A MULTIPLE OBJECTIVE OPTIMIZATION APPROACH TO QUALITY CONTROL

NAGW-1333

by

Christopher Michael Seaman

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering
Troy, New York 12180-3590

April 1991

CIRSSE REPORT #89

CONTENTS

LIST OF FIGURES vi

LIST OF TABLES xii

DEDICATION xiii

ACKNOWLEDGEMENT xiv

ABSTRACT xv

1. Introduction 1

    1.1 Motivation and Objectives 1

    1.2 Contributions 2

    1.3 Application to the Injection Molding Process 3

    1.4 Organization of the Thesis 5

2. The Injection Molding Process and Part Quality 9

    2.1 Introduction 9

    2.2 A Review of Injection Molding Research 10

        2.2.1 Research in Modeling and Control 11

        2.2.2 Research on the Effect of Microstructure on Mechanical Properties 14

        2.2.3 Research on the Effect of Processing on Mechanical Properties 14

    2.3 Preliminary Experiments 16

        2.3.1 Sensitivity of Melt Temperature to Plastication Inputs 17

        2.3.2 Part Quality Dependence upon Processing Conditions 30

        2.3.3 Sensitivity of Machine Dynamics to Operating Conditions . . 42

    2.4 Summary 48

3. Proposed Approach to the Quality Control Problem 49

    3.1 Introduction 49

    3.2 Definition of the Quality Control Problem 50

        3.2.1 Performance Criteria 50

        3.2.2 Inputs and Outputs 52

3.2.3 The System Model 56
3.2.4 A Control Policy 58
3.3 A Review of the Multiple-Objective Optimization Problem 59
3.3.1 Finding the Tradeoff Surface 62
3.4 A Linear Programming Solution to Multiobjective Quality Control . . 64
3.5 Implementation of Automatic Quality Control 66
3.5.1 A Proposed Feasible Directions Algorithm 67
3.5.2 Properties of this Algorithm 71
3.5.3 Example 73
3.6 Summary 74
4. Experimental Setup 76
4.1 The Injection Molding Machine 76
4.1.1 Actuators 76
4.1.2 Process Sensors 79
4.2 The Control Computer 80
5. Multiobjective Controller Tuning 84
5.1 Plastication Modeling and Identification 87
5.1.1 Analytical Modeling of Plastication 87
5.1.2 Identification of a Plastication Model 91
5.2 Plastication Controller Tuning: Simulation Results 95
5.2.1 Simulation Conditions 95
5.2.2 Discussion of the Simulation Tuning Results 96
5.3 Plastication Controller Tuning: Experimental Results 100
5.3.1 Discussion of the Experimental Tuning Results 101
5.4 Injection Controller Tuning 110
5.4.1 Analytical Modeling of Injection 110
5.4.2 Description of the Injection Controller Tuning Experiments . . 111
5.4.3 Experimental Results 113
5.4.4 Discussion 121
5.5 Conclusions 123

6. Tuning the Process to Achieve Quality Objectives 125
6.1 Spiral Mold Quality Tuning 125
6.1.1 Description of the Spiral Mold Quality Tuning Experiments . 125
6.1.2 Experimental Results 130
6.1.3 Discussion 135
6.2 ASTM Four Cavity Mold Quality Tuning 136
6.2.1 Experimental Conditions 138
6.2.2 Discussion of the Tuning Results 138
6.3 Conclusions 150
7. Conclusions 152
7.1 Contributions 154
7.2 Future Work 155
7.2.1 Improving Algorithm Efficiency 155
7.2.2 Improving the Decision Maker's Interface 157
8. Literature Cited 159
A. The Injection Molding Process 167
A.1 The Injection Molding Cycle 167
A.2 Plastication 169
A.2.1 Polymer Melting (Thermodynamics) 169
A.2.2 Polymer Pumping (Fluid Dynamics) 179
A.2.3 Overall Polymer Pumping Dynamics 183
A.2.4 Overall Plastication Dynamics 185
A.3 Injection 185
A.3.1 Injection Polymer Flow Dynamics 187
A.3.2 Overall Injection Dynamic Model 189
A.4 Holding 190
A.4.1 Holding Polymer Mass Balance 191
A.4.2 Overall Holding Dynamic Model 192
A.5 Cooling 193
A.6 Process Control and Phase Interaction 194
A.7 Summary of Previous Research 196
A.8 Summary 197

