
Now, the performance of the dynamic feedforward controller is shown and compared to the above steady state feedforward controller. The best dynamic feedforward compensation is also found by trial and error. Equation (4.4) gives the theoretical feedforward algorithm. Trial and error tuning results in:

$$T_{jin,sp,ff}(t) = -25(1 - 0.1e^{-0.05t})\Delta \dot{m}_m \tag{4.7}$$

where t in equation (4.7) is the time from when the feed stops.

Equation (4.7) assumes that the feedrate is known and that the control system has knowledge of when the feed stops. $T_{jin,sp,ff}$ is calculated every minute and implemented with the feedback $T_{jin,sp}$. Figure 4.3 shows the performance of the dynamic feedforward algorithm (coupled with feedback control).

Comparing Figure 4.2 and Figure 4.3, the steady state feedforward compensation and the dynamic feedforward compensation both give about the same control. This is consistent with the prediction in section 4.2.1.



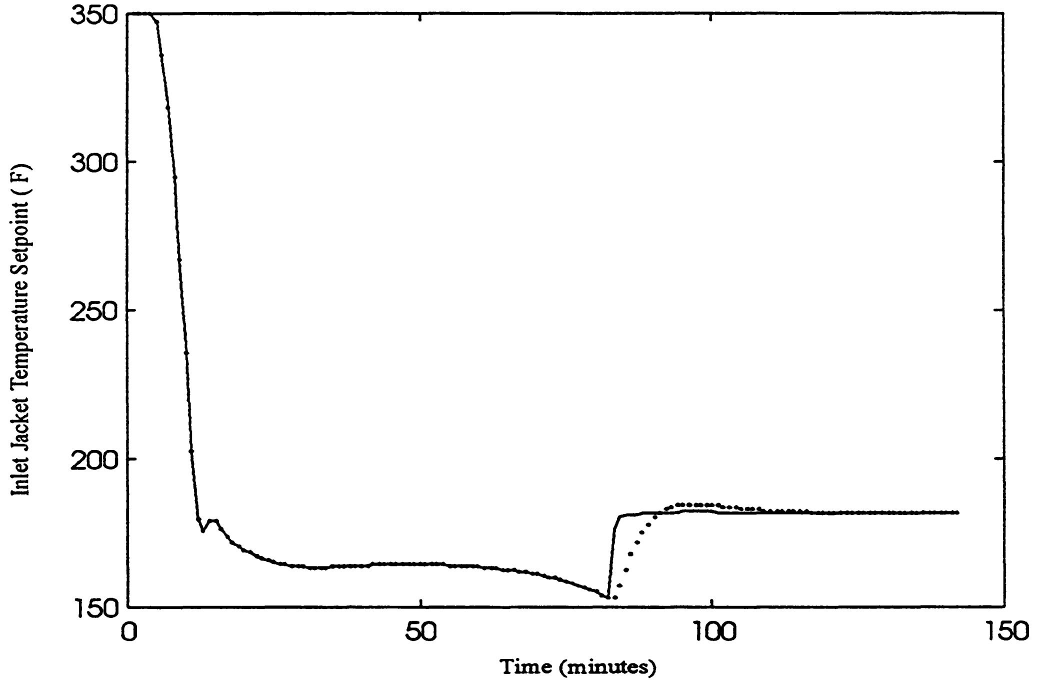
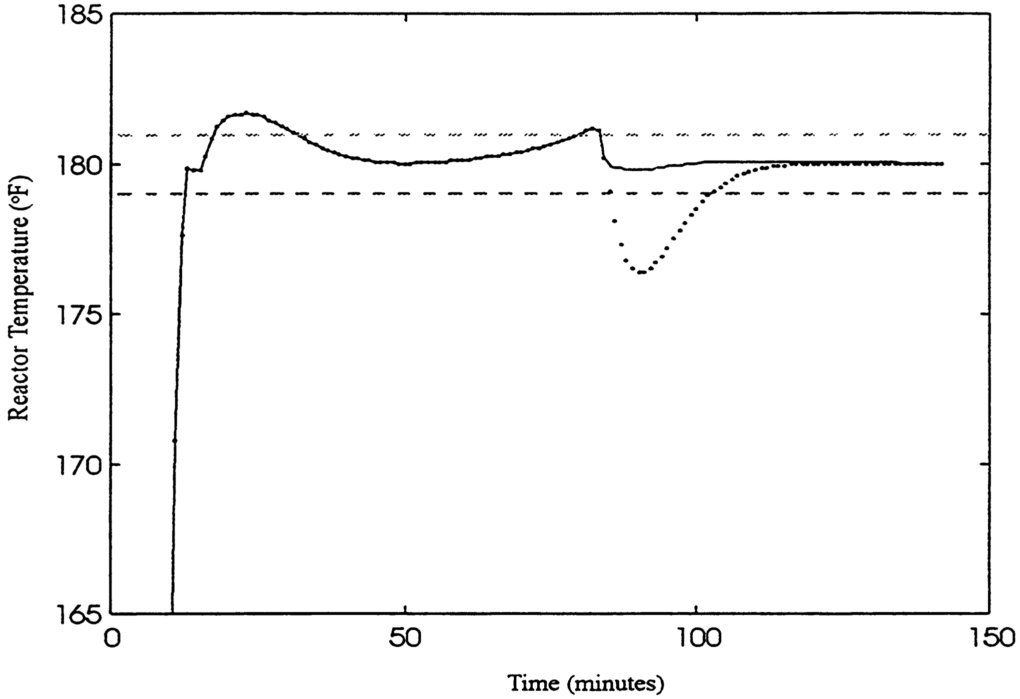


FIGURE 4.2: FEEDFORWARD/FEEDBACK CONTROL: STEADY STATE COMPENSATION FOR FEED STOPPAGE



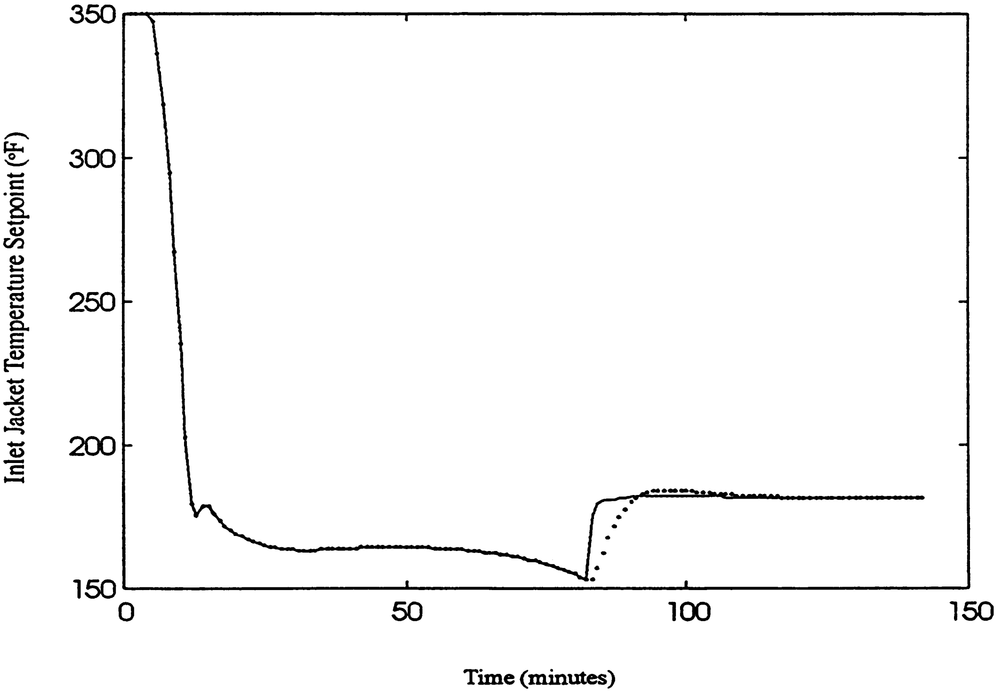


FIGURE 4.3: FEEDFORWARD/FEEDBACK CONTROL: DYNAMIC COMPENSATION FOR FEED STOPPAGE

The robustness of the dynamic feedforward/feedback controller was also studied. In the interest of brevity, the results are mentioned but no simulations are shown. The study was performed to evaluate how sensitive the feedforward/feedback controller is to errors in the process and disturbance model time constants. Several different values of τ_d and τ_p were tested in equation (4.4) and the controller implemented. It was found that the feedforward/feedback performs well if the relative sizes of the time constants are correct (that is, $\tau_d/\tau_p > 1$). Therefore, the feedforward/feedback algorithm is robust to errors in the models, as long as the relative sizes of the time constants are correct.

Despite the robustness of the dynamic feedforward controller, the constant gain feedforward algorithm is applied in all future feedforward simulations. Since performance is not seriously affected, it is desirable to use the simplest algorithm possible.

4.3 Feedforward Compensation for the Heat Release

4.3.1 Theory

This section presents a second form of feedforward control. Recall that the temperature exhibits an initial overshoot due to the heat of reaction. If it is critical to reduce the overshoot, feedforward compensation for the heat released, Q_r, may be considered.

To compensate for Q_r, a feedforward transfer function and an estimate of the heat released are required. The transfer function is derived in this section. The relationship between the heat released and reactor temperature can be estimated using the overall energy balance for the reactor (same structure for both products):

$$\sum_{i=1}^{3}(m_i C_{pi})\frac{dT}{dt} = \dot{m}_m C_{pm}(T_{feed} - T) - UA(T - T_{jout}) + Q_r \tag{4.8}$$

Rearranging equation (4.8):

$$\left(\frac{\sum(m_i C_{pi})}{\dot{m}_m C_{pm} + UA}\right)\frac{dT}{dt} + T = \frac{\dot{m}_m C_{pm}}{\dot{m}_m C_{pm} + UA} T_{feed} + \frac{UA}{\dot{m}_m C_{pm} + UA} T_{jout} + \frac{1}{\dot{m}_m C_{pm} + UA} Q_r \quad (4.9)$$

Recall from section 3.2.1, the jacket dynamics between T_{jin} and T_{jout} are very fast and may be represented by a steady state equation:

$$T_{jout}(t) = K_{j1} T_{jin}(t) + K_{j2} T(t) \quad (4.10)$$

Substituting (4.10) into (4.9) gives:

$$\tau_p \frac{dT}{dt} + T = K_{p1}(K_{j1} T_{jin} + K_{j2} T) + K_{p2} T_{feed} + K_{pQ} Q_r \quad (4.11)$$

where, in this case:

$$\tau_p = \frac{\sum(m_i C_{pi})}{\dot{m}_m C_{pm} + UA} \quad (4.12)$$

$$K_{p1} = \frac{UA}{\dot{m}_m C_{pm} + UA} \quad (4.13)$$

$$K_{p2} = \frac{\dot{m}_m C_{pm}}{\dot{m}_m C_{pm} + UA} \quad (4.14)$$

$$K_{pQ}=\frac{1}{\dot{m}_m C_{pm}+UA} \tag{4.15}$$

$$K_{j1}=\frac{\dot{m}_c C_{pc}}{\dot{m}_c C_{pc}+UA} \tag{4.16}$$

$$K_{j2}=\frac{UA}{\dot{m}_c C_{pc}+UA} \tag{4.17}$$

Rearranging equation (4.11) for the gain between Q_r and T:

$$\frac{\tau_p}{1-K_{p1}K_{j2}}\frac{dT}{dt}+T=\frac{K_{p1}K_{j1}}{1-K_{p1}K_{j2}}T_{jin}+\frac{K_{p2}}{1-K_{p1}K_{j2}}T_{feed}+\frac{K_{pQ}}{1-K_{p1}K_{j2}}Q_r \tag{4.18}$$

From equation (4.18), one obtains an expression for the transfer function (G_d) between the disturbance Q_r and the controlled variable T:

$$T(s)=\frac{\left(\frac{K_{pQ}}{1-K_{p1}K_{j2}}\right)}{\left(\frac{\tau_p}{1-K_{p1}K_{j2}}\right)s+1}Q_r(s) \tag{4.19}$$

As with the feed stoppage, this is a first order disturbance transfer function. Therefore, the feedforward algorithm will be a lead-lag transfer function. In order to improve the robustness and the simplicity of the feedforward/feedback controller, only a steady state (constant gain) feedforward controller will be considered. The feedforward algorithm is:

$$T_{jin,sp,ff}(t) = -\frac{K_d}{K_p} Q_r(t) \tag{4.20}$$

Expressions for the process and disturbance gains can be obtained from equation (4.18) and (4.19), respectively:

$$T_{jin,sp,ff} = -\frac{\dfrac{K_{pQ}}{1-K_{p1}K_{j2}}}{\dfrac{K_{p1}K_{j1}}{1-K_{p1}K_{j2}}} Q_r = -\frac{K_{pQ}}{K_{p1}K_{j1}} Q_r \tag{4.21}$$

With minimum process knowledge, a rough estimate of the feedforward gain may be obtained by noting that $K_{p1} \cong 1 \cong K_{j1}$. As well, $K_{pQ} \cong 1/(UA)_0$ allows for an order of magnitude estimate of equation (4.21). For Product One:

$$T_{jin,sp,ff} = -0.06 Q_r \tag{4.22}$$

Trial and error tuning is required to fine-tune the feedforward action.

In order to implement the feedforward algorithm, an estimate of the heat released is also required. Estimating Q_r is discussed in the following section. The transfer function derived above is combined with the estimate of Q_r in the section 4.4.

4.3.2 ON-LINE ENERGY BALANCE FOR ESTIMATING HEAT RELEASED

Several different options are available for estimating the heat released. Some of the most common are:

1/ Deterministic On-line Energy Balance: Rearranging the energy balance of the reactor results in an expression for the heat released. The expression can be evaluated on-line.

2/ Empirical Heat Released Estimator: For example, Juba and Hamer (1984) identified transfer functions between the two inputs, Heat Released and Inlet Jacket Temperature, and the output, Reactor Temperature. By rearranging this equation, they obtained an expression for the heat released as a function of the reactor temperature and inlet jacket temperature. The equation can be used to provide an estimate of the heat released.

3/ On-line Estimation Using an Extended Kalman Filter: the Heat Released can be represented as a stochastic state in the EKF and estimated on-line.

The deterministic on-line energy balance is chosen as it provides a simpler solution than the EKF and is more general than an empirical solution (important for multiple products). In this section, the derivation and implementation of the energy balance is considered (MacGregor (1986), Cott and Macchietto(1989)).

Once again, the starting point for estimating the heat released is the reactor energy balance common to both products (equation (4.8)):

$$\sum_{i=1}^{3}(m_i C_{pi})\frac{dT}{dt} = \dot{m}_m C_{pm}(T_{amb} - T) - UA(T - T_{jout}) + Q_r \tag{4.8}$$

Rearranging equation (4.8) for the heat released gives:

$$Q_r = \sum_{i=1}^{3}(m_i C_{pi})\frac{dT}{dt} - \dot{m}_m C_{pm}(T_{amb} - T) + UA(T - T_{jout}) \tag{4.23}$$

To apply equation (4.23), the mass and heat capacity of each component (or total mass and average heat capacity) in the reactor must be approximately known (the values will be different for different products). In the simulations, the actual values are used. In reality, good estimates are possible (please see Appendix C for a discussion).

The time derivative of the temperature is also required. The following equation is used:

$$\frac{dT}{dt} \cong \frac{3T^{k} - 4T^{k-1} + T^{k-2}}{2\Delta t} \tag{4.24}$$

Temperature measurements are assumed available every minute. For studies in this section, no noise is added to the system (it will be added later). It should be noted that proper treatment of very noisy measurements (for example by filtering) is imperative for the success of the estimation algorithm. Since temperature measurements are generally filtered, treating noisy data is not discussed for this application.

The final parameter in equation (4.23) that does not have a measurement associated with it is the heat transfer coefficient. If an estimate is not available from process experience, another simple procedure to get an average value of U for all batches of a single product is:

1/ Rearrange equation (4.23) and run an on-line energy balance during heating ($Q_r = 0$) to estimate U (or UA). Over a series of batches, this establishes a value for an average U_0.

2/ Run the on-line energy balance for U (or UA) again after the feed stops. Over a series of batches, this establishes a value for an average U_f.

3/ Using the average U_0 and U_f, parameterize U off-line as a function of the mass of monomer fed. Some knowledge of the shape of U may be known, or a simple linear function can be interpolated between the initial and final values.

For the thesis, the trajectory of the heat transfer coefficient from batch three is parameterized off-line to simulate the results of the above procedure. At each control interval, an estimate of U for Product One is given by:

$$\hat{U}(t) = 109.9 + 0.28 massfed_t - 0.17 massfed_t^{\,2} \qquad when\ \dot{m}_m \neq 0 \tag{4.25}$$

Figure 4.4 shows the values of U for all five batches of Product One (dashed lines), and the parameterized heat transfer coefficient from equation (4.25) (solid lines).

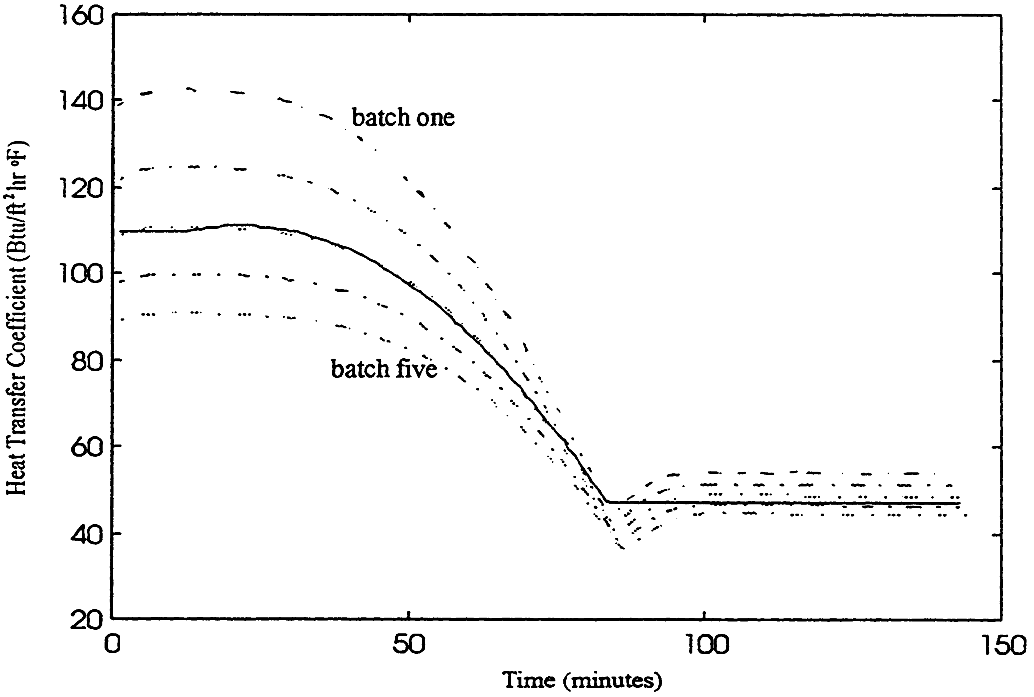


FIGURE 4.4: ACTUAL AND ESTIMATED HEAT TRANSFER COEFFICIENT

Figure 4.5 shows the performance of the heat released estimator when the true value of the heat transfer coefficient is used. In the simulation shown, the batch is run under Product One, batch one, summer conditions with the temperature being controlled by the PID of Chapter 3. The solid line is the actual heat released; the dotted line is the estimated value. Both the PID controller and the estimation algorithm are being executed every minute.

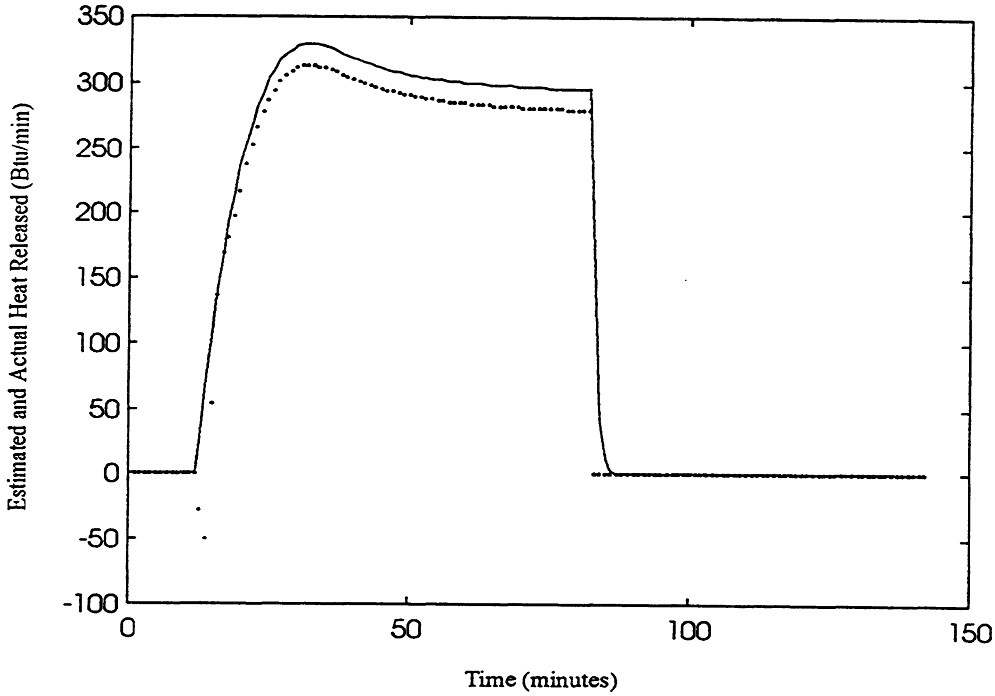


FIGURE 4.5: ESTIMATION OF HEAT RELEASED USING TRUE VALUE OF HEAT TRANSFER

Three items require comment. First, note the offset between the estimated and actual values. The reactor experiences some heat loss which is not accounted for in the on-line energy balance. If some rough estimate of the heat loss is available, it can be included in equation (4.23). If not, there will be offset in the estimate due to model error. Unless the heat loss is very large, this should not be a problem. The second item requiring note is the inverse response seen in the estimated Q_r value, possibly due to the discrete nature of the estimation algorithm. Indeed, when the temperature measurements are assumed every 0.6 seconds (instead of every minute), the suspicious inverse response disappears. In a true system, a logic check can be implemented to ensure the estimated value of Q_r is always positive (for an exothermic reaction). Finally, semi-batch emulsion reactors are often run under monomer starved conditions (Dimitratos et al.,1994). In such cases, the monomer concentration (and hence the heat released) can drop suddenly when the monomer feed stops. This process information is incorporated into the estimator by setting the estimated heat released to zero when the feed stops (instead of allowing it to track Q_r to a zero value).

Figure 4.6 demonstrates the performance of the heat release estimator when the parameterized value of the heat transfer coefficient from batch three is used in equation (4.23). Once again, the simulation is run under summer, batch one conditions. A logic check has been implemented to avoid negative estimates for Q_r:

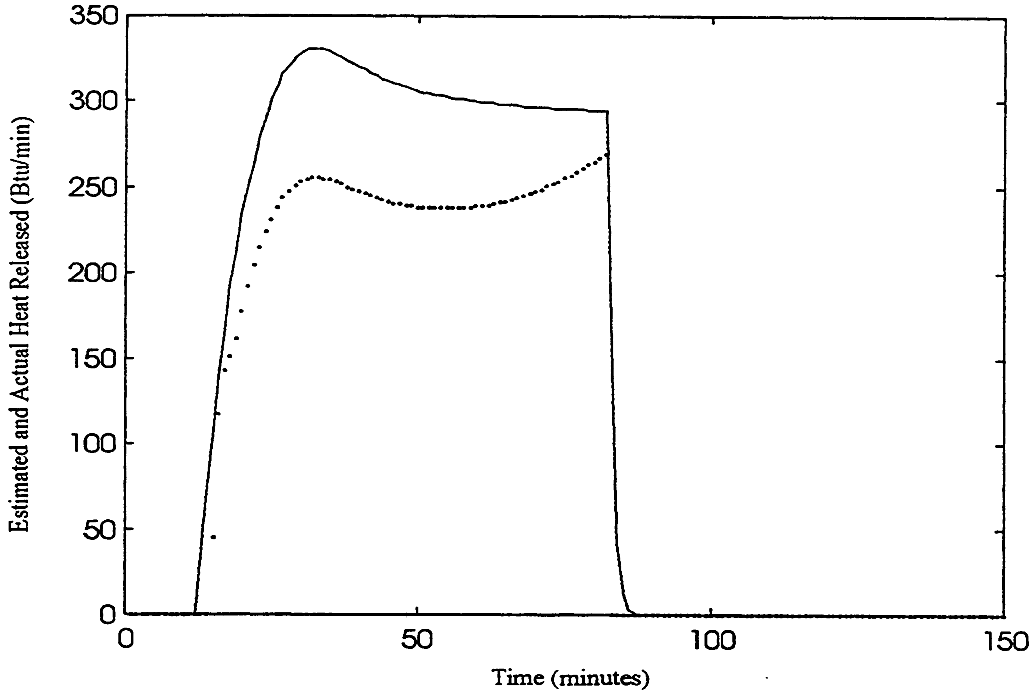


FIGURE 4.6: ESTIMATION OF HEAT RELEASED USING PARAMETERIZED VALUE OF HEAT TRANSFER

Having a parameterized value of U in the on-line energy balance results in Q_r being underestimated. The error is largest at the beginning, where the parameterized value of U differs most from the true batch one heat transfer coefficient.

In all future simulations, the heat release estimator is used with the average parameterized value of the average heat transfer coefficient.

4.3.3 Steady State Feedforward Compensation

In the previous two sections, a feedforward transfer function (for Product One) and an on-line heat release estimator were derived. In this section, the two parts are combined and the full feedforward controller implemented.

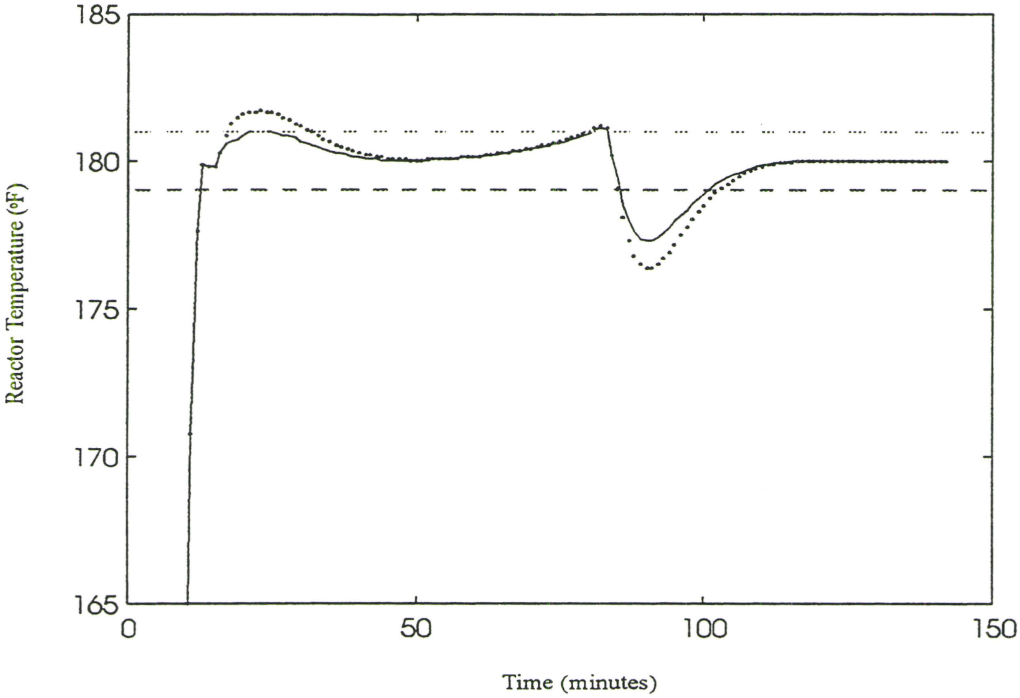
In section 4.3.1, the feedforward controller for Product One was derived:

$$T_{jin,sp,ff} = -0.06Q_r \tag{4.26}$$

Trial and error tuning is required to obtain the actual best value. With the on-line heat release estimator providing a value for Q_r, it was found that the best gain for the feedforward controller is:

$$T_{jin,sp,ff} = -0.025\hat{Q}_r \tag{4.27}$$

A larger gain in the feedforward controller results in ragged regulation of temperature. Figure 4.7 shows a Product One, batch one, summer simulation using equation (4.27) for feedforward. The solid line is the combined feedback/feedforward (for Q_r) controller. The dotted line is the control by a well tuned PID with no feedforward.