B. Symbol Table and Nomenclature for Machine Dynamics 198
B.1 Definition of the Quality Control Problem 198
B.2 A Review of the Multiobjective Optimization Problem 199
B.3 A Linear Programming Solution to Multiobjective Quality Control . . 199
B.4 Polymer Melting Due to Conduction Heating 199
B.5 Polymer Melting Due to Viscous Dissipation 200
B.6 Overall Melting Dynamics 200
B.7 Polymer Fluid Dynamics 201
B.8 Injection Dynamics 202
B.9 Holding Dynamics 203
B.10 Cooling Dynamics 203
C. An Example of Approximation Programming 204
C.1 Example 204

LIST OF FIGURES

Figure 1.1 Cut-away View of a Reciprocating Screw Injection Molding Machine 6

Figure 1.2 The Injection Molding Phases 6

Figure 1.3 Injection Molding Phases and Interactions 7

Figure 2.1 Preparation of Specimens with Two Different Orientations . . 15

Figure 2.2 The relationship between melt temperature and melt pressure, at three different barrel temperatures, for polystyrene. 19

Figure 2.3 Melt temperature error versus melt pressure for polystyrene. . 19

Figure 2.4 Melt temperature versus recovery rate for polystyrene. 20

Figure 2.5 Recovery rate versus back pressure for polystyrene. 20

Figure 2.6 Melt temperature versus plastication time for polystyrene. . . 22

Figure 2.7 Melt temperature error versus plastication time for polystyrene. 22

Figure 2.8 Estimated versus actual temperature using a three parameter model. 23

Figure 2.9 Estimated versus actual temperature using a two parameter model. 23

Figure 2.10 Polypropylene melt temperature versus back pressure for four different barrel temperatures at $200RPM$. 25

Figure 2.11 Polypropylene melt temperature versus back pressure for four different barrel temperatures at $300RPM$. 25

Figure 2.12 Polypropylene melt temperature error versus back pressure for four different barrel temperatures at $200RPM$. 26

Figure 2.13 Polypropylene melt temperature error versus back pressure for four different barrel temperatures at $300RPM$. 26

Figure 2.14 Back pressure versus recovery rate: upper curve is $300RPM$ data, lower curve is $200RPM$ data. 27

Figure 2.15 Polypropylene melt temperature versus plastication time for four different barrel temperatures at $200RPM$. 28

Figure 2.16 Polypropylene melt temperature versus plastication time for four different barrel temperatures at $300RPM$. 28

Figure 2.17 Estimated temperature versus actual temperature for polypropylene at $200RPM$. 29

Figure 2.18 Estimated temperature versus actual temperature for polypropylene at $300RPM$. 29

Figure 2.19 Melt temperature versus position in the shot at four different back pressures. 31

Figure 2.20 ASTM test mold used for part quality experiments. 33

Figure 2.21 Velocity profiles used in the quality experiments. 35

Figure 2.22 Histogram of parallel dimension data for the initial experiment 37

Figure 2.23 Histogram of perpendicular dimension data for the initial experiment 37

Figure 2.24 Histogram of parallel dimension data for the experiment using the optimal inputs. 39

Figure 2.25 Histogram of perpendicular dimension data for the experiment using the optimal inputs. 39

Figure 2.26 Histogram of parallel dimension data for the experiment using the adjusted optimal inputs. 40

Figure 2.27 Histogram of perpendicular dimension data for the experiment using the adjusted optimal inputs. 40

Figure 2.28 Melt pressure response at a setpoint of $5000PSI$. 43

Figure 2.29 Melt pressure response at a setpoint of $7000PSI$. 44

Figure 2.30 Melt pressure response at a setpoint of $3000PSI$. 44

Figure 2.31 Overshoot for 50 runs from a cold start. 45

Figure 2.32 Steady-state error for 50 runs from a cold start. 45

Figure 2.33 Plot showing evolution of the plastication step response over time. 47

Figure 2.34 Plot showing evolution of the hydraulic oil temperature over time. 47

Figure 3.1 Variability in the Manufacturing Process 53

Figure 3.2 Two dimensional example showing how the quadratic constraint is approximated by linear constraints. 66

Figure 3.3 Complete conflict versus ϵ–conflict. 70

Figure 3.4 Contour plot of objective functions showing algorithm iterations. 75

Figure 5.1 The algorithm used for performance monitoring and control. . 86

Figure 5.2 PID controller block diagram. 86

Figure 5.3 Plastication dynamics block diagram. 90

Figure 5.4 Estimated poles and zeros with confidence intervals. 93

Figure 5.5 Process input produced by relay controller. 94

Figure 5.6 Comparison of process output and simulation output (simulation is dashed-line). 94

Figure 5.7 The tradeoff boundary (nondominated set) found using the multiobjective algorithm. 98

Figure 5.8 Efficient points: K_I vs. K_P. 98

Figure 5.9 Efficient points: K_D vs. K_P. 99

Figure 5.10 Efficient points: K_D vs. K_I. 99

Figure 5.11 A comparison of the experimentally determined tradeoff boundary (x) with the "simulation" boundary (o). 104

Figure 5.12 Comparison of efficient points: K_I vs. K_P 104

Figure 5.13 Comparison of efficient points: K_P vs. K_D 105

Figure 5.14 Comparison of efficient points: K_I vs. K_D 105

Figure 5.15 Control effort using initial gains. 106

Figure 5.16 Process response using initial gains. 106

Figure 5.17 Control effort at tradeoff surface point **A** 107

Figure 5.18 Process response at tradeoff surface point **A**. 107

Figure 5.19 Control effort at tradeoff surface point **B**. 108

Figure 5.20 Process response at tradeoff surface point **B**. 108

Figure 5.21 Control effort at tradeoff surface point **C**. 109

Figure 5.22 Process response at tradeoff surface point **C**. 109

Figure 5.23 Injection dynamics block diagram. 113

Figure 5.24 Nondominated set of ITAE versus RMSE. 116

Figure 5.25 Nondominated set of ITAE versus maximum control deviation. 116

Figure 5.26 Nondominated set of maximum control deviation versus RMSE. 117

Figure 5.27 Efficient points: K_P versus K_I 117

Figure 5.28 Efficient points: K_P versus K_D 118

Figure 5.29 Efficient points: K_I versus K_D 118

Figure 5.30 Melt pressure output at nondominated point **A**. The setpoint is 2500 PSI. 119

Figure 5.31 Servovalve input at nondominated point **A**. 119

Figure 5.32 Melt pressure output at nondominated point **G**. The setpoint is 2500 PSI. 120

Figure 5.33 Servovalve input at nondominated point **G**. 120

Figure 6.1 The spiral part. 126

Figure 6.2 Plastication inputs for spiral mold optimization. 131

Figure 6.3 Injection inputs for spiral mold optimization. 131

Figure 6.4 Cycle time as the algorithm progressed. 132

Figure 6.5 Spiral length as the algorithm progressed. 132

Figure 6.6 Nondominated set after first line search. 133

Figure 6.7 Efficient points after first line search 133

Figure 6.8 Comparison of first (o) and second (*) nondominated sets. . . 134

Figure 6.9 Efficient points for first (o) and second (*) nondominated sets. 134

Figure 6.10 The ASTM test part. 137

Figure 6.11 Nondominated set of cycle time versus flash, after the first algorithm iteration. 141