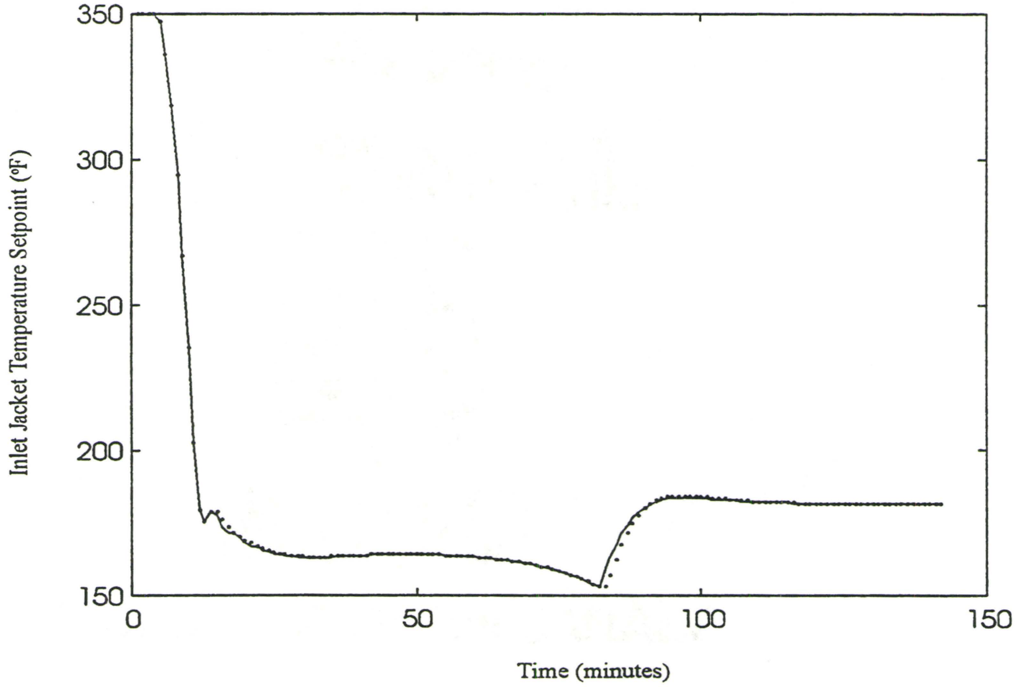


FIGURE 4.7: FEEDFORWARD/FEEDBACK CONTROL: COMPENSATION FOR HEAT RELEASED

As expected, the feedforward improves control over the well tuned feedback algorithm. The overshoot is reduced, and the temperature stays within 1 °F of setpoint until the feed stops. Note that by compensating for Q_r, the feed stoppage has less effect on the temperature. This is not surprising as the disturbance associated with the feed stopping is, in fact, the drop in the heat released. The manipulated variable, $T_{jin,sp}$, appears only slightly different from the PID however this is due to the scale of the plot.

4.4 PERFORMANCE OF THE COMBINED FEEDFORWARD/FEEDBACK CONTROLLER

Considering the heat released and feed stoppage individually results in improved control in the areas targeted. Combining both feedforward algorithms with the PID controller should result in a far superior regulation of temperature. Therefore, during the feed, feedforward action counters the heat released. After the feed, feedforward is used to reduce the observed drop in temperature.

Compensating for the heat released reduces the temperature deviation due to feed stoppage. The dual feedforward algorithm (compensation for both disturbances) is given below. When the feed is 'on', equation (4.28) gives the required feedforward. After the feed is turned off, equation (4.29) is used to calculate the feedforward compensation.

$$T_{jin,sp,ff} = -0.025\hat{Q}_r \tag{4.28}$$

$$T_{jin,sp,ff} = -18\Delta \dot{m}_m \tag{4.29}$$

The performance of the combined feedback/dual feedforward controller is shown in Figure 4.8 for a summer, batch one simulation. During the entire batch, the temperature is maintained within ±1 °F.

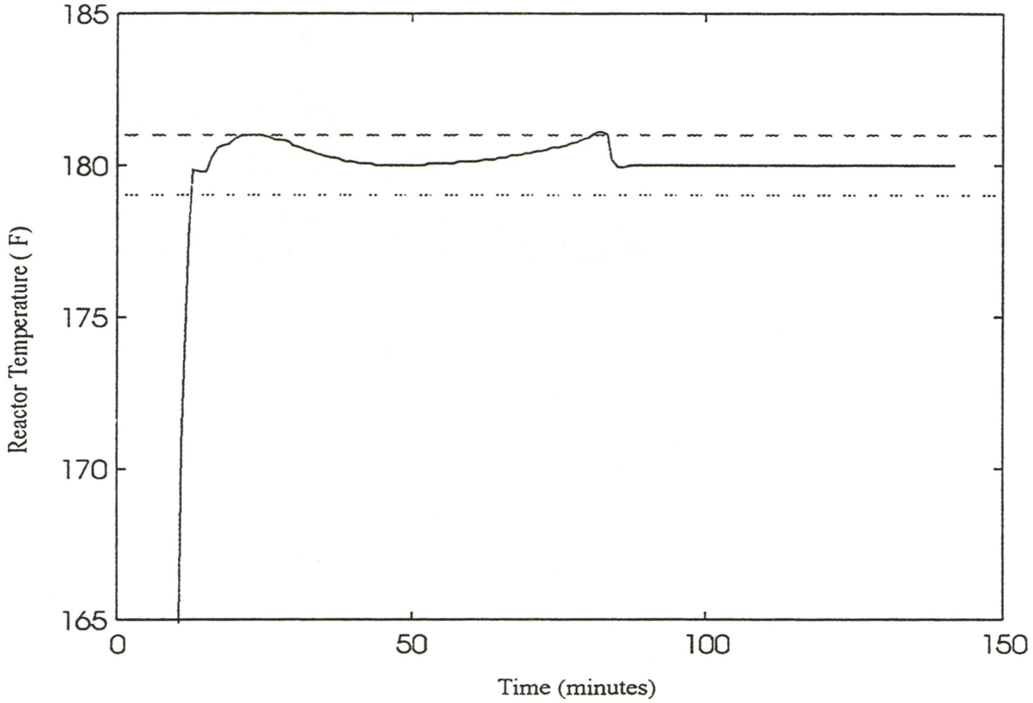


FIGURE 4.8: FEEDFORWARD/FEEDBACK CONTROL: COMPENSATION FOR BOTH DISTURBANCES

To this point, this is the only algorithm that has successfully achieved 'good' control (± 1 °F for this system). For all future chapters, the "PID with Feedforward Compensation" refers to the controller of Figure 4.8 (compensation for both disturbances). In Chapter 7, this controller will be compared to other controllers with more sophisticated process knowledge built in.

CHAPTER 5
NONLINEAR CONTROL

5.1 INTRODUCTION

Nonlinear control is frequently applied to free radical polymerization reactors. However, in many successful applications, very complex models of the process are used. In this chapter, a geometric nonlinear controller based on general energy balances is derived for the semi-batch reactor. No empirical models are available for parameters such as the heat transfer or viscosity. The performance of the nonlinear controller when all parameter values are exactly known is assessed for a Product One, batch one simulation. The sensitivity of the controller to errors in some of the parameter values is also studied.

5.2 NONLINEAR CONTROL THEORY

5.2.1 LINEARIZING CONTROLLER

There are many different approaches to nonlinear control. In this thesis, the application of a nonlinear controller derived from differential geometric control theory is considered. The theory of the linearizing controller is presented in this section.

Within the geometric nonlinear control area, several algorithms have been derived. Some of the more common ones found in chemical engineering literature are Nonlinear Decoupling Control (Balchen), Generic Model Control (Lee and Sullivan) and Globally Linearizing Control (Kravaris). While the names are different, the algorithms primarily differ in how the researcher chooses to approach the control problem and the assumptions built into the algorithm. In this work, the linearizing controller is derived in an error trajectory framework (McAuley and McGregor (1993), McLellan et al. (1990)). Parallels will be drawn to the well known Globally Linearizing Control (Soroush and Kravaris, 1992) to provide insight. Note that 'linearizing controller' and 'nonlinear controller' will be used interchangeably during this Chapter.

The general form of the process considered in nonlinear control is given by equation (5.1):

$$\frac{d\underline{x}}{dt} = \underline{f}(\underline{x}) + \underline{g}(\underline{x})u$$
$$y = h(\underline{x}) \tag{5.1}$$

where y is the controlled variable. Model equations of the above form usually arise from mass and energy balances around a reactor. It should be noted that the input, u, appears explicitly in the expression for the time derivative of the states. This is a convenient form as it results in an explicit expression for the manipulated variable. However, it is not a necessary condition to derive the nonlinear controller.

The concepts are discussed for a single input, single output system. The ideas can be easily extended to MIMO processes. In order to keep the equations compact, the dot notation is used for time derivatives in this Chapter.

Two important concepts in nonlinear control are relative order and Lie derivatives. The relative order of an input-output pair, r, is the number of times that the input must be integrated to affect the output. For a relative order one process, the first time derivative of the output is a function of the input (equation (5.2)). In a process with relative order two, the second time derivative of the output is a function of the input, however the first time derivative is not (equation (5.3)):

$$r = 1 \qquad \dot{y} = F(x,u) \tag{5.2}$$

$$r = 2 \qquad \dot{y} = F(x); \qquad \ddot{y} = F(x,u) \tag{5.3}$$

A second important concept is that of Lie derivatives. The use of Lie derivatives significantly simplifies the derivation of controllers for high (greater than one) relative order processes. In words, the Lie derivative is the directional derivative of h(x), and is an algebraic function of the states. Given a vector function f(x) and a scalar function h(x), the Lie derivative of h(x) in the direction of f(x) is:

$$L_f h(\underline{x}) = \sum_i f_i(\underline{x}) \frac{\partial h(\underline{x})}{\partial x_i} \tag{5.4}$$

Given a system described by equations (5.1), the following relationships are particularly helpful:

r = 1 system:

$$\dot{y} = L_f h(\underline{x}) + L_g h(\underline{x}) u \tag{5.5}$$

r = 2 system:

$$\dot{y} = L_f h(\underline{x}) \tag{5.6}$$

$$\ddot{y} = L^2{}_f h(\underline{x}) + L_g L_f h(\underline{x}) u \tag{5.7}$$

The nonlinear controller is discussed for a relative order two system. It is assumed that a model of the process in the form of equation (5.1) is available.

The controller is derived to achieve a desired closed loop behavior, as specified by the user. The desired closed loop behavior is conveniently expressed in an error trajectory equation. The order of the equation is set by the relative order of the process. Of course, one can modify the error trajectory by adding or removing terms. The interpretation of this is discussed later in the section. For a relative order two process, the appropriate error equation is:

$$\ddot{e} + d_1\dot{e} + d_2 e = 0 \tag{5.8}$$

where:

$$e = y_{sp} - y$$

The inclusion of $\ddot{e}$ in equation (5.8) is required for a relative order two process. The parameters d₁ and d₂ are the tuning parameters in linearizing control algorithm and are chosen by the user to provide the desired closed loop behavior. Equations (5.6) and (5.7) provide expressions for the time derivatives of the output. Substituting these into the error trajectory of equation (5.8):

$$\ddot{y}_{sp} - (L^2{}_f h(\underline{x}) + L_g L_f h(\underline{x})u) + d_1(\dot{y}_{sp} - L_f h(\underline{x})) + d_2(y_{sp} - y) = 0 \tag{5.9}$$

Equation (5.9) may be rearranged for the manipulated variable (input):

$$u = \frac{\ddot{y}_{sp} + d_1(\dot{y}_{sp} - L_f h(\underline{x})) + d_2(y_{sp} - y) - L_f{}^2 h(\underline{x})}{L_g L_f h(x)} \tag{5.10}$$

A block diagram of the linearizing controller is shown in Figure 5.1.

If a perfect model is used in the controller (equation (5.10)), the process will exhibit the specified closed loop behavior (as defined by equation (5.8)). Process-model mismatch causes a deviation from the specified error trajectory.

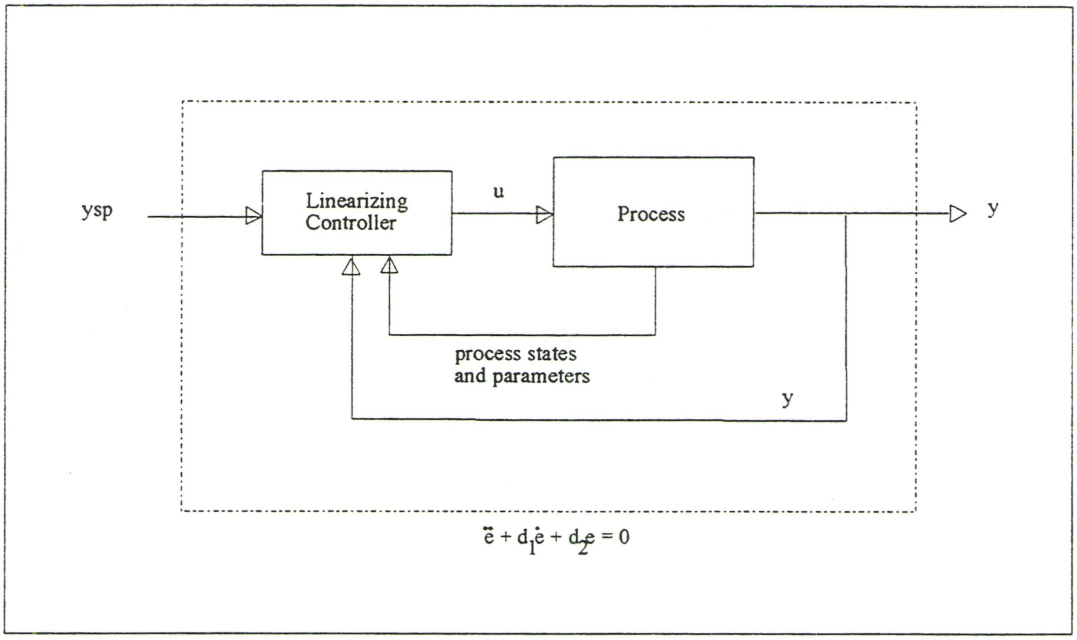

ysp
Linearizing Controller
u
Process
y
process states and parameters
y
$\ddot{e} + d_1\dot{e} + d_2e = 0$

FIGURE 5.1: BLOCK DIAGRAM OF THE LINEARIZING CONTROL ALGORITHM

In order to better understand the performance of the above nonlinear controller, it is helpful to compare it to another common formulation of linearizing control, such as Globally Linearizing Control. A brief overview of the GLC derivation is presented here. The controllers will then be compared in two aspects:

1/ closed loop behavior;

2/ manipulated variable expression;

Globally Linearizing Control is most easily explained in conjunction with a diagram. Figure 5.2 shows the block diagram of the GLC algorithm.

With the GLC approach to nonlinear control, a linearizing input, v, is defined such that the following linear relationship holds for a relative order two process:

$$\beta_1\ddot{y} + \beta_0\dot{y} + y = v \tag{5.11}$$

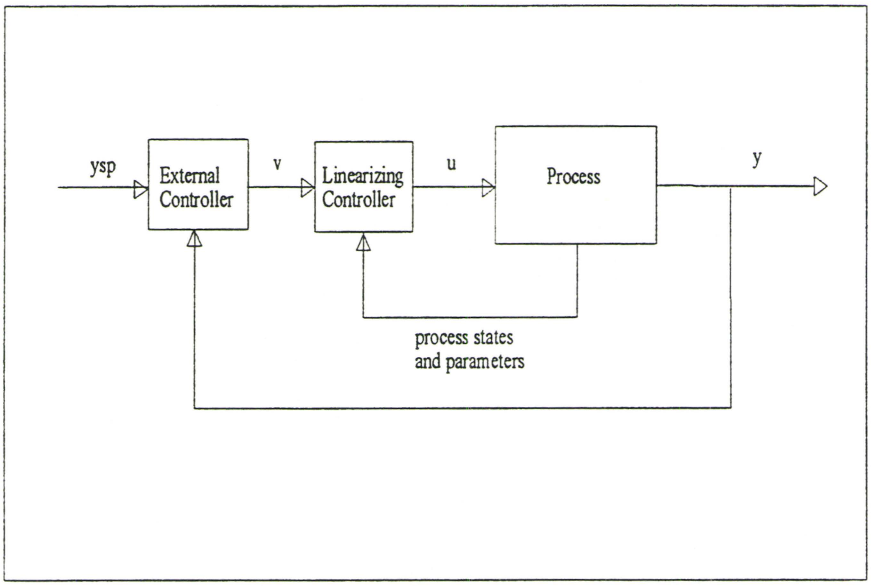


FIGURE 5.2: BLOCK DIAGRAM OF GLOBALLY LINEARIZING CONTROL

The betas are constants, selected by the user. The linearizing input, v, has no meaning in the real plant. In order to implement the GLC algorithm, an expression for the true input, u, must be calculated. Once again, equations (5.6) and (5.7) prove useful for substitutions:

$$\beta_1(L_f{}^2h(\underline{x}) + L_f L_g h(\underline{x})u) + \beta_0(L_f h(\underline{x})) + y = v \tag{5.12}$$

Rearranging for the input:

$$u = \frac{v - \beta_1 L_f{}^2 h(\underline{x}) - \beta_0 L_f h(\underline{x}) - y}{\beta_1 L_f{}^2 h(\underline{x})} \tag{5.13}$$

Equation (5.13) provides an expression for the manipulated variable in terms of the output, the model and the linearizing input. An expression for v is still required to implement GLC.

Recall the linear system between y and v. Consider controlling this 'process' with a PI algorithm, thus providing output feedback for the GLC loop. The linear controller is referred to as the 'external controller'. A PI is chosen as the external controller for convenience, however there is nothing restricting the user to this algorithm. The expression for the output of a PI controller is:

$$v = v_{bias} + K_c(y_{sp} - y) + \frac{K_c}{\tau_I}\int(y_{sp} - y)d\tau \tag{5.14}$$

Tuning parameters for the PI controller are chosen based on the behavior of the linear system between y and v (which is specified by the user). Note that the bias is usually some function of the setpoint, as decided by the user.

Substituting equation (5.14) into equation (5.13):

$$u = \frac{v_{bias} + K_c(y_{sp} - y) + \frac{K_c}{\tau_I}\int(y_{sp} - y)d\tau - \beta_1 L_f{}^2 h(\underline{x}) - \beta_0 L_f h(\underline{x}) - y}{\beta_1 L_f{}^2 h(\underline{x})} \tag{5.15}$$

Equation (5.15) is the expression for the manipulated variable for the GLC algorithm. Consider specifying a time varying bias in the PI controller:

$$v_{bias} = \beta_1 \ddot{y}_{sp} + \beta_0 \dot{y}_{sp} + y_{sp} \tag{5.16}$$

Substituting (5.16) into (5.15), and rearranging:

$$u = \frac{\beta_1 \ddot{y}_{sp} + \beta_0(\dot{y}_{sp} - L_f h(\underline{x})) + (K_c + 1)e + \frac{K_c}{\tau_I}\int (y_{sp} - y)d\tau - \beta_1 L_f{}^2 h(\underline{x})}{\beta_1 L_f h(\underline{x})} \tag{5.17}$$

If a perfect model is used in the nonlinear controller, implementing equation (5.17) as the manipulated variable results in the following closed loop behavior:

$$\beta_1 \ddot{e} + \beta_0 \dot{e} + (1 + K_c)e + \frac{K_c}{\tau_I}\int e d\tau = 0 \tag{5.18}$$

Comparing equations (5.17) and (5.18) of GLC to equations (5.10) and (5.8), respectively, from the error trajectory derivation, the following items are noted:
1/ The linearizing controller in the form of equation (5.8) has no integral action;
2/ The linearizing controller derived from an error trajectory, above, is equivalent to the GLC algorithm with a time varying bias and a Proportional only external controller;
3/ The GLC algorithm requires three variables to be specified for a relative order two process with a P-only external controller (β_1, β_0 and K_c). The error trajectory linearizing controller requires only two (d_1 and d_2) for the same algorithm. Therefore, Globally Linearizing Control has an extra (unnecessary) tuning factor.

Integral action can easily be incorporated into the linearizing controller. The error trajectory equation is modified to include an integral term:

$$\ddot{e} + d_1\dot{e} + d_2 e + d_3\int e d\tau = 0 \tag{5.19}$$

Given the specified trajectory of equation (5.19), the nonlinear controller is now equivalent to the GLC algorithm implemented with a time varying bias and a PI external controller. Once again note that the linearizing controller derived from an error trajectory specification has one less tuning factor than the equivalent GLC algorithm.

In summary, the derivation of a linearizing controller for an relative order two process has been shown. The algorithm has been compared to the well known Globally Linearizing Control. While the two formulations are shown to be equivalent in structure, each specify the controller tuning in different ways. Because of the error trajectory approach, the tuning parameters of the linearizing controller directly relate to the closed loop poles of the system. Therefore, this controller is more easily tuned and implemented. In the following section, the error trajectory-based controller is presented for the Chylla Haase semi-batch reactor.

5.2.2 Derivation of a Linearizing Controller for Chylla Haase Reactor

In this section, the specific nonlinear controller for the Chylla Haase system is derived. Proportional only control is considered first (error equation (5.8)).

Recall that temperature is being controlled by manipulating the setpoint of the inlet jacket temperature. To derive the nonlinear controller, it is assumed that the slave dynamics are very fast. Therefore, the actual inlet jacket temperature is assumed to be equal to the setpoint. This assumption significantly simplifies the controller. During the batch, there are two situations where this assumption is not strictly true:

1/ Heat-up phase. While heating the contents, the inlet jacket setpoint is at its upper limit. The actual inlet jacket temperature is much slower to respond. However, the violation of the assumption during heat-up is not critical since the controller is requesting full heating (that is, there really is no control here). Feedback of the actual reactor temperature insures that full heating is not backed off too early.

2/ Quickly changing jacket inlet setpoint. When the master controller calls for sudden changes in the setpoint, the actual jacket inlet temperature lags slightly behind. It will be shown in section 5.3.1 that this affects the nonlinear controller performance primarily when the feed stops. However, it will also be shown that several other factors are affecting the controller at that point. The process/model mismatch introduced by the assumption of no jacket dynamics plays only a small role.

The process equations that form the backbone of the nonlinear controller arise from energy balances around the reactor and the jacket contents:

$$\frac{dT}{dt}=\frac{1}{\sum(m_iC_{pi})}\left[\dot{m}_mC_{pm}(T_{feed}-T)+\mu^{0.4}k_0n_me^{-\alpha/T}(-\Delta H_r)-UA(T-T_{jout})-UA_{loss}(T-T_{amb})\right] \tag{5.20}$$

$$\frac{dT_{jout}}{dt}=\frac{1}{m_cC_{pc}}\left(\dot{m}_cC_{pc}(T_{jin}-T_{jout})+UA(T-T_{jout})\right) \tag{5.21}$$

Knowledge of the form of the kinetics and of the exact parameter values is currently assumed. It is also assumed that the jacket is well mixed. Therefore, the jacket temperature may be represented by the outlet jacket temperature. In terms of matrix notation:

$$\begin{bmatrix}\dot{T}\\ \dot{T}_{jout}\end{bmatrix}=\begin{bmatrix}f_1(T,T_{jout})\\ f_2(T,T_{jout})\end{bmatrix}+\begin{bmatrix}g_1\\ g_2\end{bmatrix}T_{jin} \tag{5.22}$$

where

$$f_1(T, T_{jout}) = \frac{1}{\sum(m_i C_{pi})}\Big(\dot{m}_m C_{pm}(T_{feed} - T) + \mu^{0.4} k_0 n_m e^{-\alpha/T}(-\Delta H_r)$$

$$- UA(T - T_{jout}) - UA_{loss}(T - T_{amb})\Big) \tag{5.23}$$

$$f_2(T, T_{jout}) = \frac{1}{m_c C_{pc}}\Big(-\dot{m}_c C_{pc} T_{jout} + UA(T - T_{jout})\Big) \tag{5.24}$$

$$g_1 = 0 \tag{5.25}$$

$$g_2 = {}^{\dot{m}_c}\!\big/_{m_c} \tag{5.26}$$

The states, $\underline{x}$, are the reactor and jacket temperatures, respectively. The output, y, is the reactor temperature. Therefore, the function h($\underline{x}$) is T.

From equations (5.20) and (5.21), this is a relative order two system. The desired closed loop behavior for a constant setpoint is given by equation (5.27):

$$-\ddot{T} - d_1\dot{T} + d_2(T_{sp} - T) = 0 \tag{5.27}$$

The selection of d_1 and d_2 will be discussed following the controller derivation.

Expressions for the time derivatives of temperature are needed. The first time derivative of temperature is given by model equation (5.20):

$$\dot{T} = L_f h(\underline{x}) = \frac{1}{\sum(m_i C_{pi})}\Big(\dot{m}_m C_{pm}(T_{feed} - T) + \mu^{0.4} k_0 n_m e^{-\alpha/T}(-\Delta H_r)$$

$$- UA(T - T_{jout}) - UA_{loss}(T - T_{amb})\Big) \tag{5.28}$$

Using the Lie derivative notation, an expression for d^2T/dt^2 may be derived. To begin, note that:

$$\begin{aligned}\ddot{T} &= L_f{}^2 h(\underline{x}) + L_g L_f h(\underline{x}) \\ &= L_f\left(L_f h(\underline{x})\right) + L_g\left(L_f h(\underline{x})\right)\end{aligned} \tag{5.29}$$

Using the definition of Lie derivatives (equation (5.4)):

$$\begin{aligned}L_f(L_f h(x)) &= L_f(f_1(T, T_{jout})) \\ &= \frac{\partial(f_1)}{\partial T}(f_1) + \frac{\partial(f_1)}{\partial T_{jout}}(f_2)\end{aligned} \tag{5.30}$$

and

$$\begin{aligned}L_g L_f h(x) &= L_g(f_1(T, T_{jout})) \\ &= \frac{\partial(f_1)}{\partial T}(g_1) + \frac{\partial(f_1)}{\partial T_{jout}}(g_2) \\ &= \frac{\partial(f_1)}{\partial T_{jout}}(g_2)\end{aligned} \tag{5.31}$$

Expressions for the partial derivative of f_1 with respect to the temperatures are required. Recall from the discussion in section 3.2.1 that the heat transfer coefficient and the viscosity are very weak functions of temperature. Therefore, the following is assumed:

$$\frac{\partial U}{\partial T} = \frac{\partial U}{\partial T_{jout}} \approx 0 \qquad \frac{\partial \mu}{\partial T} = \frac{\partial \mu}{\partial T_{jout}} \approx 0 \tag{5.32}$$

It is important to note that even if the heat transfer coefficient and viscosity are known to be strong functions of temperature, detailed models are often not available in industry. This is particularly true of multi-product systems. In such a case, the assumption of equation (5.32) may still be valid because of the tight temperature control required during the batch (deviations of less that one degree Fahrenheit).

With the assumptions of (5.32), the following expressions result:

$$\frac{\partial f_1}{\partial T} = \frac{1}{\sum(m_i C_{pi})}\left(-\dot{m}_m C_{pm} + \mu^{0.4} k_0 n_m (-\Delta H_r)\left(e^{-\alpha/T}\frac{\alpha}{T^2}\right) - UA - UA_{loss}\right) \tag{5.33}$$

$$\frac{\partial f_1}{\partial T_{jout}} = \frac{1}{\sum(m_i C_{pi})}(UA) \tag{5.34}$$

The controller is implemented at each control step as follows:

1/ Obtain measurements of temperatures and values of parameters (for now, it is assumed that the exact values are available);

2/ Estimate the partial derivatives $(\partial f_i/\partial x_i)$ from equations (5.33) and (5.34);

3/ Estimate the function values (f_i, g_2) from equations (5.23, 5.24, 5.26);

4/ Calculate $L_f^2h(x)$ and $L_gL_fh(x)$ from equations (5.30) and (5.31);

5/ Calculate the manipulated variable move required:

$$T_{jin,sp} = \frac{-d_1(L_f h(\underline{x})) + d_2(T_{sp} - T) - L_f{}^2 h(\underline{x})}{L_g L_f h(x)} \tag{5.35}$$

The value is clamped between physical limits before implementing the move in the plant.

Note that to this point, the controller is perfectly general for different products. The structure of the equations is the same for both products. However, the values of the parameters in the equations will change for each product.

In summary, a nonlinear controller for the Chylla Haase reactor has been derived. Its application is described in the next section.

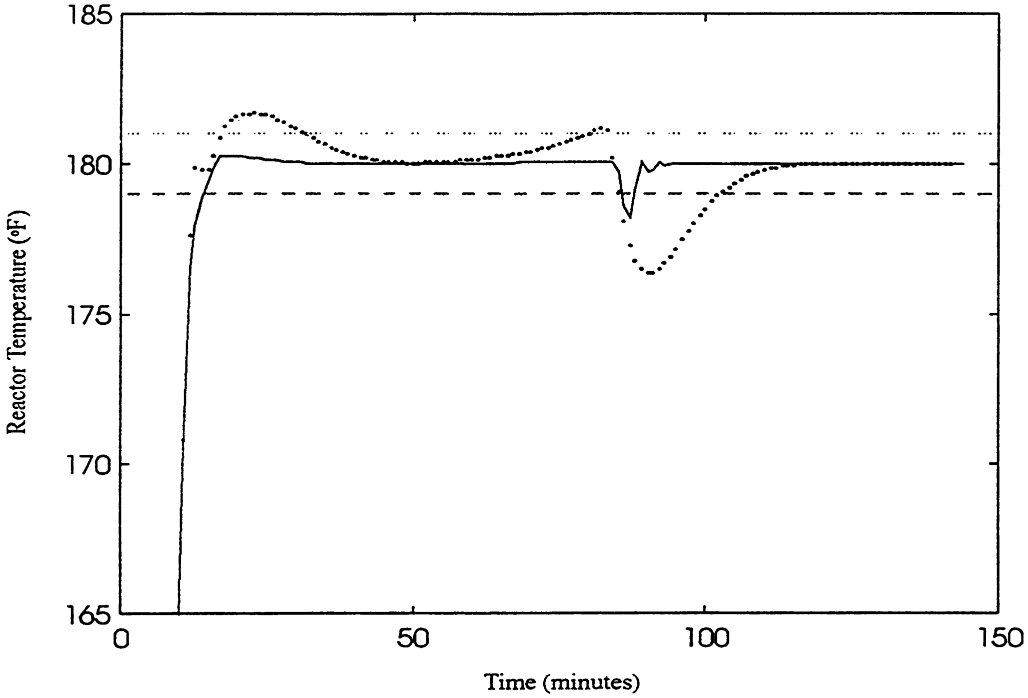
5.3 Application of Nonlinear Controller

5.3.1 Performance of the Nonlinear Controller

The purpose of this section is to evaluate the controller performance. First, the performance of the nonlinear controller is presented for the case in which all parameters are known. Proportional control on a Product One, batch one summer simulation is considered. It is seen that the controller has difficulty when the feed stops and three contributing factors are quantified. Finally, it is shown that the benefits provided by the nonlinear controller are not, in fact, due to accounting for the process nonlinearities.

The tuning parameters for the nonlinear controller are d_1 and d_2. The values of d_1 and d_2 were chosen as $d_1 = 6$ and $d_2 = 3$ by simulating the error trajectory equation in Matlab. For these values, the desired closed loop response is overdamped with a settling time of ten minutes. When implemented on the plant, these tuning parameters gave good performance for temperature control of product one.

Figure 5.3 shows the performance of the nonlinear controller (solid line) for a Product One, summer, batch one simulation. The well tuned PID (no feedforward) is shown as a reference.



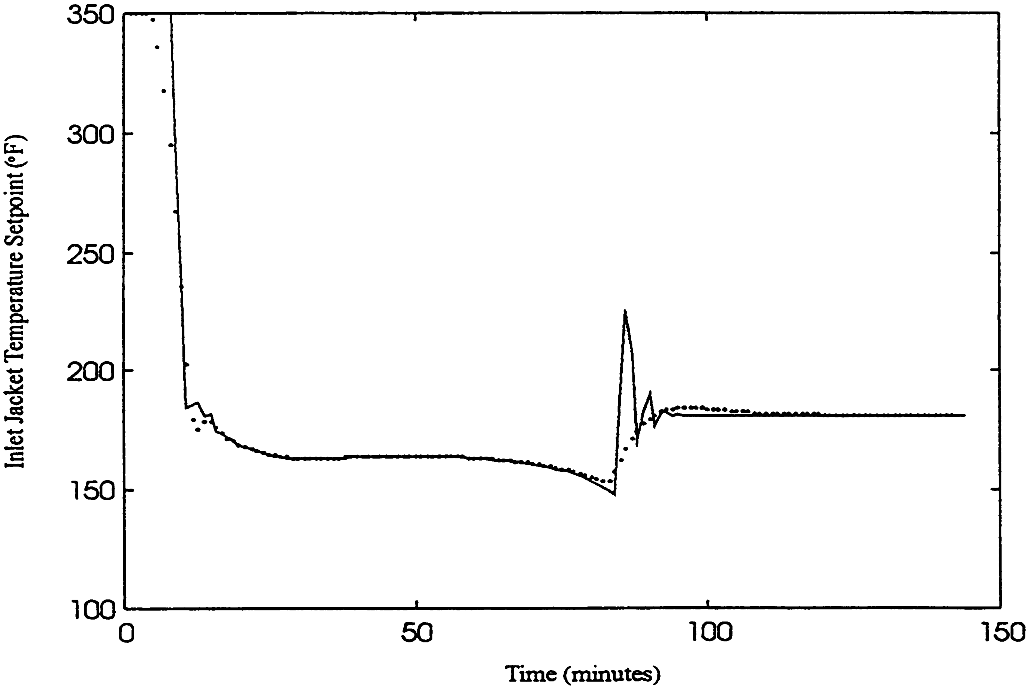


FIGURE 5.3: NONLINEAR PROPORTIONAL CONTROL USING EXACT PROCESS PARAMETER VALUES

Negligible offset is seen because a very good process model is used in the nonlinear controller. When the temperature deviates from 180 °F, the controller returns it to the setpoint along the specified closed loop trajectory.

While the nonlinear controller is outperforming the PID, the temperature drops below 179 °F after the feed stops. A short study indicates that three contributing factors are:

1/ Time that the feed stops;

2/ Execution interval of the controller;

3/ Slave dynamics;

With respect to the first point, it is noted that the feed is stopped at 84.14 minutes (exactly 70 minutes after it is started). However, the controller is unaware of the change until 85 minutes, when it makes its next control move. By that time, the reaction has ceased, no heat is being released and temperature is already falling. If the feed is stopped at 83.9 minutes, control improves significantly. This situation is shown in Figure 5.4 (the control when feed is stopped at 84.14 minutes is also shown). A small time frame is used to emphasize the differences. The temperature is maintained within one degree of setpoint when the feed is stopped at 83.9 minutes.

It should be emphasized that it is unlikely that the feed will be stopped after exactly 70.00 minutes. In the true processing environment, it is expected that some batches will have better temperature control when the feed stops because of this randomness. The simulation in Figure 5.3 shows the worst case scenario, where almost a full minute has passed before the controller records a feed flow measurement and knows that the feed has stopped. It should also be noted that this problem is less severe for the PID class controllers presented in Chapters four and five. Due to the different heating time required by the PIDs, the feed was stopped less than a half minute before the controllers next execution interval. Therefore, the plots shown in the previous Chapters were not actually the worst case scenario (with respect to the time that the feed stops).

The second contributing factor, the execution interval of the controller, is related to the discussion on the timing of the feed stop. When the controller is being executed more often, it senses process changes more quickly. Figure 5.5 show the performance of the nonlinear controller when it is being executed every 0.1 minutes instead of every minute (the same tuning is used). Once again, the original control (executing every minute) is shown on the figure for comparison.

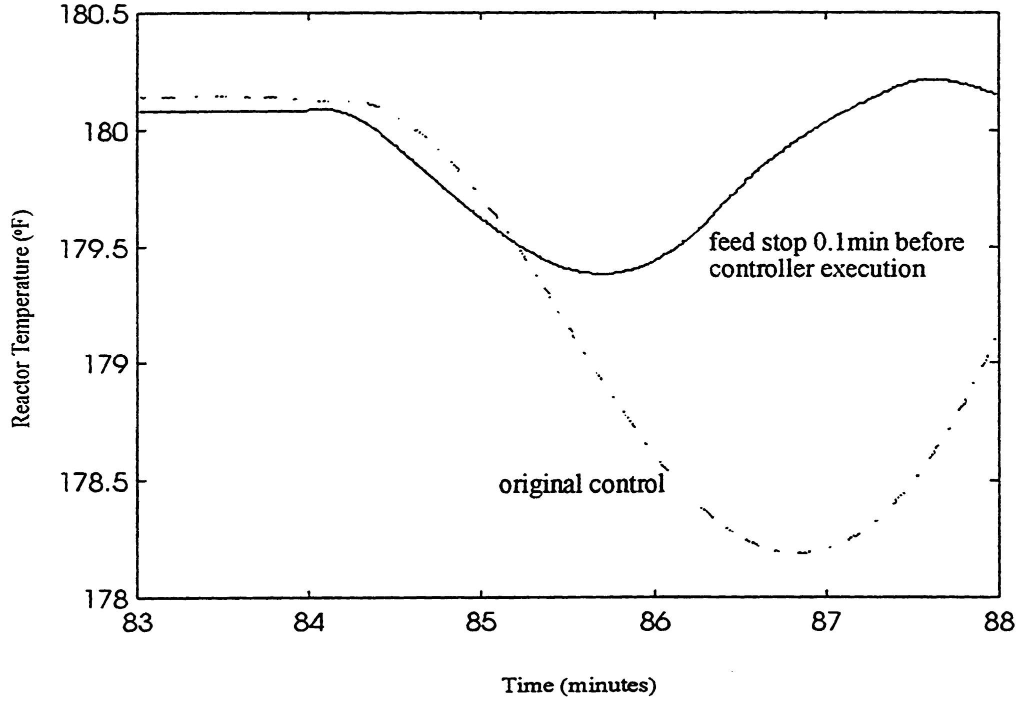


FIGURE 5.4: EFFECT OF TIME AT WHICH THE FEED STOPS

Again, an improvement in control is seen. If it is possible to change the execution interval on the controller, this is a viable method to improve temperature regulation.

Finally, it is interesting to see the impact of slave dynamics are on the nonlinear controller. To simulate this, the actual process equations are modified so that $T_{jin} = T_{jsp}$ (possible in a simulation only!). The nonlinear controller is executed every minute. Figure 5.6 shows the control under this situation.

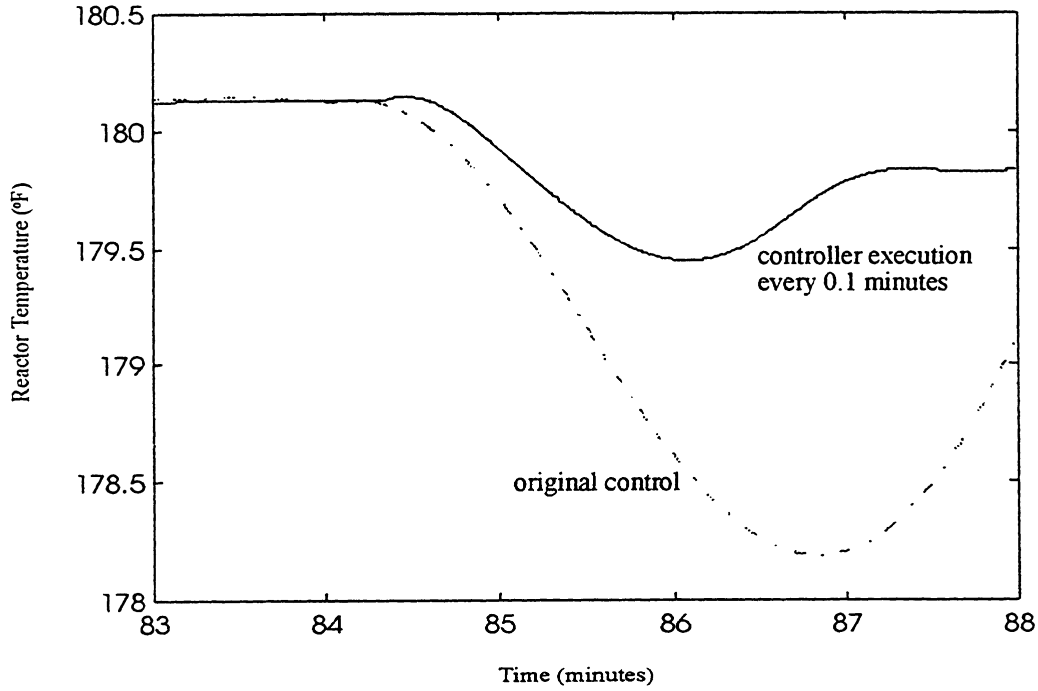


FIGURE 5.5: EFFECT OF EXECUTION INTERVAL

Figure 5.6 shows that control can be improved if there are no slave dynamics. Obviously, setting T_{jin} to T_{jsp} is not physically possible on a real plant. One physically possible option is to include a dynamic equation in the controller:

$$\tau \frac{dT_{jin}}{dt} + T_{jin} = T_{jin,sp} \tag{5.36}$$

Including this equation with the two energy balances in the nonlinear controller increases the relative order of the system to three. This was not pursued further for two reasons:

1/ Increased complexity of the controller;

2/ Larger improvements in control are possible by adjusting the time that the feed stops and changing the control interval;

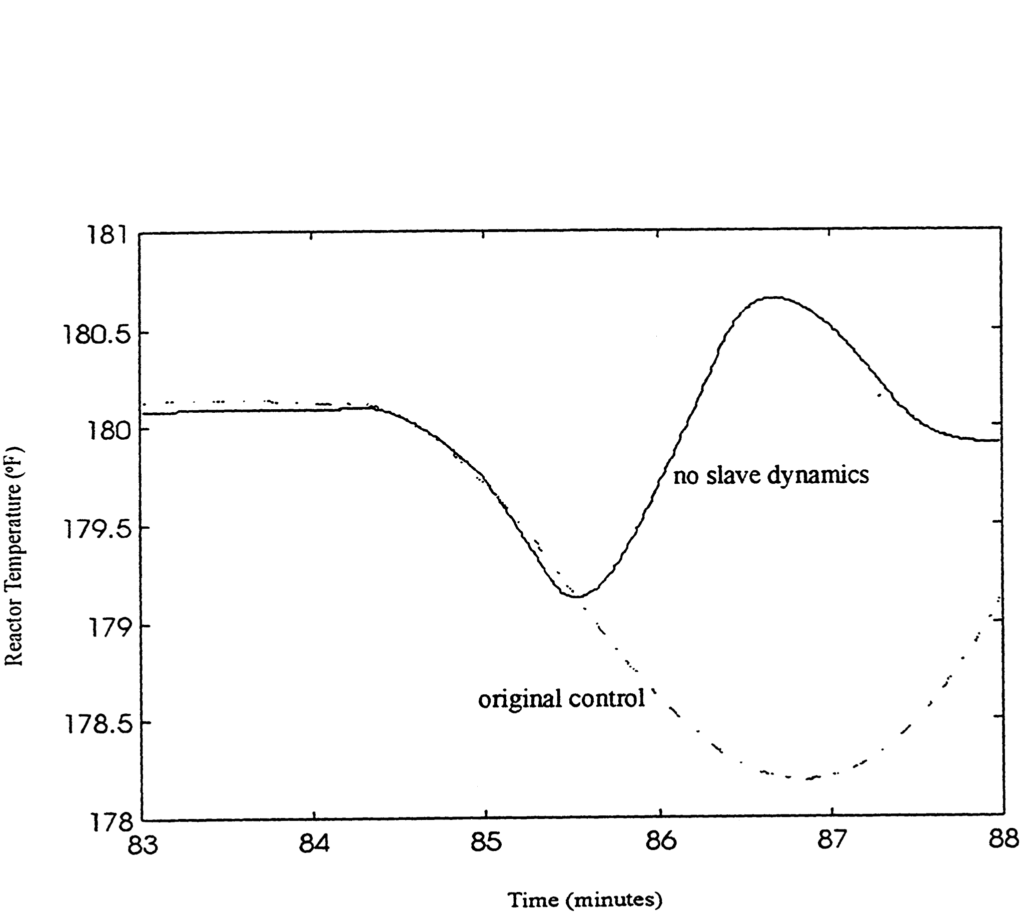


FIGURE 5.6: EFFECT OF SLAVE DYNAMICS

The slave dynamics assumption is interesting for one other reason. Often in literature, *processes are simulated* under the assumption that neglecting slave dynamics does not introduce error into the simulation. For the semi-batch Chylla Haase system, this assumption is not valid and it is questionable for how many processes it is valid. Clearly, when slave dynamics are neglected, the nonlinear controller is able to maintain the temperature within the bounds of good control. Even with fast slave dynamics (given a control interval of one minute), it cannot.

Despite the temperature drop when the feed stops, the nonlinear controller outperforms the well tuned PID over the batch. One might assume that the benefits are due compensating for nonlinearities in a highly nonlinear polymerization reactor.

Consider again the energy balance equations in the controller:

$$\frac{dT}{dt} = \frac{1}{\sum(m_i C_{pi})}\Big[\dot{m}_m C_{pm}(T_{feed} - T) + \mu^{0.4} k_0 n_m e^{-\alpha/T}(-\Delta H_r)$$

$$- UA(T - T_{jout}) - UA_{loss}(T - T_{amb})\Big] \tag{5.20}$$

$$\frac{dT_{jout}}{dt} = \frac{1}{m_c C_{pc}}\left(\dot{m}_c C_{pc}(T_{jin} - T_{jout}) + UA(T - T_{jout})\right) \tag{5.21}$$

In section 5.2.2, it was discussed that the only source of temperature nonlinearity being compensated for is the Arrhenius dependence of the rate of reaction. Many parameters are functions of other variables (fraction solids for example), but are not strongly dependent on either the reactor or jacket temperature. Figure 5.7 shows the value of the exponential term ($e^{-\alpha/T}$) during a product one simulation.

Figure 5.7 shows that, in fact, the exponential term is essentially constant once the initial heating phase is finished. That is, except between 14 and 18 minutes, the nonlinearities are not really being exercised. Therefore, any benefits seen by the nonlinear controller are probably not due to accounting for the nonlinearities in temperature. Rather, the nonlinear controller outperforms the PID because the former has more information about how the process changes with time.

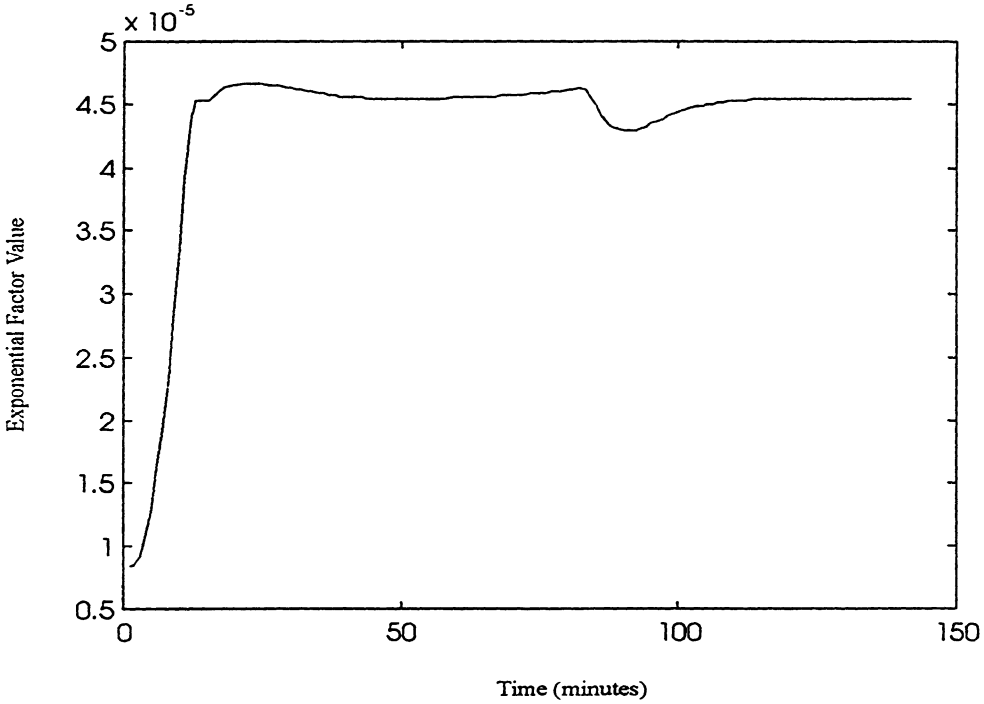


FIGURE 5.7: VALUE OF ARRHENIUS TERM IN RATE OF REACTION

In summary, the performance of the nonlinear controller is shown. An explanation of the controller behavior is provided by changing various process conditions. Finally, it is highlighted that the benefits of the nonlinear controller arise primarily from having more information about how the process is changing.

5.3.2 SENSITIVITY TO PARAMETER ERROR

The previous section showed the performance of the nonlinear controller with perfect knowledge of all parameter values. In this section, the effect of limited process information on the controller behavior is studied. Simulations without integral action are considered first.

In a true processing environment, one might expect very limited information about the heat transfer rate (UA), the auto acceleration effect (viscosity term in reaction rate) and the reactor contents heat capacity($\Sigma(m_iC_{pi})$). It was found that the nonlinear controller is most sensitive to errors in the first two parameters, therefore discussion is limited to the heat transfer and auto acceleration.

Consider first the effect on the linearizing controller performance when an average value of UA is used in the algorithm. It is assumed that all other parameters in the controller are known, or good estimates are available. Figure 5.8 shows the control when a constant value of UA = 875 Btu/(°F hr) is used (the true value ranges from ≈1400 to 500 Btu/(°F hr)). The dashed lines show the limits for good control performance.

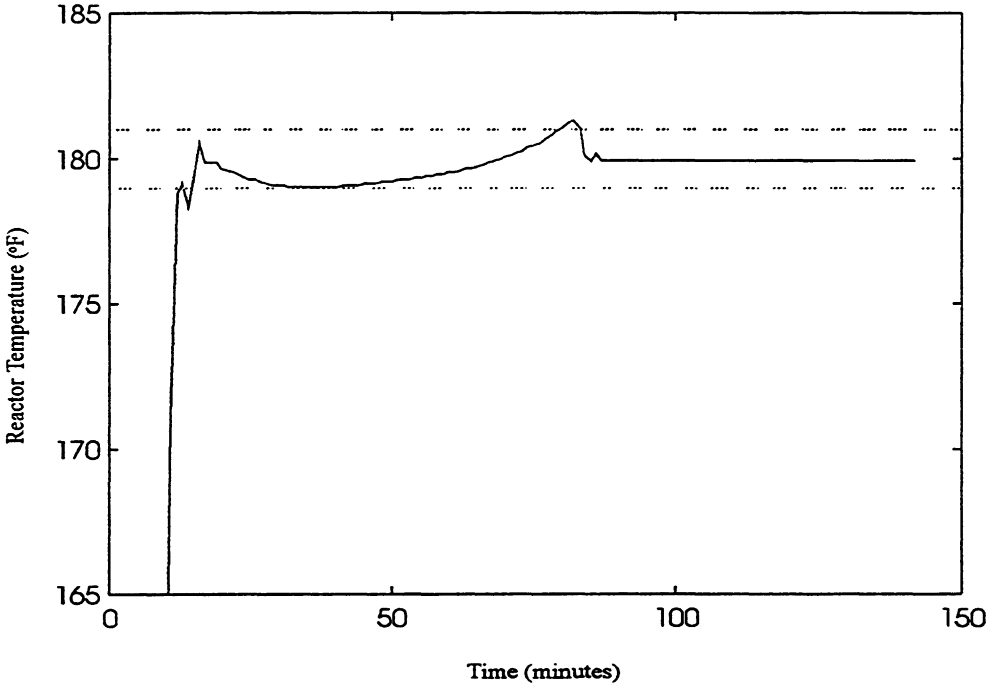


FIGURE 5.8: NONLINEAR PROPORTIONAL CONTROL USING AN AVERAGE UA VALUE

As Figure 5.8 shows, using an average value of heat transfer rate degrades controller performance. The temperature drops to 179 °F during the feed, and exceeds 181 °F at the end of the feed.

The second parameter of interest is the auto-acceleration effect. An average value of four for $\mu^{0.4}$ is used in the controller (the actual parameter varies between one and nine). Figure 5.9 shows the performance (note that UA is assumed exactly known for this plot):

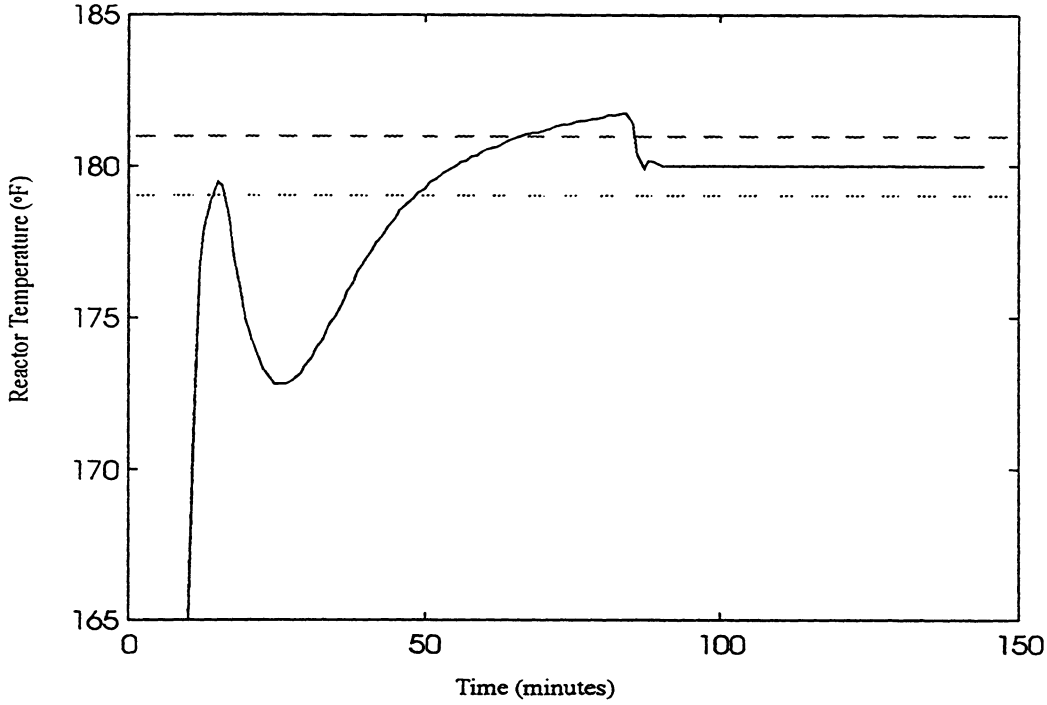


FIGURE 5.9: NONLINEAR PROPORTIONAL CONTROL USING AVERAGE AUTO ACCELERATION VALUE

Clearly the controller is very sensitive to mis-information about the auto-acceleration.