Figure 6.12 Nondominated set of underfill versus flash, after the first algorithm iteration. 141

Figure 6.13 Nondominated set of cycle time versus underfill, after the first algorithm iteration. 142

Figure 6.14 Efficient plastication and injection setpoints for the nondominated set. 142

Figure 6.15 Nondominated set of cycle time versus flash, after the first algorithm iteration. 143

Figure 6.16 Efficient plastication and injection setpoints for the nondominated set. 143

Figure 6.17 Nondominated set of cycle time versus flash, after the second algorithm iteration, under the new setpoint constraint. 146

Figure 6.18 Nondominated set of underfill versus flash, after the second algorithm iteration, under the new setpoint constraint. 146

Figure 6.19 Nondominated set of cycle time versus underfill, after the second algorithm iteration, under the new setpoint constraint. . . 147

Figure 6.20 Efficient plastication and injection setpoints for the nondominated set, under the new setpoint constraint. 147

Figure 6.21 Final nondominated set of cycle time versus flash. 148

Figure 6.22 Final efficient plastication and injection setpoints. 148

Figure 6.23 Plastication inputs for ASTM four cavity mold optimization. . 149

Figure 6.24 Injection inputs for ASTM four cavity mold optimization. . . 149

Figure A.1 Cross Section of One Screw Flight During Plastication 170

Figure A.2 Two Parallel Plates Separated by a Newtonian Fluid 173

Figure A.3 The Melt Temperature Dynamic System 178

Figure A.4 Pumping Dynamics Block Diagram 184

Figure A.5 Plastication Dynamics 185

Figure A.6 Injection Dynamics Block Diagram 190

Figure A.7 Holding Dynamics 193

LIST OF TABLES

Table 2.1 Quality Categories and Components 11

Table 2.2 Inputs used for the quality experiments. 41

Table 2.3 Statistics of the dimensional data. 41

Table 2.4 Flash output for the three experiments. 41

Table 4.1 Arburg Allrounder Machine Specifications 77

Table 4.2 Solenoid Valve Activation Table 77

Table 5.1 Symbols used in the plastication dynamic equations. 89

Table 5.2 Nominal Input Settings 101

Table 5.3 Symbols used in the injection dynamic equations. 112

Table 5.4 Nominal Input Settings 114

Table 6.1 A Sequence of Randomized Runs. 128

Table 6.2 Nominal Input Settings for the Spiral Mold 129

Table 6.3 Nominal Input Settings for the ASTM Mold 139

DEDICATION

This work is dedicated to my wife Elaine, for her love and support, and to my son Gregory, who may never know how much he's taught me.

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Marion Waltz and Mr. B.P. Wilkerson, of Alcoa, for their support of this research.

I would like to thank Dr. Alan Desrochers for challenging me to do my best. Also, I would like to thank Dr. George List for his valuable insight and inspiration.

Finally, I would like to thank Steve Murphy for his advice and friendship, Kurt Redlitz for quality, custom computer work, Peter Gennes for his injection molding expertise, and Kostas Kyriakopoulos, Deepak Sood, and the other denizens of RAL and CIRSSE.

ABSTRACT

In this work, the use of product quality as the performance criteria for manufacturing system control will be explored. The goal in manufacturing, for economic reasons, is to optimize product quality. The problem is that since quality is a rather nebulous product characteristic, there is seldom an analytic function that can be used as a measure. Therefore standard control approaches, such as optimal control, cannot readily be applied.

A second problem with optimizing product quality is that it is typically measured along many dimensions: there are many aspects of quality which must be optimized simultaneously. Very often these different aspects are incommensurate and competing. The concept of optimality must now include accepting tradeoffs among the different quality characteristics.