Figure 5.10 shows the temperature control when average values of both UA and the auto-acceleration effect are used in the controller. Temperature regulation has severely degraded, and is in fact worse than that of a PID controller.

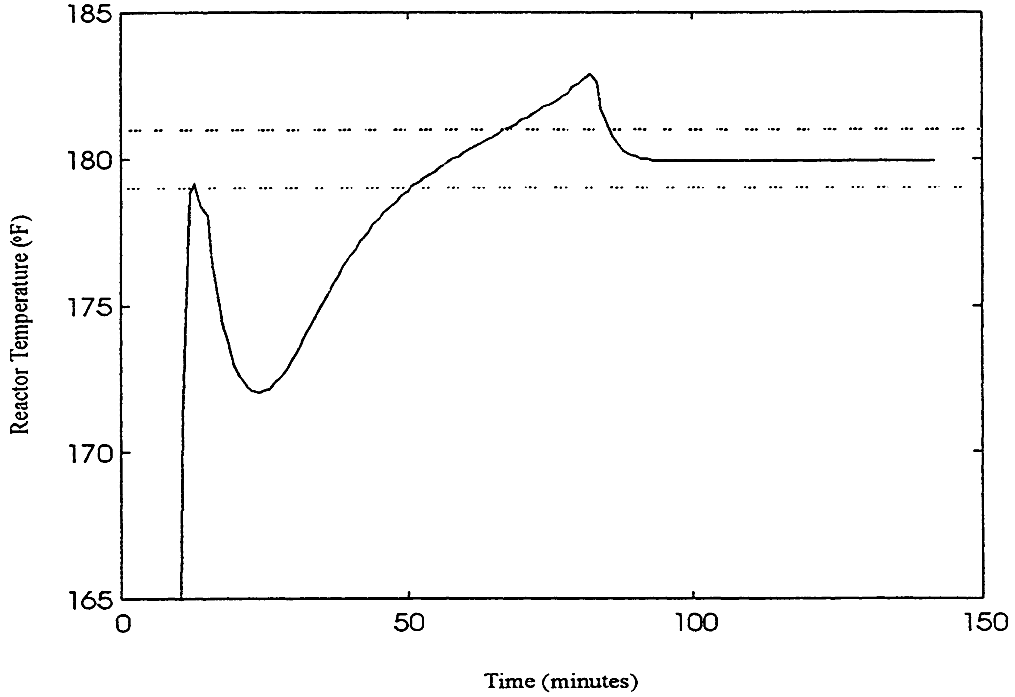


FIGURE 5.10: NONLINEAR PROPORTIONAL CONTROL USING AVERAGE HEAT TRANSFER AND AUTO-ACCELERATION VALUES

It should be noted that there is no integral action in the above simulations and the offset seen is due to process-model mismatch. To eliminate the offset, integral action can be implemented by modifying the error equation (discussed in section 5.2.1). The expression for the manipulated variable also changes to reflect the integral action:

$$\ddot{e} + d_1\dot{e} + d_2 e + d_3\int e d\tau = 0 \tag{5.37}$$

$$u = \frac{\ddot{y}_{sp} + d_1(\dot{y}_{sp} - L_f h(\underline{x})) + d_2(y_{sp} - y) + d_3\int(y_{sp} - y)d\tau - L_f^{\,2}h(\underline{x})}{L_g L_f h(x)} \tag{5.38}$$

The values for d_1 and d_2 are retained from the proportional-only nonlinear controller tuning. The best value of d_3 is found by trial and error to be 0.4. It was decided to implement the integral action when the feed starts; having integral action during heatup causes heating to be reduced too early and lengthens the time of the batch. Windup is avoided by clamping the value of the integral term between limits:

$$U\,\mathrm{int}_{\min} \le d_3 \int (T_{sp} - T)d\tau \le U\,\mathrm{int}_{\max} \tag{5.39}$$

The limits are calculated at each execution interval with equation (5.40):

$$\begin{aligned} U\,\mathrm{int}_{\min} &= T_{jin}(\min)L_g L_f h(x) + d_1(L_f h(\underline{x})) - d_2(T_{sp} - T) + L_f{}^2 h(\underline{x}) \\ U\,\mathrm{int}_{\max} &= T_{jin}(\max)L_g L_f h(x) + d_1(L_f h(\underline{x})) - d_2(T_{sp} - T) + L_f{}^2 h(\underline{x}) \end{aligned} \tag{5.40}$$

Note that equation (5.40) is simply equation (5.38) rearranged and written for the Chylla Haase reactor. If the integral term exceeds the calculated limits, it is set to the respective limit value. Figure 5.11 is the controller performance for product one, batch one when average values of UA and the auto-acceleration effect are used and integral action is implemented.

As Figure 5.11 shows, the integral action is unable to compensate for the process-model mismatch. The temperature is deviating significantly from setpoint and the control is severely degraded from the case where all parameter values are known.

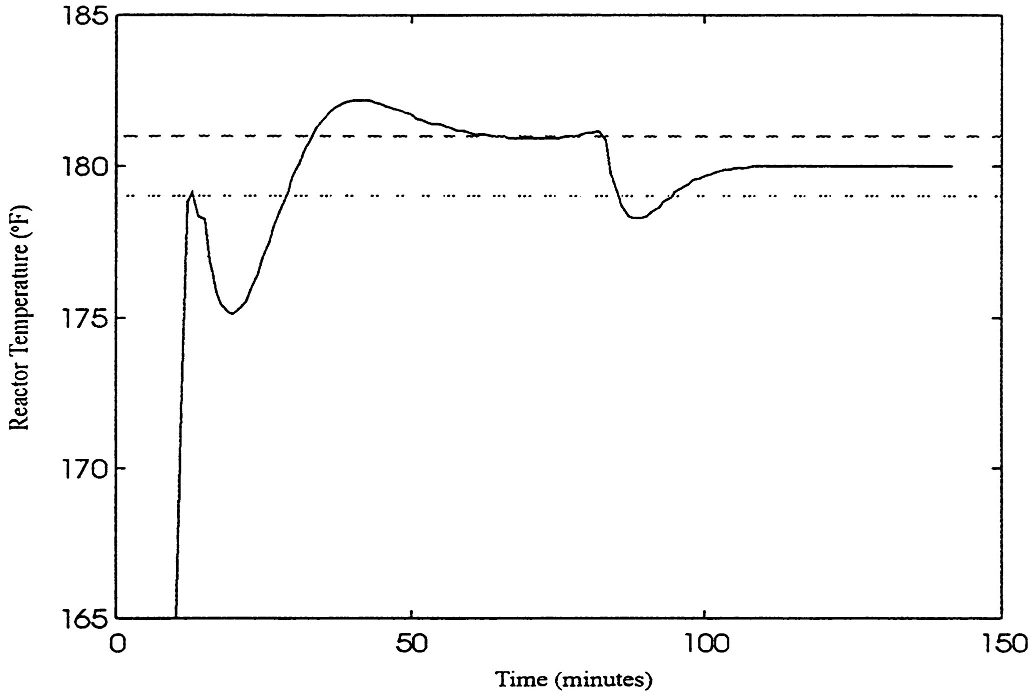


FIGURE 5.11: NONLINEAR PROPORTIONAL INTEGRAL CONTROL USING AVERAGE HEAT TRANSFER AND AUTO-ACCELERATION VALUES

In summary, the performance of the nonlinear controller is presented for the case where all parameter values are known. It easily outperforms the well tuned PID. However, temperature control degrades substantially when only average values of select parameters are available. Integral action is unable to compensate sufficiently for the process/model mismatch.

CHAPTER 6
NONLINEAR ADAPTIVE CONTROL

6.1 INTRODUCTION

The results of Chapter 5 highlight the need for good, on-line parameter estimates. An extended Kalman filter is applied in this chapter to provide the required estimates. In the first section, the theory of Kalman filtering is reviewed. Then, the details of the parameter updating are discussed, and the performance of the estimator is presented. Note that the estimation is only discussed in the context of Product One. The extension to Product Two is left as a discussion in Chapter 7. Finally, the extended Kalman filter is coupled with the nonlinear controller of Chapter 5, and the temperature control of the nonlinear adaptive controller is assessed. It is shown that the control provided by the combined estimation/control scheme is comparable to that of the nonlinear controller with all parameter values known for a Product One, batch one simulation. Robustness of the nonlinear adaptive controller to expected errors is presented at the end of this Chapter.

6.2 EXTENDED KALMAN FILTER

In this section, a brief review of the theory of Kalman Filtering is presented (Gagnon and MacGregor (1991), MacGregor (1986)).

To apply the Kalman Filter, it is assumed that a state space model of the process is available:

$$\frac{d\underline{x}^d}{dt} = \underline{f}(\underline{x_d}, u) \tag{6.1}$$

$$y = h(x^d, u) + v \tag{6.2}$$

where x_d is a vector of deterministic model states (for example, reactor temperature and jacket temperature). The measurements, y, are some function *h* of the states of the system. The measurement error (from sensors) is represented by v. Since the state equations are identical to those required by the nonlinear controller, a minimal amount of process modeling is required to apply the Kalman filter in conjunction with the controller from Chapter 5. In the extended Kalman filter, the unknown parameters are represented as stochastic states. Therefore, process equation (6.1) is augmented with equation (6.3):

$$\frac{d\underline{x}^s}{dt} = \underline{\alpha} \tag{6.3}$$

In the absence of any knowledge about the parameters, random walk behavior is generally assumed. In such a case, alpha is a white noise vector.

The Kalman filter may be visualized as two main parts. First, the process model (equations (6.1) and (6.3)) is integrated from time t_k (last execution time) to time t_{k+1} (current time), thus providing estimates of the deterministic and stochastic states at t_{k+1} Second, the Kalman filter updates the state estimates (predicted by the model) based on the on-line measurements. In the simulations, the extended Kalman filter is implemented every minute, and provides updated parameter values to the nonlinear controller. Five steps are followed at each execution:

1/ Integrate the process equations from t_k to t_{k+1}:

$$\underline{x}_{k+1|k} = \underline{x}_{k|k} + \int_{k}^{k+1} f(x,u)d\tau \tag{6.4}$$

where $\underline{x}$ includes deterministic and stochastic states. The Runge-Kutta numerical integration algorithm is used.

2/ Linearize the model equations about the predicted state values. Taylor series expansion is used to linearize the equations. Subsequent discretization results in equation (6.5):

$$\underline{x}_{k+1} = A_k \underline{x}_k + B_k u_k + \underline{w}_k \tag{6.5}$$

where x_{k+1} includes deterministic and stochastic states. w_k is a white noise vector with covariance matrix R_w. The matrix R_w is generally assumed to be diagonal, and its elements (particularly those associated with the stochastic states) are the main tuning factors of the extended Kalman filter. A comment about tuning is included below.

3/ If necessary, linearize the measurement equation (6.2);

$$\underline{y} = H_k \underline{x}_k + \underline{v}_k \tag{6.6}$$

v_k is a white noise vector with a covariance matrix R_v. For a given sensor, the measurement error is usually well known.

4/ Solve the Ricatti equations for the Kalman gain:

$$P_{k+1|k} = A_k P_{k|k} A_k^T + R_w$$

$$K_k = P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R_v)^{-1} \tag{6.7}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_k H_k P_{k+1|k}$$

5/ Update the state and parameter estimates when new measurements, y_{k+1}, become available. For the Chylla Haase reactor, the measurement vector is [T T_{jout}]';

$$\underline{x}_{k+1|k+1} = x_{k+1|k} + K_k (y_{k+1} - h(x_{k+1|k}, u)) \tag{6.8}$$

Before proceeding, a comment about the tuning is required. Each diagonal element in R_w is associated with a state or parameter. As indicated above, the elements in R_w associated with the stochastic states are the main tuning factors of the extended Kalman Filter. Initially, the value of an R_w element is selected to represent the amount the associated parameter is expected to change during the execution interval. Throughout the chapter, this is used as a guideline for characterizing the tuning of a given Kalman filter. The expected change in a state or parameter is roughly equal to 2σ. Therefore, an order of magnitude guideline for the value of the corresponding R_w element, σ_w^2, is:

$$\sigma_w{}^2 \approx (\exp ected\ change / 2)^2 \tag{6.9}$$

In the simulations in Chapter 6, a qualitative level (large, reasonable or small) is given to the R_w elements, based on the guideline of equation (6.9). The size of the R_w elements reflect the weight given to the measurements and that given to the model predictions (Wells, 1971). Increasing the size of the R_w values increases the Kalman gain thus placing more weight on the measurements and less on the model predictions. Very large values of σ_w^2 effectively mask any good information that is in the process model by heavily emphasizing the measurements. Therefore, it is desirable to keep the R_w elements small in order to take advantage of the process model. A second advantage of keeping the σ_w^2 values small (particularly those associated with the stochastic parameters) is to moderate the updating. Large σ_w^2 values can cause the updated parameters to change significantly at each interval. Since the estimates of the unknown parameters are used in the control algorithm, this type of behavior should be avoided for stability concerns.

In order to assess the performance of the Kalman filter, the estimated parameters are compared to the true values. Of course, in practice, one may never know how well the Kalman Filter is estimating the actual parameters. The only true measure is how the combined estimation/controller is regulating temperature. However, comparing the estimated parameters to the true values is helpful for establishing guidelines for the successful application of the Kalman filter.

Implementation issues are discussed in the next section.

6.3 Selection of Updated Parameters

The equations in the nonlinear controller contain several parameters that may not be known. However, the Kalman filter has limited degrees of freedom, and not all parameters can be updated. Recall the equations required for the nonlinear controller:

$$\begin{aligned}\frac{dT}{dt} = \frac{1}{\sum(m_i C_{pi})}\Big[&\dot{m}_m C_{pm}(T_{feed} - T) + \beta n_m e^{-\alpha/T}(-\Delta H_r) \\ &- UA(T - T_{jout}) - UA_{loss}(T - T_{amb})\Big]\end{aligned} \tag{6.11}$$

$$\frac{dT_{jout}}{dt} = \frac{1}{m_c C_{pc}}\left(\dot{m}_c C_{pc}(T_{jin} - T_{jout}) + UA(T - T_{jout})\right) \tag{6.12}$$

where $\mu^{0.4}$ and k_0 are lumped together as β and henceforth referred to as the auto-acceleration factor. To implement the nonlinear controller, values for all the process parameters are required. The Kalman filter will be considered for updating the time varying parameters whose values cannot be obtained by other means. Table 6.1

summarizes the process parameters, and the assumptions about their estimates. Note that values will be required for each polymer product. Appendix C outlines possible methods for obtaining values of the parameters assumed 'known'.

Table 6.1: Summary of Information about Process Parameters

Parameter	Assumption
$\Sigma(m_i C_{pi})$	known
$\dot{m}_m$	known
$C_{pm,}$ C_{pc}	known
T_{feed}	known
A	known
$\dot{m}_c$	known
m_c	known
$e^{-\alpha/T}$	known*
ΔH_r	known*
UA_{loss}	known
β	unknown
n_m	unknown
U	unknown
T_{amb}	known

From Table 6.1, the parameters that require estimation are the auto acceleration factor (β), the moles of monomer (n_m) and the heat transfer coefficient (U). With respect to the items marked with asterisks, if estimates of $e^{-\alpha/T}$ and ΔH_r are not available, these two parameters can also be lumped into beta and estimated by the Kalman filter.

Implementing the Kalman filter is discussed in the next section.

6.4 Parameter Estimation

6.4.1 Estimation of Auto Acceleration Factor

One of the time varying parameters to be updated is the auto acceleration factor, β. Key ideas about the extended Kalman filter (EKF) are introduced in this section by presenting the estimation of beta. Building a priori knowledge into the extended Kalman filter is the focus of this section. The auto acceleration parameter is modeled as a random walk, a ramp and an exponential. It is seen that the Kalman filter can track the auto acceleration factor most effectively when it is represented as an exponential. The section highlights the importance of incorporating process structure into the Kalman filter.

Initially when considering the rate of reaction parameters (β and n_m), both were lumped into a single variable, βn_m. The Kalman filter equations include equations (6.11) and (6.12) for the deterministic states and a random walk model for the parameter, βn_m. It is assumed that the heat transfer coefficient is exactly known for these simulations. Note as well that the estimation is only run during the feed period. The nonlinear controller is regulating temperature during the batch independent of the EKF. The true process values are used in the controller for now (combined estimation and control is considered at the end of this Chapter). Figure 6.1 shows true (solid line) and updated (dotted line) values of βn_m for a Product One, batch one, summer simulation. No noise is added to the temperature measurements until the end of the chapter.

It should be noted that the extended Kalman filter in Figure 6.1 has a very high value of σ_w^2 associated with βn_m. From the plot, the expected change in the true parameter value during one execution interval is about 18 units. Therefore $2\sigma \approx 18$ for this parameter and the corresponding R_w element should be approximately 81 (from the guideline presented in the previous section). The R_w element corresponding to βn_m is set actually set to 2000 in Figure 6.1, two orders of magnitude higher.

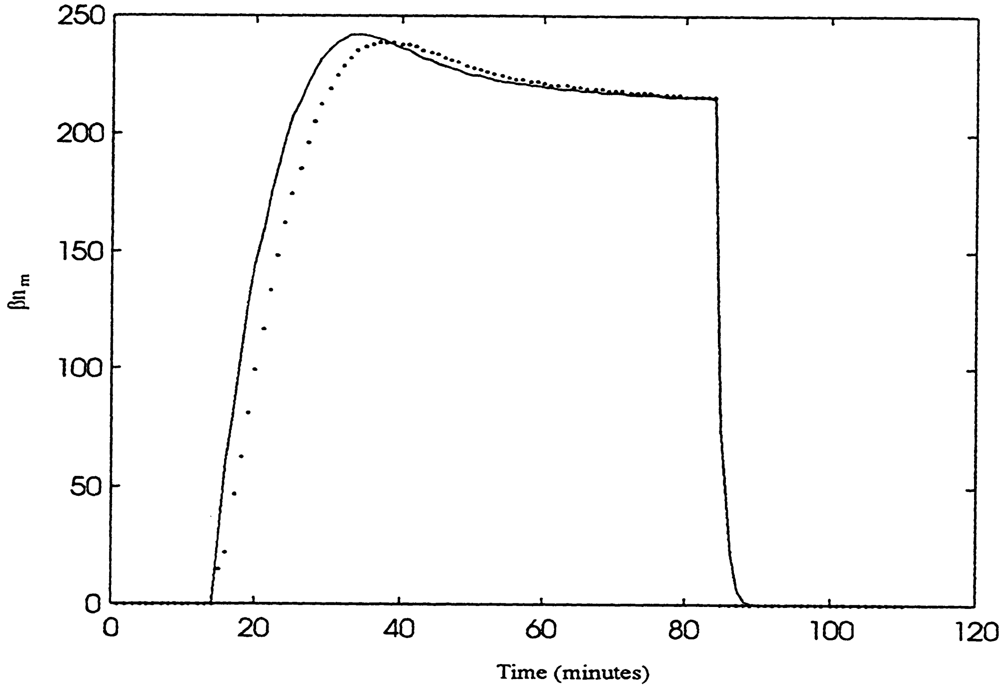


FIGURE 6.1: ESTIMATION OF PRODUCT βn_m (Rw = DIAG{2, 2, 2000})

The shape of the parameter βn_m is causing problems for the Kalman filter for two reasons. First, it is not a random walk (as it is modeled). Second, the true βn_m value changes very rapidly, making it difficult to track. This highlights the fact that the Kalman filter is ideally suited for slow, time varying parameters (if a random walk model is assumed). For these reasons, the auto acceleration and the moles of monomer will be treated separately.

By considering the auto acceleration factor alone, the problem of a fast changing parameter is less severe. However, it varies exponentially, changing slowly at the beginning of the feed and more quickly near the end. The performance of the extended Kalman filter when estimating β is shown in Figure 6.2. In the estimation, beta is scaled by 100 so that its magnitude is on the same order as the temperatures. Even with a large value of σ_w^2, the EKF has difficulty tracking β.

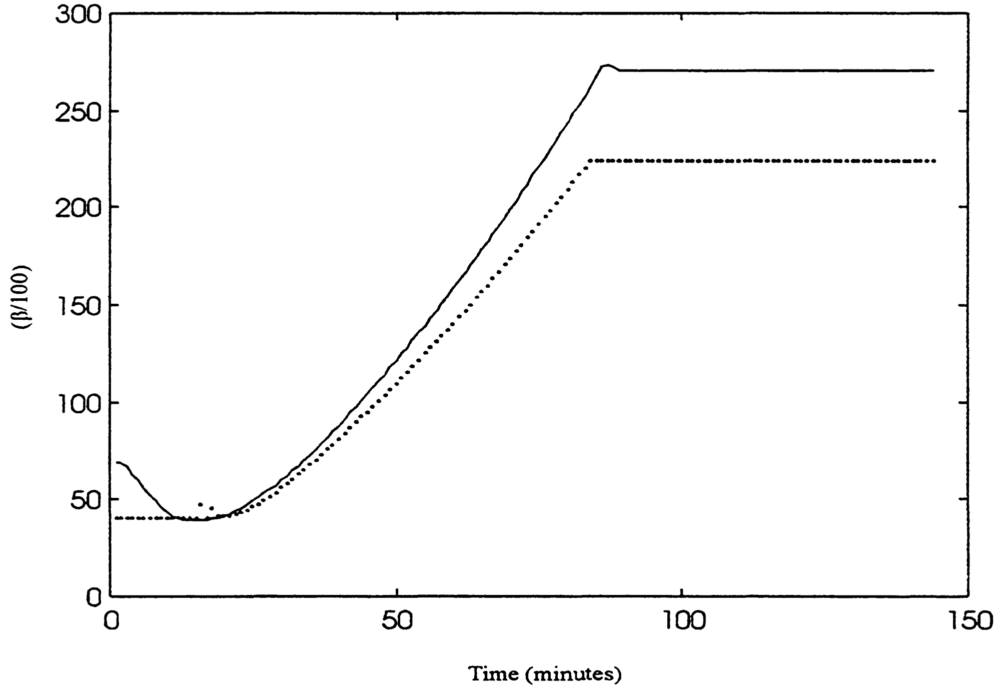


FIGURE 6.2: ESTIMATION OF AUTO ACCELERATION FACTOR MODELED AS A RANDOM WALK; RW = DIAG{2, 2, 500}

Note that the auto acceleration factor is only estimated during the monomer feed. For control purposes, this is the only time that an estimate of β is required: the moles of monomer (and hence the reaction rate) are zero otherwise and β does not appear in the nonlinear control equations. For the purpose of estimating the parameter, once the feed is stopped, the last estimate of β is retained.

Models for unknown parameters are not restricted to random walks. A second alternative is to model the auto acceleration factor as a ramp. Therefore, both the slope and the level are updated as random walk variables. The extended Kalman filter equations are:

$$\frac{d\beta}{dt} = b_0 + w_{t1}$$
$$\frac{db_0}{dt} = w_{t2} \tag{6.13}$$

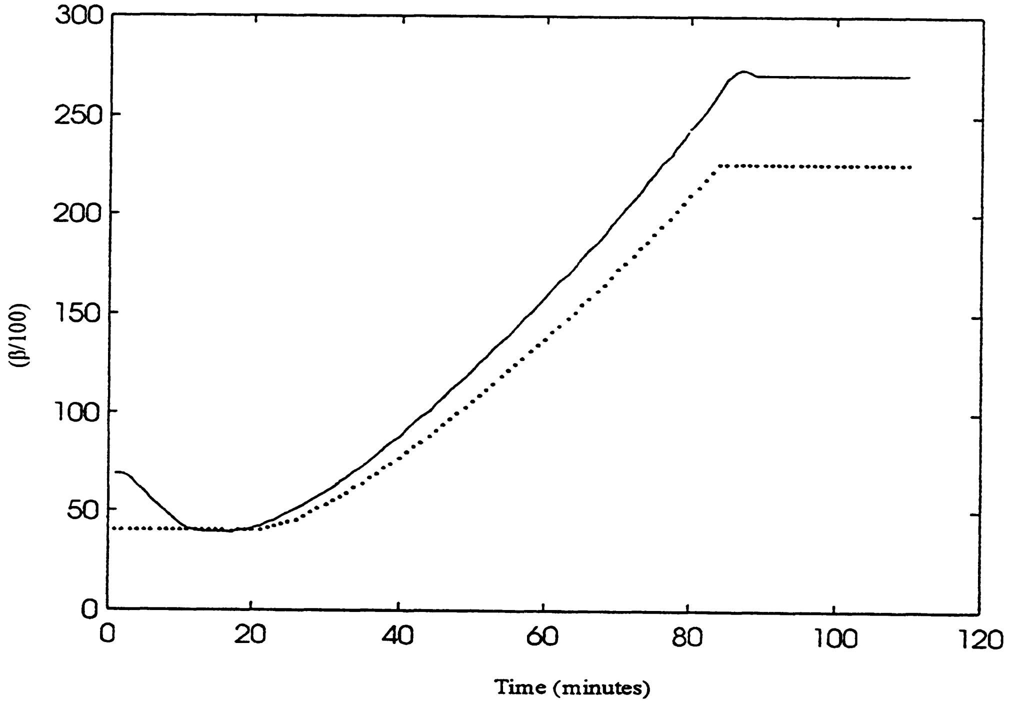


FIGURE 6.3: ESTIMATION OF AUTO ACCELERATION MODELED AS A RAMP; RW = DIAG{2, 2, 50, 0.02}

Parameter tracking when a ramp model is used is shown in Figure 6.3. Once again, the estimated auto acceleration factor is shown as a dotted line. The Kalman filter in Figure 6.3 has a 'reasonable' value of the R_w elements associated with the updated parameters.

With reasonable σ_w^2 values and a ramp model, the Kalman Filter can track the auto-acceleration parameter as well as it could with a high σ_w^2 value and a random walk model.

However, the estimate is still lagging behind the true parameter and two parameters (the slope and the level) require updating.

A third option is to model the auto acceleration factor as an exponential function of mass of monomer fed. The auto acceleration factor is represented in the state equations as:

$$\beta = \beta_0 \exp(\xi * massfed) \tag{6.14}$$

The parameter ξ is updated as a random walk. The mass of monomer fed is used instead of time so that any variation in the feedrate can be accounted for. First, note that to apply equation (6.14), an estimate of β_0 is required. For the current simulations, the exact value of β_0 when the estimation begins is assumed. The robustness of the extended Kalman filter to this assumption is discussed in a later section. Second, ξ is scaled so that its value is on the same order of magnitude as the temperature measurements. Without scaling, the estimate of the auto acceleration factor exhibits a sharp spike when the estimation begins.

Figure 6.4 shows the performance of the extended Kalman Filter for a Product One, batch one simulation when the auto acceleration is modeled as an exponential. The value of σ_w^2 associated with ξ has a reasonable value (as per the guideline of equation (6.9)).

As Figure 6.4 shows, the Kalman filter can more easily track the auto acceleration when it has correct information about the parameter's structure. In future applications, the auto acceleration is modeled as an exponential in the Kalman filter equations.

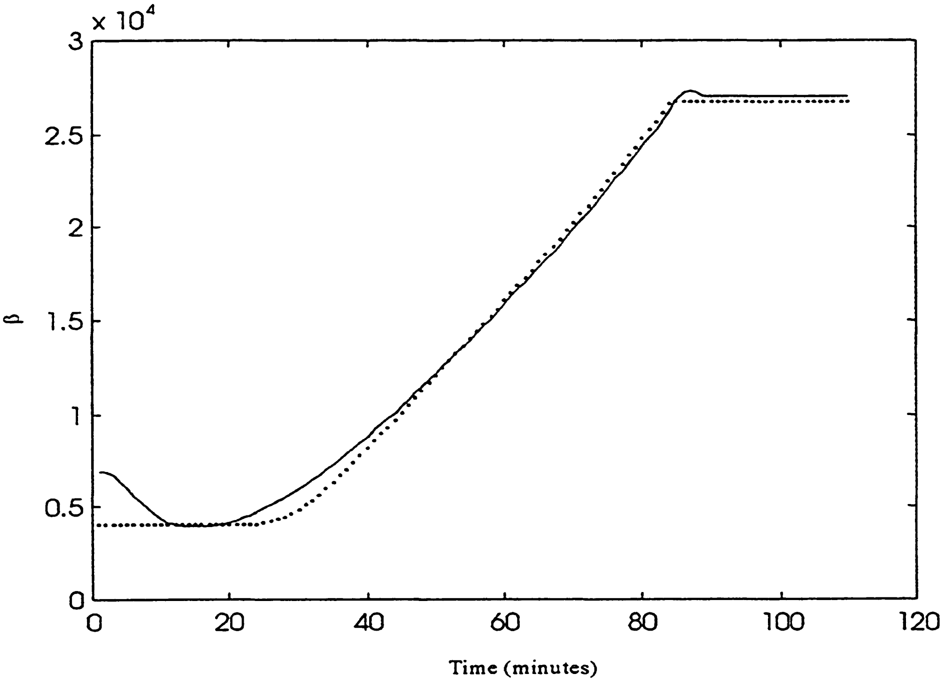


FIGURE 6.4: ESTIMATION OF AUTO ACCELERATION MODELED AS AN EXPONENTIAL; RW=DIAG{2, 2, 25}

In summary, several different models for the auto acceleration factor have been attempted. It is seen that the exponential model for the auto-acceleration most closely represents the parameter's true behavior and this is reflected in the smaller (more desirable) σ_w^2 value required to track β. Therefore, a priori knowledge about an unknown parameter is preferable for the successful implementation of the Kalman Filter.

6.4.2 ESTIMATION OF MONOMER IN THE REACTOR

In this section, the estimation of the moles of monomer in the reactor is considered. Due to observability constraints, only one of the moles of monomer and the auto acceleration factor can be updated by the extended Kalman filter. As a result, an on-line mole balance is coupled with the EKF to provide an estimate of the moles of monomer. It is assumed that the heat transfer coefficient is known for simulations in this section.

The main issue when estimating n_m and β individually is one of observability. In all molar and energy equations, the two terms appear as a product. Therefore, the individual parameters cannot be estimated separately by the Kalman filter. However, in the previous section, it was seen that the EKF performs poorly when trying to track the quickly changing product of βn_m.

The solution is to build structure into the Kalman filter for the individual parameters, thus providing the Kalman filter with information about how the product of the two parameters changes. In the previous section, structure is included for the auto acceleration factor (β) by modeling it as an exponential during the feed period. In this section, structure is included for the moles of monomer in the reactor in the form of a mole balance.

While the moles of monomer cannot be estimated directly with the Kalman filter, a mole balance can be integrated on-line. The mole balance for the moles of monomer is given by equation (6.15):

$$\frac{dn_m}{dt} = F_m - \hat{\beta} n_m e^{-\alpha/T} \tag{6.15}$$

where the molar flow of monomer, F_m, can be calculated knowing the mass flow and an estimate of the molecular weight of the monomer.

The mole balance is easily coupled with the Kalman Filter. Recall that integration of the process equations is the first step of Kalman filtering. The monomer mole balance is included with the reactor energy balances in this step. Therefore, the model equations are integrated to predict the reactor temperature, jacket temperature, moles of monomer and the stochastic state at t_{k+1}. The Kalman filter then updates the temperatures and the stochastic state based on on-line measurements. The mole balance is not updated (i.e. there is no feedback for the moles of monomer). Recall that β is only estimated during the monomer feed period. Likewise, the mole balance is integrated only if $\dot{m}_m \neq 0$. The moles of monomer are set to zero otherwise (please refer to the discussion of monomer-starved conditions in Chapter 4).

As one might expect, the openloop status of the monomer mole balance results in the estimate of the moles of monomer diverging from the true value as the batch proceeds. As a direct result, the estimate of the auto-acceleration factor (updated by the Kalman filter) also diverges. Figure 6.5 shows how the auto acceleration factor (β) and the moles of monomer (n_m) are tracked by the Kalman filter/on-line mass balance. The 'true' values of the parameters are solid lines and the estimated values are shown as dots.

Reasonable values for the R_w elements are used for the simulation in Figure 6.5. It should be noted, however, that even with very tight tuning, the individual parameters β and n_m are not tracked closely.

The results shown in Figure 6.5 may appear dismal at first. However, recall that the purpose of the extended Kalman filter is to provide estimates of unknown parameters to the controller. The auto-acceleration factor and the moles of monomer always appear as a product (the original source of problems). Therefore, the estimation of the product, not the individual components, is important for the controller. Figure 6.6 shows the estimated value of βn_m and the true value of this product.

By breaking βn_m into components, and incorporating structure about the components into the estimation scheme, excellent tracking of the product results. Of particular interest is how well the value of βn_m is estimated in the first few minutes after the feed starts. This is important to keep the initial temperature overshoot to a minimum. It should be noted that there is very little process/model mismatch in the simulations in this section. The effect of error on the performance of the estimator is discussed at the end of Chapter 6.

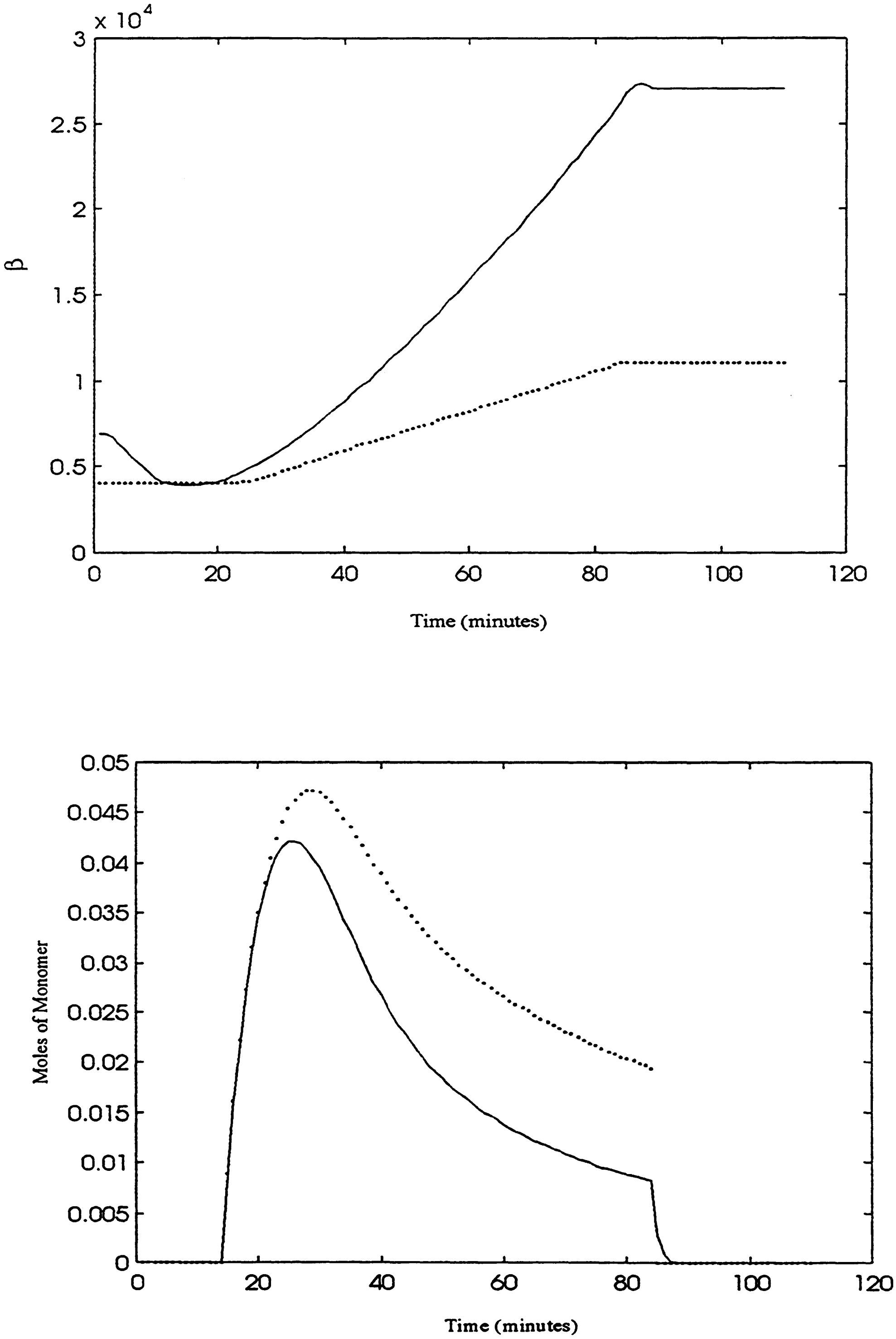


FIGURE 6.5: ESTIMATING OF AUTO ACCELERATION FACTOR AND MOLES OF MONOMER; RW=DIAG{2, 2, 50}

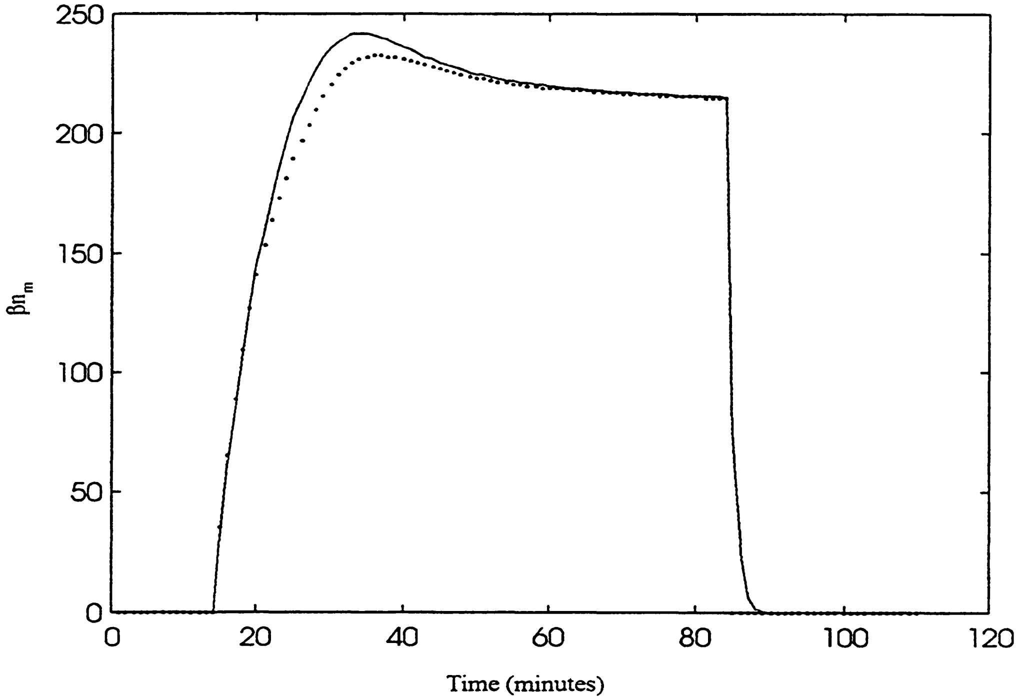


FIGURE 6.6: ESTIMATION OF THE PRODUCT βN_M ; RW=DIAG{2, 2, 50}

In summary, an openloop, on-line mass balance is coupled with the extended Kalman filter to provide an estimate of both the moles of monomer and the auto acceleration factor. In fact, by providing structure for the individual components, β and n_m, the product of the two, βn_m, is easily tracked by the Kalman Filter. This section re-emphasizes the idea that a priori knowledge of the process is helpful to benefit from the Kalman Filter.

6.4.3 ESTIMATION OF THE HEAT TRANSFER COEFFICIENT

The final parameter that requires on-line estimation is the heat transfer coefficient. In this section, the selection of a model for the heat transfer coefficient is discussed. It is shown that an inappropriate model can introduce estimation problems. A reasonable model for the heat transfer coefficient is presented and its implementation is shown to yield good results.

Consistent with the conclusions of the previous sections, an appropriate model for the heat transfer parameter (U) is very important for the success of the extended Kalman Filter. In pre-heat mode, the value of U is expected to be relatively constant. From a priori knowledge of polymerization processes, the heat transfer coefficient is expected to exponentially decrease during the feed (semi-batch mode). The change in U is due primarily to the increasing viscosity, and hence fouling, as the batch proceeds. Recall that the auto acceleration (β) of the rate of reaction is also caused by the increasing viscosity. From this, equation (6.16) was attempted as a model for the heat transfer coefficient during the feed period:

$$U = U_0 - \sigma\beta \tag{6.16}$$

where $\beta = \beta_0 \exp(\xi * \text{massfed})$. The parameter σ is modeled as a random walk and is updated on-line by the extended Kalman filter. Therefore, the EKF consists of:

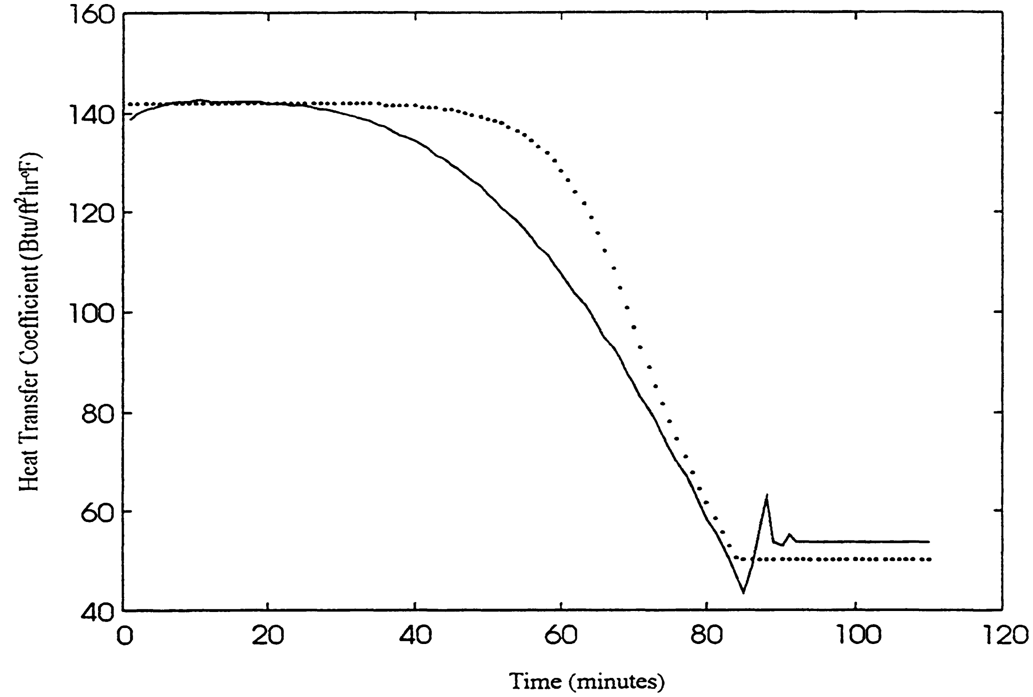
1/ Deterministic state equations for reactor temperature and jacket temperature;

2/ The auto acceleration factor, β, represented as an exponential in the state equations; the parameter ξ is modeled as a random walk and is updated by the Kalman filter;

3/ On-line openloop mole balance to estimate the moles of monomer;

4/ The heat transfer coefficient (U) is represented by equation (6.16) in state equations; the parameter σ is updated by the Kalman filter;

In Figure 6.7, it is assumed that the initial value of the heat transfer coefficient is known (U_0 will be treated later in this Chapter). The extended Kalman filter provides estimates of the heat transfer coefficient during the feed period. When the feed stops, the last estimate of U is retained (unlike the estimate of β, this value will be required for the nonlinear controller).



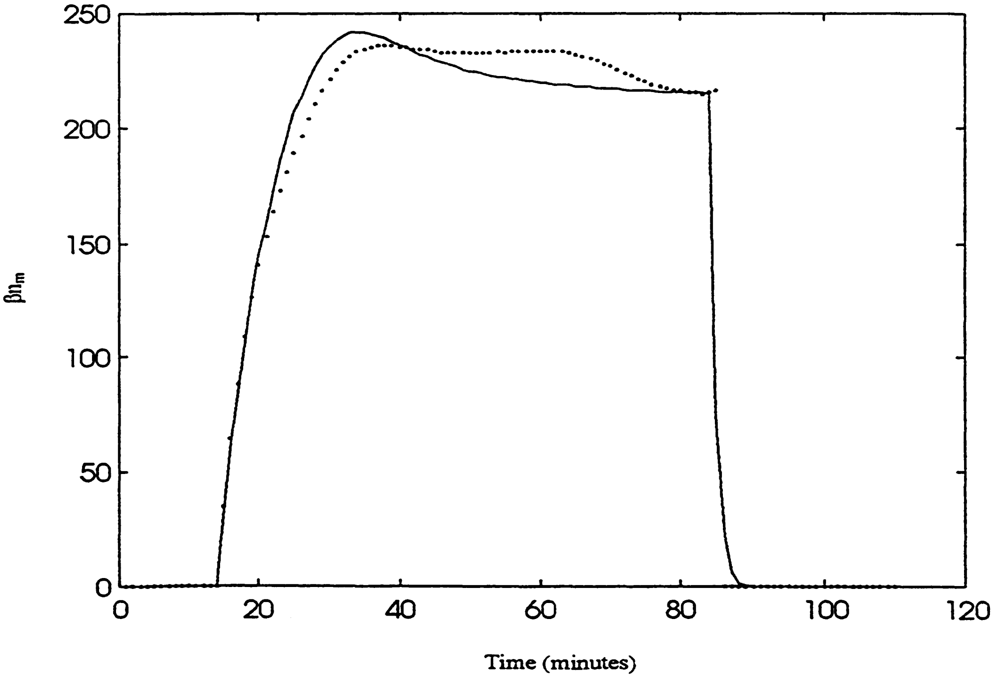


FIGURE 6.7: ESTIMATION OF βN_M AND THE HEAT TRANSFER COEFFICIENT (MODEL ONE); RW=DIAG{2, 2, 50, 0.1}

Because of the selection of the model, a large value of the R_w element associated with the heat transfer coefficient is required to track the heat transfer coefficient. Even with a large σ_w^2 element, the EKF has difficulty tracking the heat transfer coefficient. When noise is added to the temperature measurements, and the estimates of the parameters used in the nonlinear controller, stability problems are encountered in the estimates of the heat transfer coefficient (Figure 6.8).

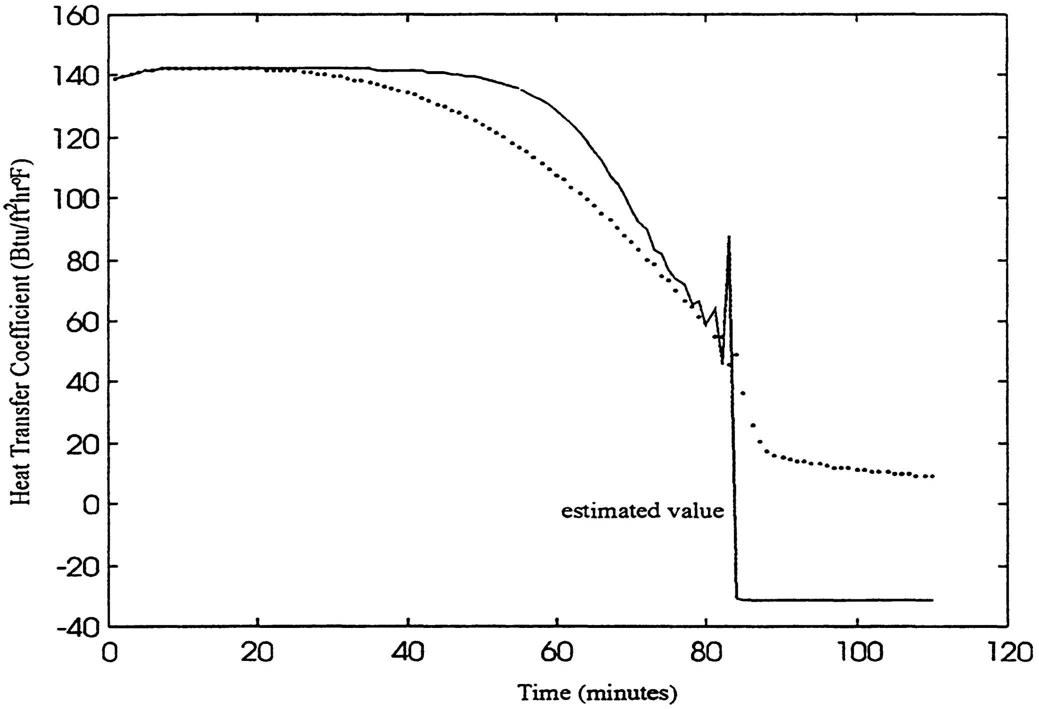


FIGURE 6.8: ESTIMATION OF U IN THE PRESENCE OF NOISE (MODEL ONE)

Once again, this illustrates the need for appropriate models for the unknown parameters in the Kalman Filter.

After some trial and error, the final form of the model for the heat transfer coefficient (U) is arrived at:

$$U = U_0 - \phi_3 \omega^{\phi_4} \tag{6.17}$$

where ϖ is the power to the agitator. By representing U as a function of the agitator power, equation (6.17) acknowledges that the heat transfer coefficient depends on the reactor contents viscosity. In a polymer processing environment, viscosity measurements are not readily available, however the a variable such as agitator power is. The two are directly related; as viscosity increases, so will the power supplied to the agitator. Depending on the system, variables other than agitator power may be used to infer viscosity.

Since there is no expression for agitator power in the Chylla Haase system, equation (6.17) is modified, and the actual viscosity is used to represent the agitator power:

$$U = U_0 - \phi_3 \mu^{\phi 4} \tag{6.18}$$

Equation (6.18) contains two unknown parameters, ϕ_3 and ϕ_4. However, an expression for ϕ_3 can be obtained by noting that at the end of the feed:

$$U_f = U_0 - \phi_3 \mu_f{}^{\phi 4} \tag{6.19}$$

Rearranging equation (6.19):

$$\phi_3 = (U_0 - U_f)\left(\frac{1}{\mu_f}\right)^{\phi 4} \tag{6.20}$$

Therefore, equation (6.18) becomes:

$$U = U_0 - (U_0 - U_f)\left(\frac{\mu}{\mu_f}\right)^{\phi 4} \tag{6.21}$$

Equation (6.21) describes how the heat transfer coefficient changes with time during the feed period. To implement it, initial and final values of U and the final viscosity (agitator power) are required. The final viscosity should be the approximately same for every batch of polymer one, therefore an average value for the final viscosity (agitator power) is easily obtained from experience. The final heat transfer coefficient will not be the same for each batch. Prior to implementing the EKF, the value of U_f for each batch can be obtained by running an on-line energy balance at the end of each batch:

$$U_f = \frac{-UA_{loss}(T - T_{amb}) - (m_w C_{pw} + m_s C_{ps})_f \dfrac{dT}{dt}}{A(T - T_{jout})} \tag{6.22}$$

Since the temperature is held constant, and there is no heat released, the implementation of this energy balance is simple. It is assumed that good estimates of the parameters on the right hand side of equation (6.22) are available (Appendix C). It is seen that while U_f is unique to each batch, the difference from batch to batch is not large. Therefore, an average value for the final heat transfer coefficient, $\overline{U}_f$, can be used in equation (6.21).

Estimating the initial value of the heat transfer coefficient is more difficult. The value of U_0 can be estimated using an on-line energy balance during the heating period:

$$U_0 = \frac{-UA_{loss}(T - T_{amb}) - (m_w C_{pw} + m_s C_{ps})_0 \dfrac{dT}{dt}}{A(T - T_{jout})} \tag{6.23}$$

Implementing the on-line energy balance of equation (6.23) indicates that excellent estimates of U_0 are possible when the manipulated variable, $T_{jin,\,sp}$, is at its maximum value. During this time, the reactor temperature is increasing at a constant rate, and accurate estimates of dT/dt are possible. After the manipulated variable comes off the upper bound, the reactor temperature behaves less predictably, and the estimate of the heat transfer coefficient deteriorates. Therefore, the last estimate of U_0 while $T_{jin,sp}$ is at its constraint is used in equation (6.21).

Unlike the final value of the heat transfer coefficient, U_0 is significantly different from batch to batch. Therefore, an on-line energy balance for U_0 is required for each batch. Note that if no process changes are expected, the on-line energy balance for U_0 can eventually be replaced with a table of initial heat transfer values for each batch.

Returning to the model of the heat transfer coefficient, only ϕ_4 remains unknown. A brief off-line study in a spreadsheet shows that a constant value of ϕ_4 adequately models the trajectory of the heat transfer coefficient for all batches of product one. Therefore, it is not necessary to update ϕ_4 on-line. Of course, one could proceed to model ϕ_4 as a random walk, and update it on-line via the extended Kalman filter. The same conclusion, ϕ_4 is constant, would be found. Eventually the Kalman filter (for ϕ_4) could be taken off-line and replaced with the appropriate constant value.

Equation (6.21) is coupled with the on-line energy balance for U_0 (equation (6.23)) and implemented on the semi-batch reactor. Table 6.2 summarizes the parameter values that the user is required to supply to the estimation algorithm (values are given for Product One). The same parameters are required for Product Two, however they will have different values.

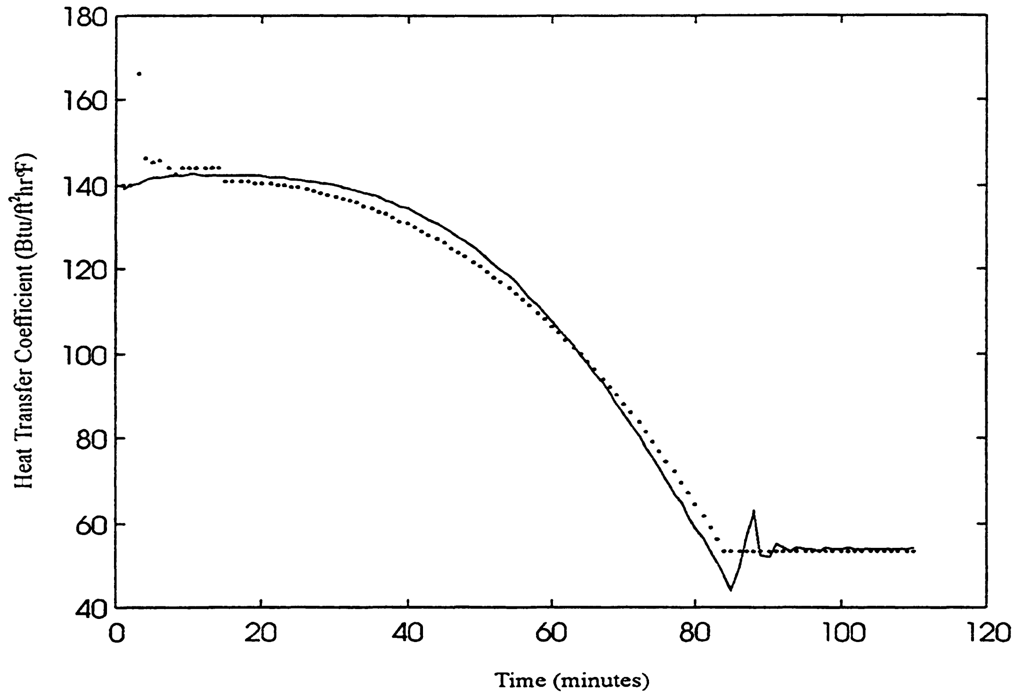
Table 6.2: Summary of Required Parameter Values, Product One

Parameter	Value for Product One
β_0	4000
U_f	48 Btu/hr ft^2 °F
μ_f	193 cp
ϕ_4	0.7

Figure 6.9 shows the estimation of the product, βn_m, and the heat transfer coefficient for a Product One, batch one, summer simulation. Some measurement noise ($\sigma = 0.04$ °F) is included.

As Figure 6.9 shows, the combination of the extended Kalman filter and off-line parameterization is successful in estimating the unknown process parameters. Similar performance is observed under batch five conditions.

In summary, the estimation of the heat transfer coefficient has been discussed in this section. Once again, the importance of choosing an appropriate model is seen. With some modeling effort, the EKF can successfully estimate the required parameters. The next step is implementing the estimates of the process parameters in the nonlinear controller. This is discussed in the next section.



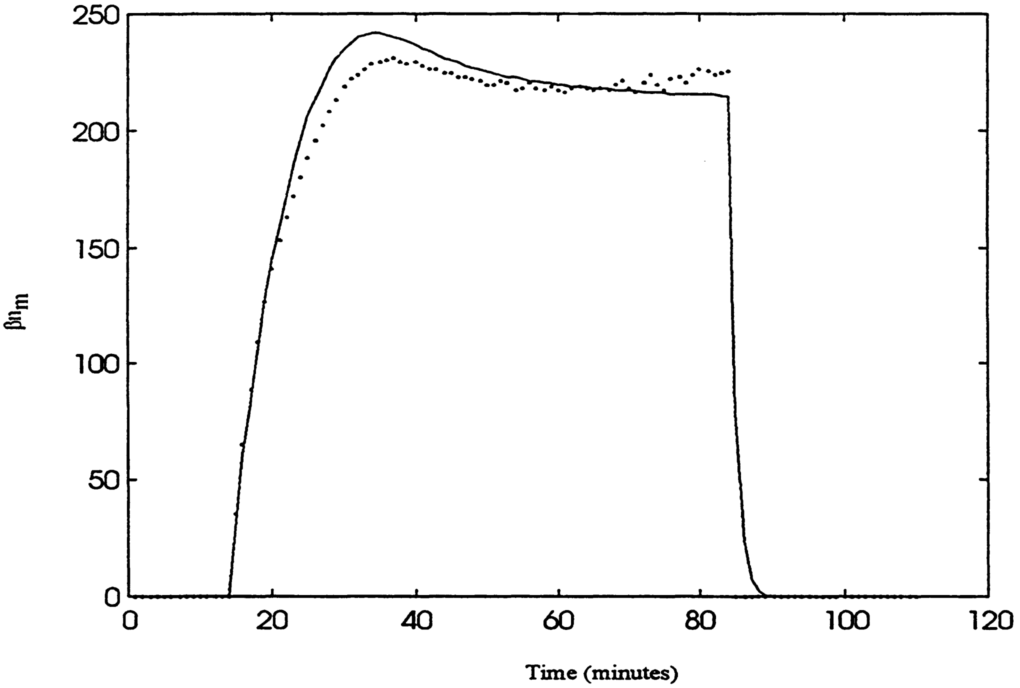


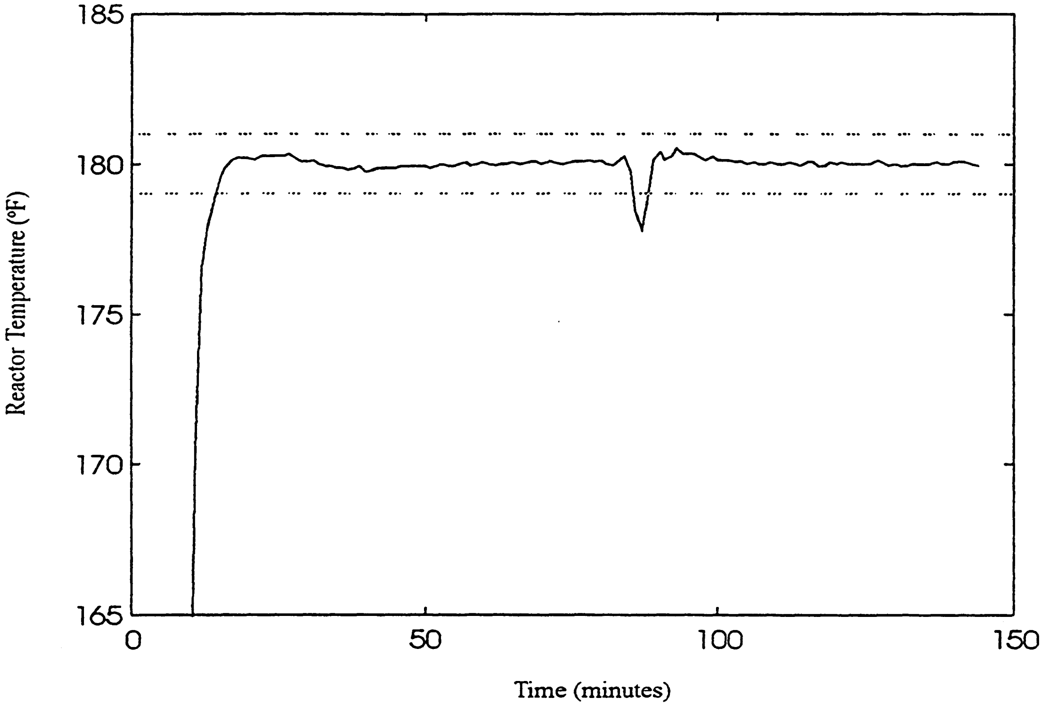
FIGURE 6.9: ESTIMATION OF βN_M AND THE HEAT TRANSFER COEFFICIENT (MODEL TWO); RW=DIAG{2, 2, 50}

6.5 Nonlinear Adaptive Control Performance

6.5.1 Performance under Ideal Conditions

The goal of the estimation algorithm is to provide the required parameter estimates to the controller. Ultimately, the performance of the estimator is determined by the resulting temperature control. In this chapter, the nonlinear controller using the parameter estimates is presented for Product One. Integral action is included in the nonlinear controller. It is found that for the ideal case (summer, batch one), the nonlinear adaptive control is comparable to the nonlinear controller in which all parameters are known.

Noise with a standard deviation of 0.04 °F is added to the simulations in this section (note that noise of standard deviation of up to 0.5 °F was tested to ensure the robustness of the estimator, however 0.04 °F is considered representative of noise on filtered temperature measurements). Parameter estimates for the auto acceleration, moles of monomer and heat transfer coefficient are supplied to the nonlinear controller by the estimation algorithm during the feed. The on-line energy balance (discussed in the previous section) provides an estimate of U to the nonlinear controller during heat up as well as providing a value of U_0 for the heat transfer model. Figure 6.10 shows the temperature control for a summer, batch one, Product One simulation:



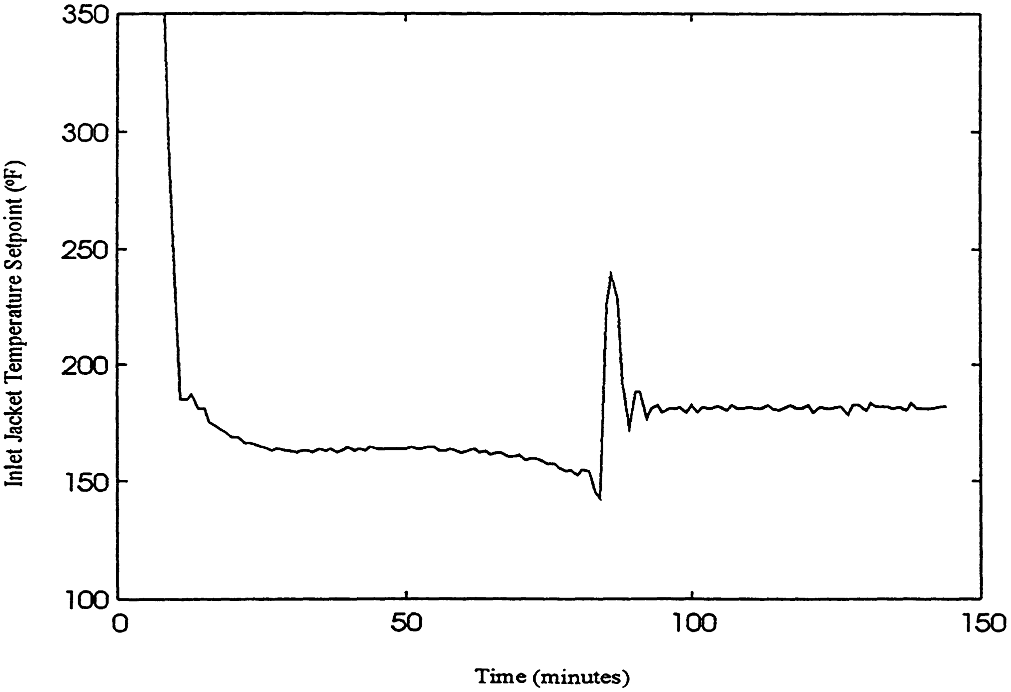


FIGURE 6.10: NONLINEAR ADAPTIVE CONTROL (PRODUCT ONE)

The nonlinear adaptive control performance is comparable to that of the nonlinear controller in which all parameter values are known. Therefore, the estimation is not degrading control from the best possible case. The nonlinear adaptive controller however, also has problems maintaining the temperature within one degree of setpoint when the feed stops. From the discussion in Chapter 5, this can be fixed by decreasing the execution interval of the controller, or by adjusting the time that the feed stops by a half minute. With respect to the last option, it is unlikely that the feed will be stopped after exactly 70.00 minutes for every batch. In the true processing environment, it is expected that some batches will have better temperature regulation when the feed stops because of this randomness. Figure 6.10 shows the worst case possible, where almost a full minute has passed before the controller records a feed flow measurement and knows that the feed has stopped.

In summary, it is seen that the nonlinear adaptive controller provides good temperature regulation for the situation considered in this section.

6.5.2 Robustness of the Nonlinear Adaptive Controller

In this section, the robustness of the nonlinear adaptive controller is evaluated for two different types of errors. First, the controller is implemented with errors in the initial parameter guesses (see Table 6.2) and it is found that the nonlinear adaptive controller is robust to reasonable uncertainties in these variables. Second, the effect of unmodeled events, such as changes in process parameters listed Table 6.1, is evaluated. It is shown that the Extended Kalman Filter provides the nonlinear controller with an added robustness feature.

For the batch simulation in Figure 6.10, good values of U_0, $\overline{U}_f$, ϕ_4 and β_0 are assumed. It is important to understand the robustness of the nonlinear adaptive controller to errors in these values. The regulation of temperature is considered as a measure of the robustness of the estimation/control algorithm. A short robustness study is undertaken and Table 6.3 summarizes the simulations. Over- and underestimated values of the initial parameters are implemented in the estimation, and the resulting control assessed. Only Product one, batch one, summer conditions are considered.

Table 6.3: Robustness Simulations for Product One, batch one, summer Conditions

Simulation	U_0(Btu/hr ft^2 F)	U_f (Btu/hr ft^2 F)	β_0	ϕ_4
nominal value	140	48	4000	0.7
high U_0	180 (+28%)	48	4000	0.7
low U_0	100 (-28%)	48	4000	0.7
high U_f	140	60 (+25%)	4000	0.7
low U_f	140	35 (-25%)	4000	0.7
high β_0	140	48	8000 (+100%)	0.7
low β_0	140	48	1000(-75%)	0.7
high ϕ_4	140	48	4000	0.5 (-28%)
low ϕ_4	140	48	4000	0.9 (+28%)

An error in U_0 represents a situation where the initial on-line energy balance is incorrect. The other three parameters are actually specified by the user. The error in β_0 is tested most rigorously, as it is expected that the user has very little information about the auto acceleration factor.

In the interest of brevity, only two representative simulations from Table 6.3 are shown. The nonlinear adaptive controller is remarkably robust to the selected errors. Figure 6.11 shows the temperature control when the initial value of the heat transfer coefficient is overestimated. The nonlinear adaptive controller with the nominal U_0 value is shown as a dotted line.

Figure 6.11 represents a situation in which the controller believes it has more heat transfer ability that it actually does. This is of particular concern for exothermic reactors and it could result in a case where the reaction runs away. For the semi-batch reactor, good temperature control is maintained despite the error in U_0. In fact, it appears that better control results when the feed is stopped. However the improvement is actually due to the fact that the feed stops at 84.8 minutes, only 0.2 minutes before the controller's next execution.

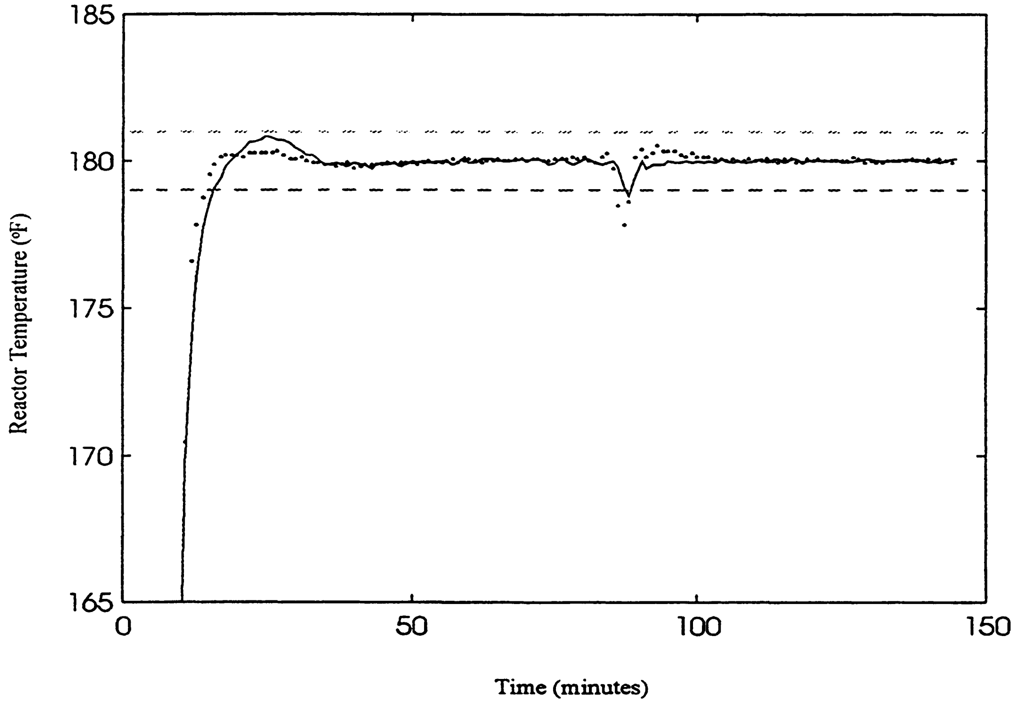


FIGURE 6.11: NONLINEAR ADAPTIVE CONTROL WITH INCORRECT U_0 VALUE

Figure 6.12 shows the batch run in which the user underestimates β_0. The temperature overshoots the setpoint by 1.5 degrees and is slow to return to the desired value. Once again, the nonlinear adaptive controller with the nominal value of β_0 is shown as a dotted line.

Despite the degradation in control, the system remains stable, which is of primary concern. Since the auto acceleration factor is updated by the Kalman filter, a good initial guess of β_0 can be iteratively obtained with the following procedure:

1/ Specify an initial guess of β_0 and running the extended Kalman filter off-line during a batch;

2/ Observe the value of the auto acceleration predicted by the estimation during the batch;

3/ Specify a 'better' guess of β_0 for the next batch and repeat the process;

The above procedure will eventually result in a good estimate of β_0.

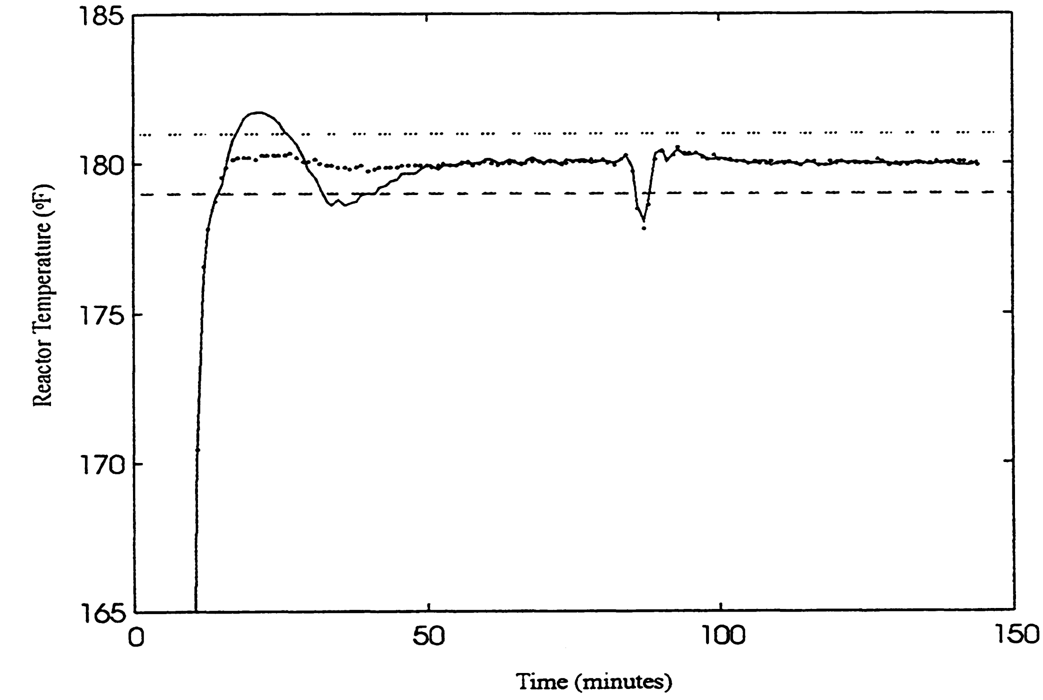


FIGURE 6.12: NONLINEAR ADAPTIVE CONTROL WITH INCORRECT β_0 VALUE

The second type of process/model mismatch considered relates to unknown or unmodeled events occurring in the process. An example would be unknown changes in the variables in listed Table 6.1. Recall that many of these parameter values were assumed known and/or constant. It is important that the nonlinear adaptive controller be able to handle these situations.

For brevity, only one simulation is shown in this section, however several different cases were tested. In order to evaluate the robustness of the nonlinear adaptive controller, the monomer specific heat, C_{pm}, is modeled in the process as a drifting variable with an initial value of 0.4. The model in the nonlinear controller and the Kalman Filter retains a constant value of 0.4. Figure 6.13 shows the drifting process variable C_{pm}:

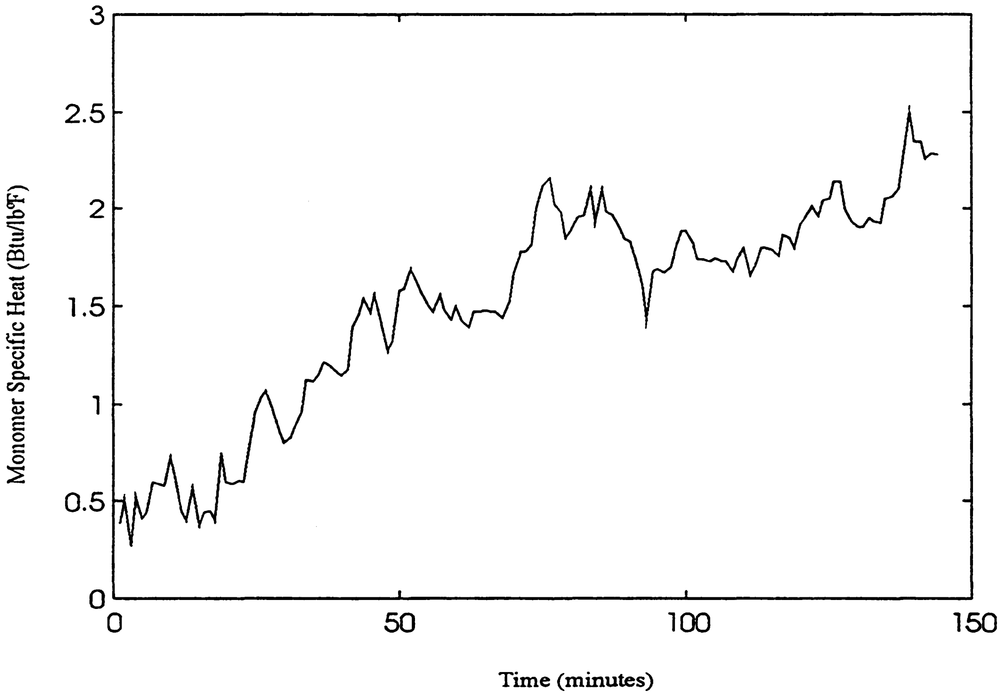


FIGURE 6.13: VALUE OF MONOMER SPECIFIC HEAT IN PROCESS EQUATIONS

The performance of the nonlinear adaptive controller is shown in Figure 6.14. The estimation of the product βn_m is shown below the temperature plot.

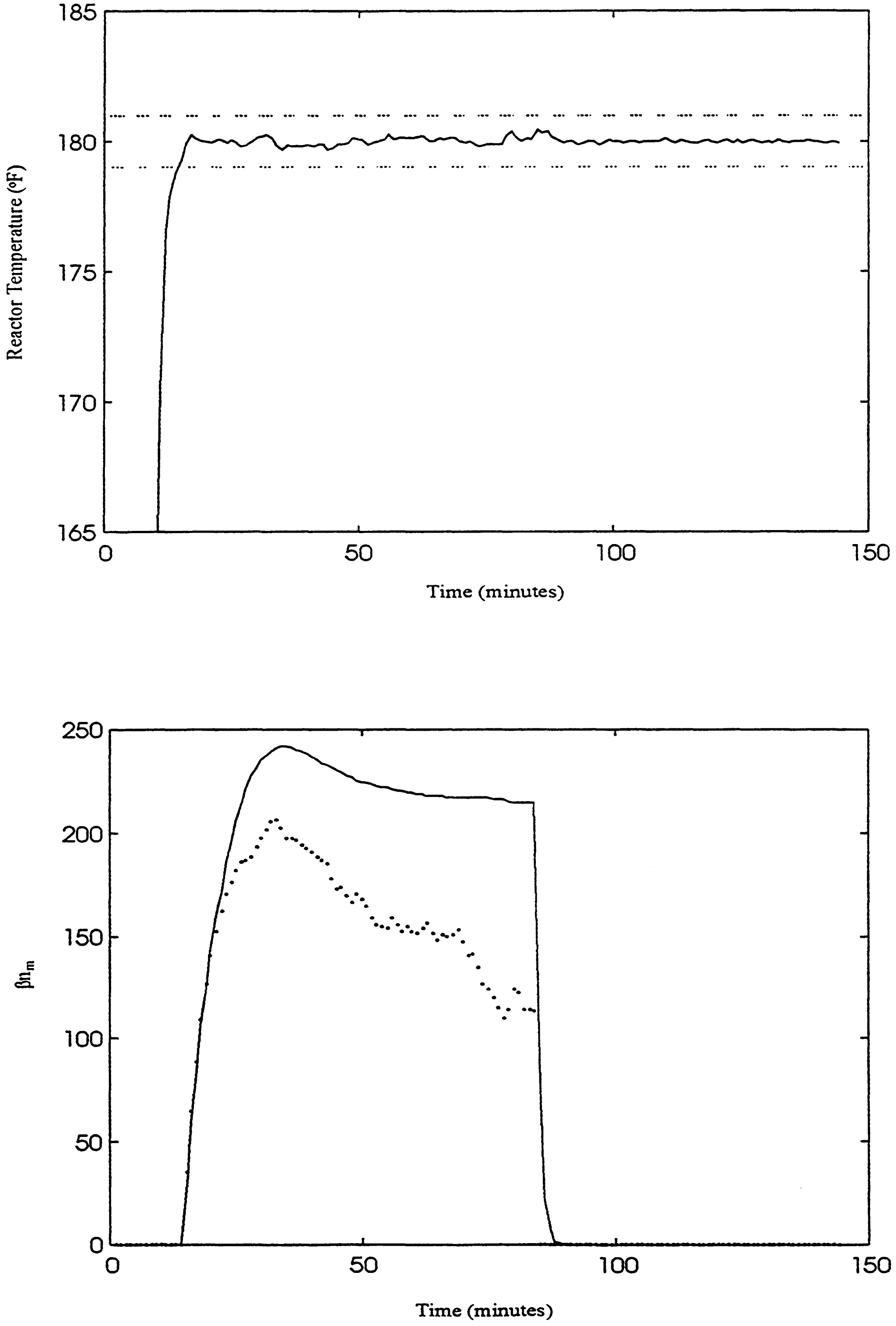


FIGURE 6.14: NONLINEAR ADAPTIVE CONTROLLER PERFORMANCE

Despite the significant and unknown variations in C_{pm}, the nonlinear controller provides good temperature regulation. The robustness of the controller is a direct result of the Extended Kalman Filter. As Figure 6.13 shows, the estimated value of βn_m (dotted line) slowly departs from the true value of the parameter (solid line). In order to compensate for the model error introduced by the drifting C_{pm} value, the EKF decreases the estimate of βn_m. That is, the error in the C_{pm} value (physical meaning: the feed is providing more cooling to the reactor contents) can be effectively represented in the model equations as a lower βn_m value (physical meaning: less heat released due to reaction). Any process/model mismatch which can be interpreted as a change in the heat released falls into a broad class of error that will be handled by the EKF in this manner. Therefore, the Extended Kalman Filter is providing a flexible robustness feature to the nonlinear controller.

In summary, the nonlinear controller using parameter estimates has been implemented in this chapter. It is seen that the nonlinear adaptive controller provides temperature control that is comparable to that of the nonlinear controller with all parameters exactly known. Several robustness studies are also presented. It is found that the performance of adaptive controller is robust to reasonable uncertainties in initial estimates of β_0, U_0, ϕ_4 or U_f. Finally, it is seen that the Extended Kalman Filter improves the robustness of the nonlinear controller for a general class of process/model mismatch.

CHAPTER 7

COMPARISON OF CONTROL ALGORITHMS

7.1 INTRODUCTION

In this chapter, three of the control algorithms developed in the thesis are compared:

- PID algorithm;
- PID with feedforward compensation for heat released and feed stoppage;
- Nonlinear Adaptive controller;

The following five situations are presented: performance under ideal conditions (summer, batch one, product one), effect of fouling (batch to batch variation), effect of ambient conditions (summer/winter), process/model mismatch, and extension to product two. These situations are suggested in Chylla and Haase (1993) for testing controllers for robustness.

7.2 PERFORMANCE UNDER IDEAL CONDITIONS

Ideal conditions for the controllers are assumed to be Product One, summer, batch one simulations. These simulations are shown in previous chapters and are repeated here to compare the controllers. Noise ($\sigma = 0.04$ °F) is added to all simulations in this chapter.

Figure 7.1 shows the performance of the three controllers. The well tuned PID is shown as dash-dot lines, the PID with feedforward compensation plots with dots and the nonlinear adaptive controller is shown as a solid line. The ± 1 °F limits are also plotted.

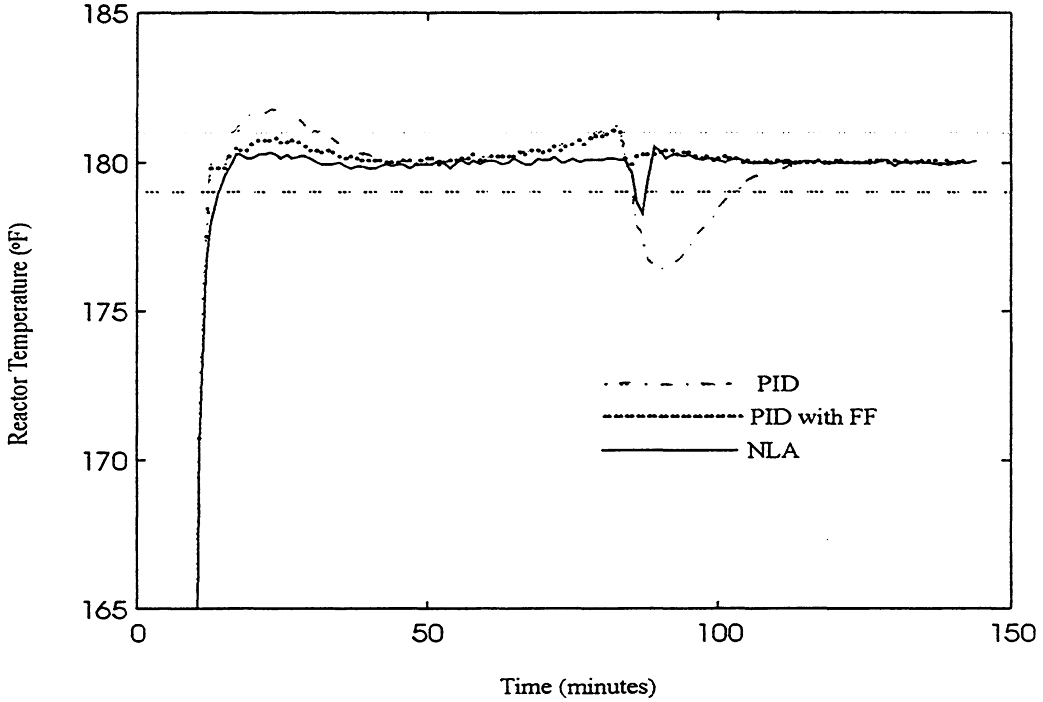


FIGURE 7.1: PERFORMANCE OF CONTROLLERS FOR PRODUCT ONE, BATCH ONE

Only the PID with feedforward compensation maintains the reactor temperature within the required bounds. The information about the heat released helps this controller minimize the initial temperature overshoot, however the temperature slowly moves off its setpoint near the end of the feed. The well tuned PID exhibits the same behavior around 80 minutes. The likely cause of this can be related to the process information incorporated in the PID algorithms: neither knows about the decreasing heat transfer coefficient. Both the PID class controllers are relying on feedback to correct the temperature deviation introduced by falling heat transfer ability. The nonlinear adaptive controller does not exhibit a control problem at the end of the feed, and it does have information about the decreasing heat transfer coefficient. A model for U provides the nonlinear adaptive controller with a form of feedforward control for this 'disturbance'.

From Figure 7.1, it is not clear which controller, the nonlinear adaptive or the PID with feedforward action, provides superior control. The PID controller maintains the temperature between 179 °F and 181 °F at all times. However, the nonlinear adaptive algorithm keeps the temperature closest to the setpoint (180 °F) overall, and a shorter control interval would eliminate the excursion at the end of the monomer feed. Ultimately, the effect that each temperature trajectory has on polymer quality will determine the superior controller.

7.3 EFFECT OF FOULING

In this section, the robustness of the controllers to fouling is presented. Each of the algorithms is applied to a Product One, batch five, summer simulation. Tuning from the previous section (summer, batch one) is used. Noise is present in all simulations.

Figure 7.2 shows the performance of the three controllers - well tuned PID (dash-dot), PID with feedforward compensation (dot) and nonlinear adaptive (solid). Once again, the limits of 181 °F and 179 °F are shown.

The PID algorithms show a serious degradation in control. Both allow the reactor temperature to overshoot the setpoint by 6 °F. The nonlinear adaptive controller is very robust to different batches, and shows little change from the summer, batch one simulation.

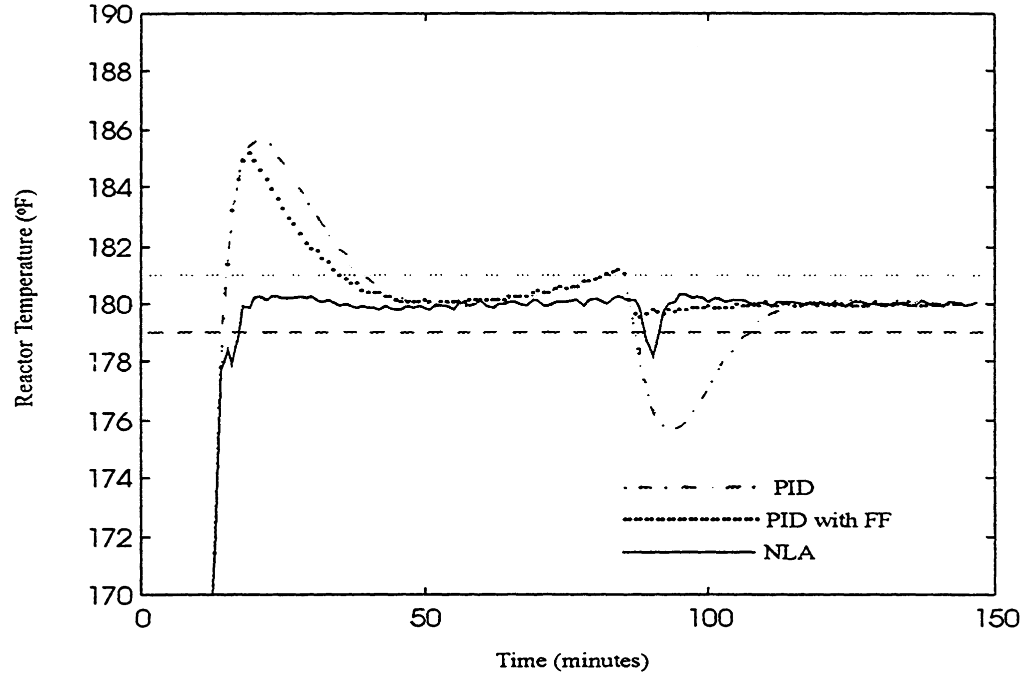


FIGURE 7.2: EFFECT OF FOULING: BATCH FIVE CONDITIONS

Of course, the robustness of the nonlinear adaptive controller is a direct result of the information built into its structure. Recall that this algorithm has an general on-line energy balance to provide it with an estimate of the initial heat transfer coefficient (U_0) during heatup. Therefore, it is aware of how the heat transfer varies from batch to batch. The same type of information may be given to the PID class controllers by having different tuning for each batch. Figure 7.3 shows the performance of the well tuned PID for a batch five, summer simulation when the controller has been tuned for batch five. The temperature control of the well tuned PID for a batch one simulation is shown as a dotted line for reference.

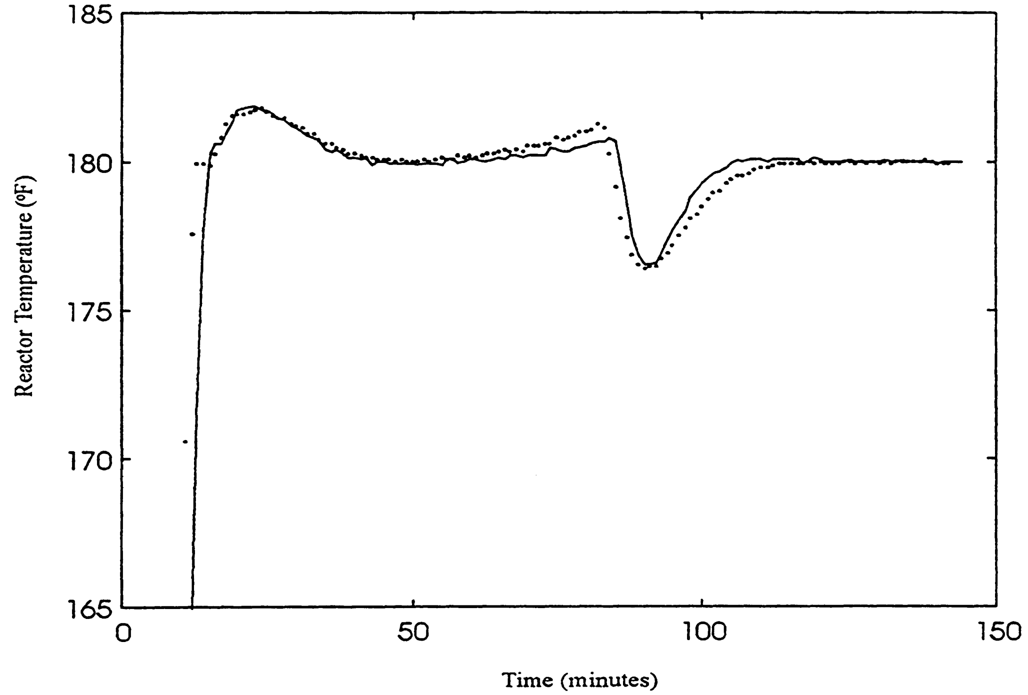


FIGURE 7.3: WELL TUNED PID WITH TUNING FOR BATCH FIVE

Figure 7.3 shows that the performance of the PID class controllers is consistent from batch to batch *if* different tuning is implemented for each batch.

For a process with many batches, finding new tuning parameters for each new situation can be very time consuming. In such a case, the nonlinear adaptive controller becomes very appealing. Its structure changes with the values of the process parameters such as the heat transfer coefficient, in order to provide the same closed loop response (the desired error trajectory) when the plant model changes. In this section, the plant model changes as a result of different levels of fouling. It should be noted that if an average value of U_0 is implemented in the adaptation algorithm, a degradation of control would undoubtedly be observed in the nonlinear adaptive controller.

In summary, the performance of the three controllers for batch five conditions has been shown. It is seen that the nonlinear adaptive controller is more robust to changes in the level of fouling. This is a direct result of the information included in its structure about the initial heat transfer coefficient. The PID class controllers tuned for ideal conditions exhibit poor control for batch five runs. The control performance can be improved by retuning the PID algorithms for each batch.

7.4 Effect of Ambient Conditions

In this section, the robustness of the three controllers to changing ambient conditions is tested. Simulations for product one, batch one, winter conditions are considered.

In winter, the ambient temperature is lower, and this affects the system in two ways. First, heat loss from the reactor is larger. Second, the monomer feed is cooler when entering the reactor in winter. Figure 7.4 shows the performance of the three controllers for a Product One, batch one, winter simulation.

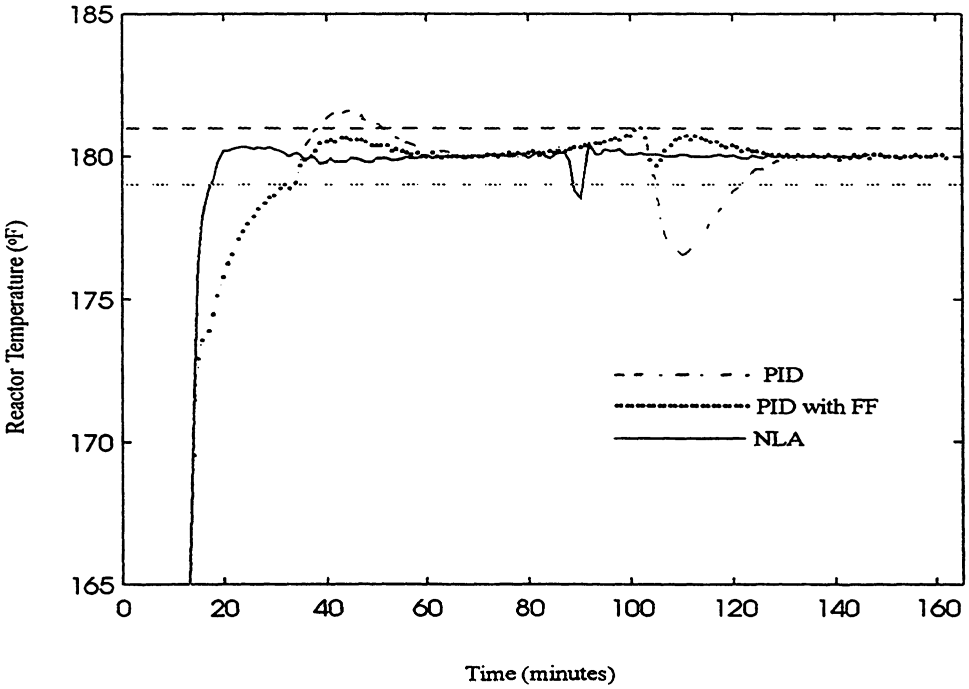


Figure 7.4: Effect of Ambient Conditions: Winter Simulation

Once again, the nonlinear adaptive controller shows excellent robustness to changing ambient conditions. While the PID with feedforward compensation is able to maintain the reactor temperature within ± 1°F, is has problems quickly heating the reactor contents. This could easily be amended, however, by requiring that the manipulated variable $T_{jin,sp}$ be kept at its upper bound until a specified temperature (one degree before setpoint for example).

The robustness of the nonlinear adaptive controller is once again due to the information included in its structure. The parameter T_{amb} appears in the process model used in the controller. Therefore, different values of T_{amb} change the structure of the nonlinear adaptive controller so that the desired closed loop response is achieved.

In summary, the nonlinear adaptive controller is more robust to changing ambient conditions than the PID class controllers (in their current form).

7.5 Effect of Process/Model Mismatch

7.5.1 Random Variations in Reaction Rate

Chylla and Haase (1993) include a random factor, i, that multiplies the rate of reaction. The value of 'i' ranges between 1.2 and 0.8. This is an attempt to model the batch to batch variations in the reaction rate due to feed impurities. In previous simulations, 'i' was set to one. In this section, simulations with i = 1.2 and i = 0.8 are considered. It is seen that all three controllers are robust to unknown changes in the rate of reaction.

When 'i' is equal to 1.2, the rate of reaction is described by equation (7.1):

$$R_p = 1.2\beta e^{-\alpha/T} n_m \tag{7.1}$$

The control algorithms are not modified to account for this change. Figure 7.5 shows the performance of the three controllers for a Product One, batch one, summer simulation with 'i' set to 1.2:

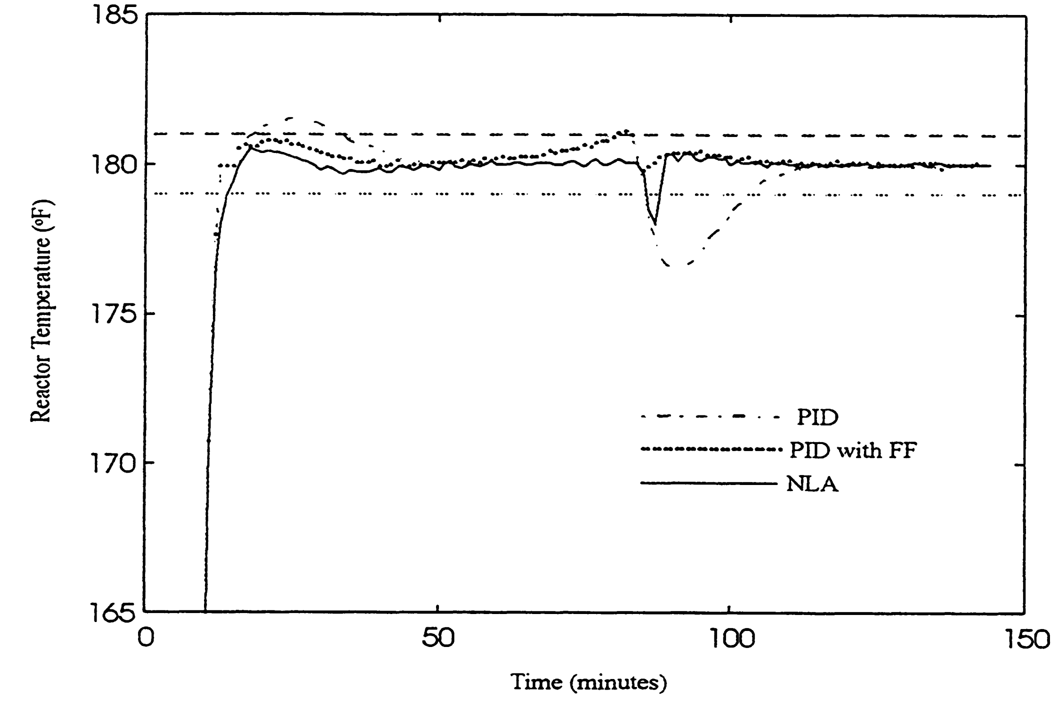


FIGURE 7.5: EFFECT OF RANDOM VARIATIONS IN RATE OF REACTION

Very little difference is seen in the control under these circumstances, as compared to an 'ideal' batch. The reason for this is simple. Although R_p includes a factor of 1.2, the rate of reaction (and therefore the heat released) does not increase. A higher rate of reaction results in the monomer being consumed more quickly. Therefore, the moles of monomer, n_m, in the s system is lower. For the Chylla Haase semi-batch reactor, the factor of 1.2 is almost exactly balanced by a decrease in monomer. The rate of reaction (and the heat released) in Figure 7.5 is very close to that in Figure 7.1 (ideal conditions).

7.5.2 Changes in Jacket Dynamics

Chylla and Haase (1993) also indicated that the jacket dynamics change from batch to batch. Equation (7.2) models the coolant recirculation loop (note that this equation is not included in the nonlinear controller).

$$\frac{dT_{jin}(t)}{dt} = \frac{dT_{jout}(t-\theta_2)}{dt} + \frac{T_{jout}(t-\theta_2) - T_{jin}(t)}{\tau_j} + \frac{K_j}{\tau_j} \tag{7.2}$$

where K_j is the heating/cooling process gain and is a function of the slave controller output. Chylla and Haase state that changes of 25% are possible in the time constants and gains. In this section, a change in the time constant (τ_j) of 50% is considered.

The nominal value of the time constant is 0.67 minutes. Figure 7.6 shows the control performance with the jacket time constant equal to one minute.

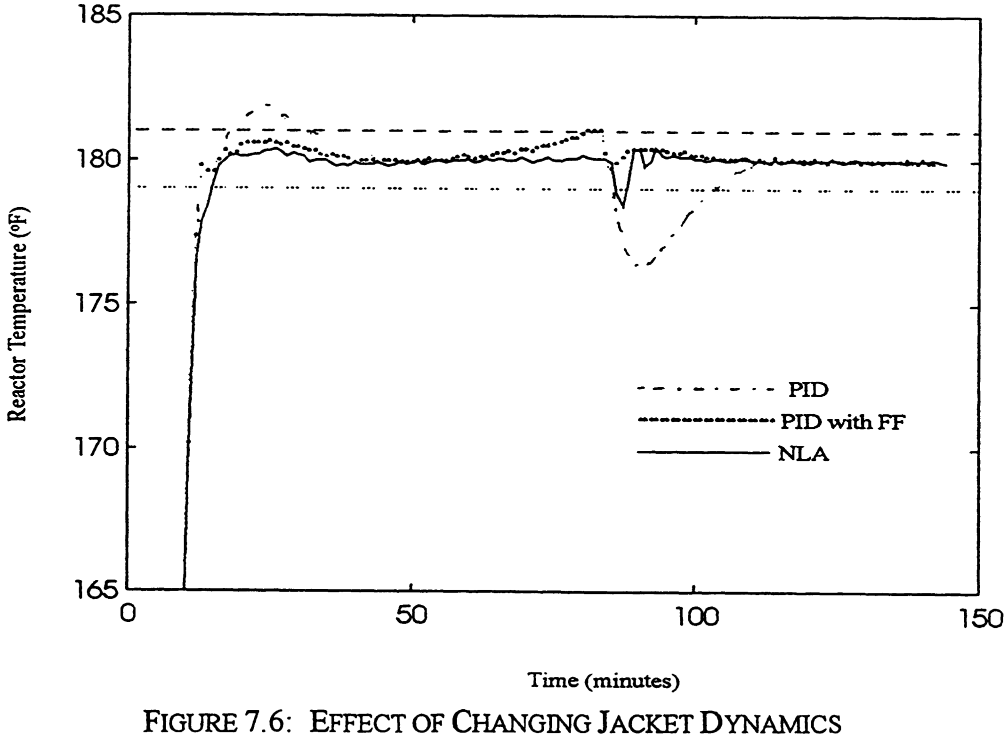


FIGURE 7.6: EFFECT OF CHANGING JACKET DYNAMICS

Once again, very little deviation from the ideal situation is seen. While the time constant of the jacket dynamics has changed, it is well within the bounds of stable control for the tuning selected. As well, even with a 50% increase, the jacket dynamics are much faster than the process (reactor) response times.

In summary, it is seen that all three controllers are robust to the expected process variations suggested by Chylla and Haase (1993).

7.6 MULTIPLE PRODUCT CONSIDERATIONS

7.6.1 EXTENSION TO PRODUCT TWO

The primary concern with the second product is the increase in information required. Two issues are addressed in this section:

- Tuning
- Process Parameters

For the PID class controllers (PID and PID with feedforward compensation), the retuning presents the largest barrier for multiple product systems. For each new product, new tuning is required for good performance. Well tuned controllers are the result of numerous time consuming trial and error batches. For the Chylla Haase reactor, separate PID tuning factors are required for each batch of each product, for a total of ten sets of tuning rules for only two products. For the feedforward algorithm, transfer functions for the heat released and the feed stop must be derived for each product. The on-line energy balance for Q_r can be implemented in the same form regardless of the product.

The nonlinear adaptive controller has advantages over the PID controllers because of its mechanistic model based approach. A desired closed loop trajectory can be specified for each product if necessary (for the Chylla Haase reactor, values of d_1, d_2 and d_3 for product one also give good control for product two). The controller structure changes automatically in response to different on-line measurements and values of the process parameters. At most, for a two product reactor, two sets of tuning (two error trajectories) will require specification.

The second point of interest is the process parameters. For the three controllers, Table 7.1 summarizes the process parameters that the algorithm requires to calculate the manipulated variable output. A value for each process parameter is required for each product:

Table 7.1: Summary of Process Parameters

Controller	Process Parameters Required
PID	-
PID with Feedforward Compensation	U_{est}, A, C_{pm}, $\dot{m}_m$, $\Sigma(m_iC_{pi})$, T_{feed} where U_{est} is parameterized off-line
Nonlinear Adaptive Controller	U, A, C_{pm}, $\dot{m}_m$, $\Sigma(m_iC_{pi})$, T_{feed}, $e^{-\alpha/T}$, ΔH_r, m_c, $\dot{m}_c$, C_{pc}, β, n_m, UA_{loss} where U, β, n_m are updated on-line (approximate shapes must be known)

Since U, β and n_m are updated on-line, some knowledge of how these parameters are expected to change is needed. Clearly the nonlinear controller requires far more process knowledge, and more modeling effort to implement. The payoff is in its more general nature, once the initial modeling is complete.

Figure 7.7 shows a simulation for Product Two, batch one, summer conditions. Only the nonlinear adaptive controller is able to maintain the temperature within the bounds of good control for Product Two.

Only the first feed period is shown, due to the process problems noted in Chapter 3. Both the PID class controllers have been tuned for Product Two, batch one, summer conditions. It should be noted that the nonlinear adaptive controller is straightforward to implement for two reasons:

1/ the form of the kinetics is the same for the two products; therefore, the same model for R_p is implemented;

2/ the parameters U and β in product one and product two change in the same manner. Therefore, the models from Product One adequately represent the parameter behavior when producing polymer two.

If the second point is not valid, different models will need to be formulated for the changing parameters in each Product.

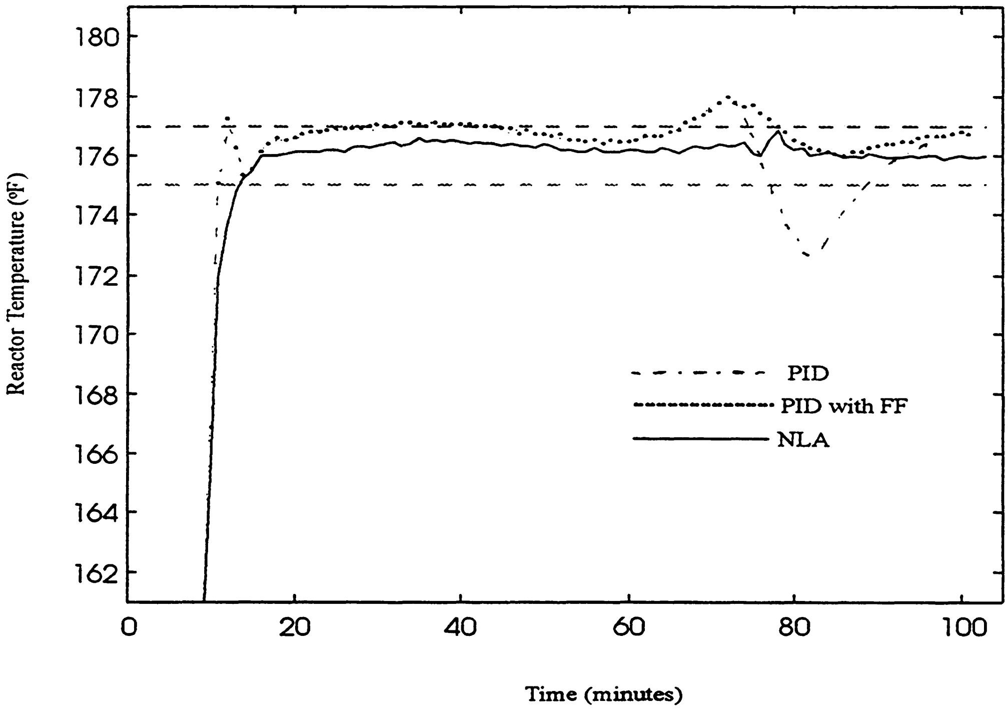


FIGURE 7.7: PERFORMANCE OF CONTROLLERS FOR PRODUCT TWO, BATCH ONE

In summary, Product Two considerations have been discussed.

7.6.2 MULTIPLE FEED PERIODS

In the recipe for product two calls for two feed periods. Due to process equipment limitations, only one is shown in the product two simulation. The existence of multiple feed periods introduces further complications for all the controllers. A brief discussion for each algorithm is included below.

For the PID feedback controllers, it is expected that new tuning rules will not be required for each feed period. However, the implemented tuning must give good control for both feeds, at the expense of giving better control for each individual period. Therefore, it is expected that the PID feedback algorithms will show a degradation of control.

The feedforward algorithm may need to be re-tuned for the heat released in each feed period and for the disturbance of each feed stop. Therefore, for two feed periods, four separate transfer functions may be required, depending on the flowrate and monomer type of each feed.

The nonlinear controller handles different feed periods easily because of its general nature. The estimation algorithm, however, may require specialization. If the updated parameters change in the same manner in each feed period, the estimation algorithm presented in the thesis will perform well. However, if the updated parameters exhibit different behavior during different feed periods, new models are required. This type of situation reduces the general nature of the nonlinear adaptive controller. Regardless of the models, the estimation will need to be reset at the beginning of each new feed. For the most part, this can be done by setting the initial parameter values (for example β_0) to the last estimated value (estimated β from the last feed period).

In summary, a brief discussion has raised some of the issues associated with multiple feed periods.

CHAPTER 8

SUMMARY

The temperature control of an 'industrial challenge' semi-batch reactor has been considered. Solutions evaluated include a well tuned PID, an adaptive PID, a PID with feedforward, a nonlinear controller and a nonlinear adaptive controller. Some of the key results are summarized below:

- in the family of PID controllers, the PID with feedforward compensation provides the best temperature control;
- the nonlinear controller with all parameters known outperforms the PID with feedforward compensation;
- using constant, average values for time-varying parameters in the nonlinear controller results in very poor control; this illustrates the need for precise process knowledge to achieve benefits from nonlinear control;
- some 'knowledge about what you don't know' is required for good performance of the extended Kalman filter: models for the unknown parameters must be chosen intelligently;
- the nonlinear adaptive controller outperforms the PID with feedforward compensation; it is comparable to the nonlinear controller with all parameters known (estimation does not degrade control);
- the nonlinear adaptive controller gives good control performance for a range of operating conditions. The PID class controllers require retuning in order to provide consistent temperature control.

For systems in which one set of operating conditions is consistently repeated, excellent improvements in control are possible with a PID class controller. Recall that for Product One, batch one conditions, the PID with feedforward provided acceptable control with significantly less process information that the nonlinear adaptive controller. Under these circumstances, there may be little incentive (or need) to apply a complex, model based controller.

However, for multi-batch, multi-product systems, the nonlinear adaptive controller shows more promise. In such situations, controllers such as the PID with feedforward compensation require retuning for each case. Empirical model based controllers would require model identification for each situation. The nonlinear adaptive controller structure, however, is automatically modified by changing process conditions to achieve a specific closed loop response.

CHAPTER 9

REFERENCES

Adebekun, D.K. and F.J. Schork, "Continuous Solution Polymerization Reactor Control I. Nonlinear Reference Control of Methyl Methacrylate", Ind. Eng. Chem. Res., **28**, 1308-1324 (1989).

Alvarez, J., R. Suarez and A. Sanchez, "Nonlinear Decoupling Control of Free-Radical Polymerization Continuous Stirred Tank Reactors", Chem. Eng. Sci, **45**, 3341-3357 (1990).

Alvarez, J., E. Hernandez and R. Suarez, "Nonlinear Control of a Two-Input, Three State Polymerization Reactor" in "Proc. of Amer. Contr. Conf.", Atlanta, GA (1988), pp 233-239.

Alvarez, J., J. Alvarez and E Gonzalez, "Global Nonlinear Control of a Continuous Stirred Tank Reactor", Chem. Eng. Sci., **44**, 1147-1160 (1989).

Alvarez, J., R. Suarez and R. Martinez, "Feedforward -feedback Nonlinear Control for Linearizable Systems" in "Proc. of Amer. Contr. Conf.", June 26-28, Boston , MA(1991), pp 1800-1805.

Balchen, J.G., B. Lie and I. Solberg, "Internal Decoupling in Non-linear Process Control", Mod. Idnt. and Cntrl., **9**, 137-148 (1988).

Bequette, B.W., "Nonlinear Control of Chemical Processes: a Review", Ind. Eng. Chem. Res., **30**, 1391-1413 (1991).

Calvet, J. and Y. Arkun, "Feedforward and Feedback Linearization of Nonlinear Systems with Disturbances", Int. J. Contr., **48**, 1551-1559 (1988).

Chien, D.C. H and A. Penlidis, "Effect of Impurities on Continuous Solution Methyl Methacrylate Polymerization Reactors: II. Closed Loop Real Time Control", Chem. Eng. Sci., **49**, 1855-1868 (1994).

Chien, D.C.H and A. Penlidis, "Effect of Impurities on Continuous Solution Methyl Methacrylate Polymerization Reactors: I. Open-loop Process Indentification Results", Chem. Eng. Sci., Poly. Reac. Eng., **2**, 163-213 (1994).

Chylla, R.W. and D.R. Haase, "Temperature Control of Semi-Batch Polymerization Reactors", Comp. Chem. Eng., **17**, 257-264 (1993), with corrected updates.

Chylla, R.W, personal correspondence, July 1994.

Cluett, W.R., S.L. Shah and D.G. Fisher, "Adaptive Control of a Batch Reactor", Chem. Eng. Commun., **38**, 67-78 (1985).

Congalidis, Richards, Ray, "Feedforward/Feedback Control of a Copolymerization Reactor", AIChE J., **35**, 891-907 (1989).

Cott, B.J. and S. Macchietto, "Temperature Control of Exothermic Batch Reactors using Generic Model Control", Ind. Eng. Chem. Res., **28**, 1177-1184 (1989).

Crowe, C., personal correspondence, class notes, 1995.

Daoutidis, P., M. Soroush and C. Kravaris, "Feedforward/Feedback Control of Multivariable Nonlinear Processes", AIChE J., **36**, 1471-1484 (1990).

Davidson, R.S., "An Intelligent Temperature Controller for Jacket Reactors" in "Proc. of Amer. Contr. Conf.", Minneapolis, MN (1987), pp. 1380-1385.

Defaye, G., N. Regnier, J. Chabanon, L. Caralp and C. Vidal, "Adaptive Predictive Temperature Control of Semi-Batch Reactors", Chem. Eng. Sci, **48**, 3373-3382 (1993).

Dimitratos, J., G. Elicabe and C. Georgakis, "Control of Emulsion Polymerization", AIChE J., 40, 1993-2021, 1994.

Dittmar, R., Z. Ogonowski and K. Damert, "Predictive Control of a Nonlinear Open-Loop Unstable Polymerization Reactor", Chem. Eng. Sci., **46**, 2679-2689, (1991).

Dougherty, E.P., P.H. Westkaemper and R.S Wu, "Temperature Control in Semi-Batch Reactors" in "ACS Symp. Series 404", 1988, pp 478-489.

Farber, J.N. and B.E. Ydstie, "Adaptive Compensation for Large Disturbances in a Continuous Polymerization Reactor", Ind. Eng. Chem. Foundam., **25**, 350-356 (1986)

Gagnon, L. and J.F. MacGregor, "State Estimation for Continuous Emulsion Polymerization", CjChE, **69**, 648-656 (1991).

Gattu, G. and E. Zafirio, "Nonliner Quadratic Dynamic Matrix Control with State Estimation", Ind. Eng. Chem. Res., **31**, 1096-1104 (1992).

Ham, J.Y. and H. Rhee, "Modelling and Control of an LDPE Autoclave Reactor"in Proc. of PSE '94, May 31-June 3, Kyonngju, Korea (1994), pp1047-1053.

Henderson, L.S. and R.A. Cornejo, "Temperature Control of Continuous Bulk Styrene Polymerization Reactors and the Influence of Viscosity: an Analytical Study", Ind. Eng. Chem. Res., **28**, 1644-1653 (1989).

Hindalgo, P.M. and C.B. Brosilow, "Nonlinear Model Predictive Control of Styrene Polymerization at Unstable Operating Points", Comp. Chem. Eng., **14**, 481-494 (1990).

Houston, W.E. and F.J. Schork, "Adaptive Predictive Control of a Semi-Batch Polymerization Reaction", Poly. Proc. Eng., **5**, 119-144 (1992).

Inglis, M. P., W.R. Cluett and A. Penlidis, "Long Range Predictive Control of a Polymerization Reactor", CjChE, **69**, 120-129 (1991).

Jutan, A. and A. Uppal, "Combined Feedforward/Feedback Servo Control Scheme for an Exothermic Batch Reactor", Ind. Eng. Chem. Proc. Des. Dev., **23**, 597-602 (1984).

Kravaris, C., R.A. Wright and J.F. Carrier, "Nonlinear Control for Trajectory Tracking in Batch Processes", Comp. Chem. Eng., **13**, 73-82 (1989).

Kwalik, K.M. and F.J. Schork, "Adaptive Control of a Continuous Polymerization Reactor" in "Proc. of Amer. Cont. Conf.", June 19-21, Boston, MA (1985), pp 872-877.

Lee, K.S, S.H. Bang, S. Yi, J.S. Son and S.C. Yoon, "Iterative Learning Control of Heatup Phase for a Batch Polymerization Reactor" in "Proc. of PSE '94", May 31-June 3, Kyonngju, Korea (1994), pp1203-1208.

Lie, B. and J.G. Balchen, "A Comparison of Strategies for the Control of a Polypropene Reactor" in "3rd IFAC Symp. on DYCORD, Maryland (1992), pp 265-270.

MacGregor, J.F., "On-line Reactor Energy Balances via Kalman Filtering" in "Proc. IFAC PRP-6 Automation", Akron, OH, Oct. Perganmon Press, Oxford (1986).

Marlin, T.E, "Process Control: Design of Processes and Control Systems for Dynamic Performance", McGraw-Hill Book Company, New York, NY (1995).

McAuley, K.B. and J.F. MacGregor, "Nonlinear Product Property Control in Industrial Gas Phase Polyethylene Reactors", AIChE J., 39, 855-866, 1993.

McLellan, P.J., T.J. Harris and D.W. Bacon, "Error Trajectory Descriptions of Nonlinear Controller Designs", Chem. Eng. Sci., **45**, 3017-3034 (1990).

Mendoza-Bustos, S.A., A. Penlidis and W. R. Cluett, "Adaptive Control of Conversion in a Simulated Solution Polymerization Continuous Stirred Tank Reactor", Ind. Chem. Eng. Res., **29**, 82-89 (1990).

Merkle J.E. and W.K. Lee, "Adaptive Strategies for Automatic Startup and Temperature Control of a Batch Process", Comp. Chem. Eng., **13**, 87-103 (1989).

Papadoulis, A.V., C.A. Tsiligiannis and S.A. Svoronos, "A Cautious Self Tuning Controller for Chemical Processes", AIChE J., **33**, 401-409 (1987).

Peterson, T., E. Hernandez, Y. Arkun and F.J. Schork, "A Nonlinear DMC Algorithm and its Application to a Semi-Batch Polymerization Reactor", Chem. Eng. Sci, **47**, 737-753, (1992).

Prasad, P.R., P.B Deshpande, K.M. Kwalik, F.J. Schork and K.W. Leffew, "Multi-loop Control of a Polymerization Reactor", Poly. Eng. and Sci.,**30**, 350-354 (1990).

Seborg, D.E., T.F. Edgar, S.L. Shah, "Adaptive Control Strategies for Process Control: A Survey", AIChE J., **32**, 881-913 (1986).

Singstad, P., H. Nordhus, K. Strand, M. Lien, L. Lyngmo and O. Moen, "Multivariable Nonlinear Control of Industrial LDPE Autoclave Reactors" in "Proc. of Amer. Contr. Conf.", June 24-26, Chicago, IL (1992), pp 615-619.

Smith, C.A. and A.B. Corripio, "Principles and Practice of Automatic Process Control", John Wiley & Sons, Inc. (1985)

Smith, J.M., and H.C. Van Ness, "Introduction to Chemcial Engineering Thermodynamics", McGraw-Hill Book Company, Toronto, Ont (1987)

Soroush, M. and C. Kravaris, "Multivariable Nonlinear Control of a Continuous Plymerization Reactor: an Experimental Study", AIChE J., **39**, 1920-1937 (1993).

Soroush, M. and C. Kravaris, "Nonlinear Control of a Batch Polymerization Reactor: an Experimental Study", AIChE J., **38**, 1429-1448 (1992).

Tzouanas, V.K. and S.L. Shah, “Adaptive Pole-assignment Control of a Batch Polymerization Reactor”, Chem. Eng. Sci, **44**, 1183-1193 (1989).

Wang, F. and Y. Lin, “Adaptive Pole Removal Control for a Batch Polymerization Reactor”, Chem. Eng. Tech., **14**, 240-247 (1991).

Wang, Z.L., J.P Corriou and F. Pla, “Nonlinear Adaptive Control of a Continuous Polymerization Reactor” in “IFAC Symposium ADCHEM ’94”, May 1994, Kyoto, Japan (1994), pp 513-518.

Wells, C.H., “Application of Modern Estimation and Identification Techniques to Chemical Processes”, AIChE J., **17**, 966-973, 1971.

Whatley, M.J. and D.C. Pott, Hydro. Pr., May, 75-78 (1984).

APPENDIX A

CORRECTIONS AND CLARIFICATIONS FOR THE SEMI-BATCH REACTOR

A.1 ENERGY BALANCE FOR THE SEMI-BATCH REACTOR

In Chylla and Haase(1993), an error was discovered in the derivation of the energy balance for the semi-batch reactor. The correct form is derived here and is used to simulate the behavior of the system. Notes from Crowe (1995), Marlin (1995) and Smith and Van Ness (1987) are used as a basis. Notation for this system may be found in Appendix C.

From Crowe (1995), Marlin (1995), the energy balance of the system may be written as:

$$\frac{d(\sum \rho_i u_i V_i)}{dt} = \rho_m u_m v_m + Q + W \qquad \text{(A.1)}$$

where: u_i = specific internal energy of component I (energy/mass)

Q = rate of heat addition/loss to system (energy/time)

W = rate of doing work on system (energy/time)

The work may be divided into components:

Flow Work (due to stream entering reactor only):

$$W_f = P_m v_m \tag{A.2}$$

where P is the pressure and v the volumetric flow (volume/time).

Expansion Work:

$$W_e = -P\frac{dV}{dt} \tag{A.3}$$

The remaining work done on the system is lumped into W_r. Therefore the total work is given by:

$$W = W_f + W_e + W_r \tag{A.4}$$

Substituting equations (A.2) and (A.3) into (A.1) and assuming that W_r is negligible:

$$\begin{aligned}\frac{d(\sum \rho_i u_i V_i)}{dt} &= \rho_m u_m v_m + Q + P_m v_m - P\frac{dV}{dt} \\ &= \rho_m v_m\left(u_m + \frac{P_m}{\rho_m}\right) + Q - P\frac{dV}{dt}\end{aligned} \tag{A.5}$$

Noting that specific enthalpy may be written as:

$$h_i = u_i + \frac{P_i}{\rho_i} \tag{A.6}$$

Replacing u$_i$ in equation (A.5) with (A.6) gives:

$$\frac{d\sum(\rho_i h_i - P_i)V_i}{dt} = \rho_m v_m h_m + Q - P\frac{dV}{dt} \tag{A.7}$$

and:

$$\frac{d\sum(p_i h_i V_i)}{dt} = \rho_m v_m h_m + Q + V\frac{dP}{dt} \tag{A.8}$$

In equation (A.8), it is assumed that the pressure of each of the components is equal and has a value P.

Assuming a constant system pressure and rewriting equation (A.8) in terms of mass instead of volume:

$$\frac{d\sum(m_i h_i)}{dt} = \dot{m}_m h_m + Q \tag{A.9}$$

The left hand side of equation (A.9) may be expanded:

$$\frac{d\sum(m_i h_i)}{dt} = \frac{d(m_w h_w)}{dt} + \frac{d(m_s h_s)}{dt} + \frac{d(m_m h_m)}{dt} \tag{A.10}$$

Using the chain rule for each component:

$$\frac{d(m_i h_i)}{dt} = m_i \frac{dh_i}{dt} + h_i \frac{dm_i}{dt}$$

$$= m_i \left(\frac{dh_i}{dT} \frac{dT}{dt} \right) + h_i \frac{dm_i}{dt} \qquad \text{(A.11)}$$

Noting, from Smith and Van Ness (1987):

$$C_{pi} = \frac{dh_i}{dT} \qquad \text{(A.12)}$$

Therefore, equation (A.11) becomes:

$$\frac{d(m_i h_i)}{dt} = m_i C_{pi} \frac{dT}{dt} + h_i \frac{dm_i}{dt} \qquad \text{(A.13)}$$

Substituting (A.13) into (A.10), the left hand side of equation (A.9) may therefore be written:

$$\frac{d\sum(m_i h_i)}{dt} = \sum(m_i C_{pi}) \frac{dT}{dt} + h_w \frac{dm_w}{dt} + h_s \frac{dm_s}{dt} + h_m \frac{dm_m}{dt} \qquad \text{(A.14)}$$

Incorporating mass balances for the monomer and the solids (polymer):

$$\frac{dm_m}{dt} = \dot{m}_m - R_p MW_m \qquad \text{(A.15)}$$

$$\frac{dm_s}{dt} = R_p MW_m \qquad \text{(A.16)}$$

$$\frac{dm_w}{dt} = 0 \tag{A.17}$$

Noting that for a constant heat capacity value (Smith and Van Ness):

$$h_i|T = h_i\Big|T_{ref} + C_{pi}(T - T_{ref}) \tag{A.18}$$

The reference temperature is a temperature at which the enthalpy of the component is known. Substituting equations (A.15) - (A.18) into (A.14):

$$\begin{aligned} \frac{d(\sum(m_i h_i))}{dt} &= \sum(m_i C_{pi})\frac{dT}{dt} + (h_m\Big|T_{ref} + C_{pm}(T - T_{ref}))(\dot{m}_m - R_p MW_m) \\ &+ (h_s\Big|T_{ref} + C_{ps}(T - T_{ref}))(R_p MW_m) \end{aligned} \tag{A.19}$$

Replacing the left hand side of equation (A.9) with (A.19):

$$\begin{aligned} &\sum(m_i C_{pi})\frac{dT}{dt} + (h_m\Big|T_{ref} + C_{pm}(T - T_{ref}))(\dot{m}_m - R_p MW_m) \\ &+ (h_s\Big|T_{ref} + C_{ps}(T - T_{ref}))(R_p MW_m) = \dot{m}(h_m\Big|T_{ref} + C_{pm}(T_{feed} - T_{ref})) + Q \end{aligned} \tag{A.20}$$

Expanding and rearranging:

$$\sum (m_i C_{pi}) \frac{dT}{dt} = -\overset{\bullet}{m}_m (h_m \big| T_{ref} + C_{pm}(T - T_{ref})) + R_p MW_m (h_m \big| T_{ref} + C_{pm}(T - T_{ref}))$$

$$- R_p MW_m (h_s \big| T_{ref} + C_{ps}(T - T_{ref})) + \overset{\bullet}{m}_m (h_m \big| T_{ref} + C_{pm}(T_{feed} - T_{ref})) + Q \qquad \text{(A.21)}$$

Canceling terms and rearranging again:

$$\sum (m_i C_{pi}) \frac{dT}{dt} = \overset{\bullet}{m}_m C_{pm}(T_{feed} - T)) + R_p MW_m (h_m \big| T_{ref} - h_s \big| T_{ref})$$

$$+ R_p MW_m (C_{pm} - C_{ps})(T - T_{ref}) + Q \qquad \text{(A.22)}$$

Equation (A.22) still contains enthalpy terms (h_m and h_s). However, noting that the difference in enthalpy between the reactants (monomer) and products (solids) is the heat of reaction:

$$(h_m \big| T_{ref} - h_s \big| T_{ref}) = (-\Delta H_r) \big| T_{ref} \qquad \text{(A.23)}$$

where the units of ($-\Delta H_r$) are energy/mass.

As well, the heat to the reactor, Q, can be broken into its components, resulting in:

$$\sum (m_i C_{pi}) \frac{dT}{dt} = \overset{\bullet}{m}_m C_{pm}(T_{feed} - T) + R_p MW_m (-\Delta H_r) \big| T_{ref} + R_p MW_m (C_{pm} - C_{ps})(T - T_{ref})$$
$$- UA(T - T_{jout}) - UA_{loss}(T - T_{amb}) \qquad \text{(A.24)}$$

Note that in equation (A.24), the jacket is assumed to be well mixed. Therefore, the jacket temperature can be represented by T_{jout} in the expression for Q. This assumption is valid due to the small temperature difference between T_{jin} and T_{jout} (MacGregor, 1986).

The jacket energy balance in Chylla and Haase (1993) is also modified to included the same expression for Q (the original expression represented the jacket temperature as T_{javg}). The modifications were made in order to maintain consistency with the author's implicit assumption of well mixed behavior in the jacket energy balance (where the enthalpy of the jacket contents is expressed as a function of T_{jout}). It should be pointed out that the changes to the expression Q do not (noticeably) affect the results, again because of the small temperature difference between T_{jin} and T_{jout}.

Finally, the heat of reaction at T can be expressed as:

$$\begin{aligned}(\Delta H_r)|T &= C_{pm}(T_{ref}-T)+(\Delta H_r)\big|T_{ref}+C_{ps}(T-T_{ref}) \\ &= (\Delta H_r)\big|T_{ref}-(C_{pm}-C_{ps})(T-T_{ref})\end{aligned} \tag{A.25}$$

The final form of the energy equation is:

$$\sum(m_iC_{pi})\frac{dT}{dt}=\dot{m}_m\,C_{pm}(T_{feed}-T)+R_pMW_m(-\Delta H_r)|T-UA(T-T_{jout})-UA_{loss}(T-T_{amb}) \tag{A.26}$$

Equation (A.26) is used in the matlab files to simulate the semi-batch reactor behavior. In the Chylla Haase reactor, ($-\Delta H_r$) is given in units of energy/mole therefore the monomer molecular weight is not included in the simulation. Note also that in the process simulation, temperature sensor dynamics are assumed negligible.

A.2 Implementation of Slave Controller

In Chylla and Haase (1993), the following equation describes the split range controller for heating and cooling:

$$\begin{array}{ll} 0 \le c(t) \le 49: & \textit{cooling} \\ 50: & \textit{no heating or cooling} \\ 51 \le c(t) \le 100: & \textit{heating} \end{array} \tag{A.27}$$

From equation (A.27), the controller output is not specified between 49 and 50, and between 50 and 51. The equation that is actually implemented is:

$$\begin{array}{ll} 0 \le c(t) < 50: & \textit{cooling} \\ 50: & \textit{no heating or cooling} \\ 50 < c(t) \le 100: & \textit{heating} \end{array} \tag{A.28}$$

A.3 Coolant Recirculation Loop

Figure 1.1 presents a schematic of the process equipment and control system that differs slightly from Chylla and Haase (1993). In the original publication, it is stated that the dump value is manipulated for cooling. The cooling valve is adjusted by a pressure regulator to exactly compensate for the volume of water leaving the coolant recirculation loop. Steam is injected directly into the loop to provide heating.

Unfortunately, the recirculation loop does not material balance and no allowance for the increase in volume (and hence pressure) of the system is included in the model equations. A more likely setup (and one corresponding to the published equations) is presented in Figure 1.1. In this figure, the slave controller manipulates the cold water and steam valves directly. The dump valve is adjusted by a pressure regulator and removes coolant when either the cold water or the steam is being injected into the loop.

APPENDIX B
NOTATION

Names and units of variables referred to during the thesis are included in this appendix.

m_m: Mass of monomer in reactor (lb)
m_w: Mass of water in reactor (lb)
m_s: Mass of polymer (solids) in the reactor (lb)
m_c: Mass of coolant in jacket (lb)
C_{pm}: Specific heat of monomer (Btu/lb °F)
C_{pw}: Specific heat of water (Btu/lb °F)
C_{ps}: Specific heat of polymer (Btu/lb °F)
C_{pc}: Specific heat of coolant (Btu/lb °F)
T: Reactor temperature (°F)
T_{jout}: Jacket outlet temperature (°F)
T_{jin}: Jacket inlet temperature (°F)
T_{amb}: Ambient temperature (°F)
T_{feed}: Temperature of feed (equal to ambient temperature) (°F)
T_{ref}: Reference temperature (°F)
U: Overall heat transfer coefficient (Btu/(ft^2 °F hr)
A: Heat transfer area (ft^2)
k_0: Pre-exponential factor (min^{-1})
n_m: Moles of monomer in reactor (lbmol)
μ: Viscosity of reactor contents (cP)
α: Kinetic constant (activation energy/R) (°F)
$(-\Delta H_r)$: Heat of reaction (Btu/lbmol)
R_p: rate of reaction (lbmol/min)
MW_m: Molecular weight of monomer (lb/lbmol)
h: Enthalpy (Btu/lb)
UA_{loss}: Heat loss coefficient (Btu/°F)
$\dot{m}_m$: Monomer feedrate (lb/min)
$\dot{m}_c$: Coolant flow (lb/min)

t: time (min)
Q_r: Heat released due to reaction (Btu/min)

SUBSCRIPTS:
$_m$: monomer
$_w$: water
$_s$: polymer (solids)
$_c$: coolant
$_{sp}$: setpoint
$_{op}$: operating point

GENERAL NOTATION:
$\underline{x}$: vector of states
u: input
y: measurement (Kalman filter) or controlled variable (nonlinear control)
$\dot{x}$: time derivative (dx/dt)
$\ddot{x}$: second time derivative (d^2x/dt^2)

APPENDIX C
OBTAINING VALUES FOR REQUIRED PROCESS PARAMETERS

Although some of the process parameters may not be currently known for a process, it is possible to get good estimates of the values if some simple research is conducted and common sense applied. This section outlines some methods that could be used to get quick estimates of the unknown parameters.

$\Sigma(m_iC_{pi})$

For this parameter, both specific heat data and mass information is required. One option for evaluating this expression is to consider the initial and final values. Initially, there is only water and polymer, according to the recipe:

$$\Sigma(m_iC_{pi})_0 = (m_wC_{pw} + m_sC_{ps})_0$$

The mass of water and polymer are known because the quantities are specified by the recipe. The specific heat of water can be found in standard property books. The specific heat of the polymer may be known from product specifications, or a good estimate can be made using water and a few other typical compounds as references. At the end of the batch, there is only water and polymer again:

$$\Sigma(m_iC_{pi})_f = (m_wC_{pw} + m_sC_{ps})_f$$

The mass of water is the same since only pure monomer is fed to the reactor. The final mass of polymer is the sum of the initial mass of polymer and the mass of monomer fed (assuming all monomer is reacted). Knowing how long the monomer is typically fed and the feedrate, the mass of at the end of the batch can be estimated. The value of $\Sigma(m_iC_{pi})$ during the batch can be interpolated between the initial and final values. For the Chylla Haase system, a linear interpolation gives good estimates. The value can be parameterized as a function of mass of monomer fed, or a lookup table can be programmed into the controller.

$\dot{m}_m$

The feedrate of monomer should be associated with a flow measurement. If it is not, the recipe value can be inserted into the process model when the feed is 'on'.

C_{pm}

The monomer specific heat may be known from supplier data or physical tables. It could also be estimated based on the specific heat of water and other reference compounds.

T_{feed}

The temperature of the feed should be associated with a temperature measurement.

A

The heat transfer area can be calculated with a level measurement and the dimensions of the reactor (these dimensions should be available design knowledge).

$\dot{m}_c$

The feedrate of the coolant to the jacket should be associated with a flow measurement.

C_{pc}

Since the coolant is water, this value is easily obtained.

m_c

The mass of coolant in the jacket should be available design knowledge.

$e^{-\alpha/T}$

The kinetic parameter α is required to evaluate the exponential. It is {activation energy of the reaction/R} in the appropriate units. It may be available in proprietary kinetic information, or an rough estimate could be possible from reference reactions.

ΔH_r

The heat of reaction may also be available in proprietary kinetic data, or a rough estimate may be possible using other reactions (whose data is available) as a reference.

UA_{loss}

A rough estimate of this value is usually available from experience. However, once again, a literature survey may uncover guidelines for specifying a value based on reactor size or volume.

T_{amb}

This value may be associated with a temperature measurement, or average values for a season or time of day may be summarized in a look up table in the controller.