These problems are addressed using multiple objective optimization. In this thesis, it is shown that the quality control problem can be defined as a multiple objective optimization problem. A controller structure is defined using this as the basis. Then, an algorithm is presented which can be used by an operator to interactively find the *best* operating point.

Essentially, the algorithm uses process data to provide the operator with two pieces of information:

- If it is possible to simultaneously improve all quality criteria, then determine what changes to the process input or controller parameters should be made to do this;
- If it is not possible to improve all criteria, and the current operating point is not a desirable one, select a criteria in which a tradeoff should be made, and make input changes to improve all other criteria.

The process is not operating at an optimal point in any sense if, no tradeoff has to be made to move to a new operating point. This algorithm ensures that operating points are optimal in some sense and provides the operator with information about tradeoffs when seeking the best operating point.

This approach is new because it is an *integration* of quality control and automatic control. Quality control provides a way of *monitoring* process quality but does not specifically address the need to react to process changes so as to maintain or improve the level of quality. This is typically left up to a process expert. Automatic control provides a way to methodically react to output deviations, decreasing the variability of the process, but it is assumed that there is an accurate analytical model of the process. This can be a very limiting assumption, especially when addressing complex process control problems where the underlying models are nonlinear and time-varying.

In this work, the multiobjective algorithm was implemented in two different injection molding scenarios: tuning of process controllers to meet specified performance objectives and tuning of process inputs to meet specified quality objectives. Five case studies are presented.

1. Introduction

1.1 Motivation and Objectives

In manufacturing, the ultimate measure of any product is quality. Quality is "the totality of features and characteristics of a product ... that bears on its ability to satisfy given needs" [1]. A product being of "good" quality implies that there exist measures for the characteristics and that predefined levels for these measures are being met or exceeded by the product. The goal in manufacturing, then, is to *control* this quality.

Quality control is usually the responsibility of the operator. He makes subjective decisions based on observations and experience. One aspect of the problem which makes the decision making process difficult is that quality is made up of many characteristics that may be incommensurate and competing. The operator will have to make tradeoffs amongst the criteria so that all of them are satisfied simultaneously and the process is operating at the "best" point. One problem with operator control is that the level of quality control which is achieved depends upon the expertise of a given operator.

The goal is to control quality *automatically*. To do this, the process of making tradeoffs must be dealt with. It is proposed that the quality control problem be formulated as a multiple objective optimization problem. A motivation for this approach is that any important quality characteristic can be represented by an appropriate cost function. Optimization takes place with respect to the *vector-valued* cost function, of which each quality characteristic is an element. In this way, making tradeoffs among quality characteristics is an explicit part of the decision process, rather than being hidden in some aggregate cost function. Furthermore, the operator's role changes from one of *monitoring and operating the process* to one

of *monitoring and improving quality.*

This approach goes beyond the typical methods of quality control, which consist mostly of charting measurements. In these methods, the charts are used to *detect* process variations. Any decisions regarding control actions are left to the process operator. His experience and intuition are the guide. Conversely, an *automatic* controller, based on a process model, will predictably and methodically control the process. An added advantage is that because this method is optimization-based, improved quality performance may result over that obtained by a human operator alone.

1.2 Contributions

In this thesis, a method for automatic control of quality is presented. This method is based on building a quality–based model of the process. For this model, output measurements are part quality characteristics, and inputs are process command signals and parameters of process controllers. This model is then used within a multiobjective optimization algorithm, the quality controller. Essentially the quality controller iteratively *tunes* or *adapts* the process inputs and controllers on–line, so that quality is continuously optimized.

This approach is new in several ways. First, automatic controllers are usually designed based on outputs which represent some physical measurement of the process. For example, temperature is controlled to a setpoint, and the controller is judged on how tightly this is done. A typical optimal control approach involves the minimization of some function of the process measurements, the foremost example being the minimization of a scalar linear quadratic function of the process inputs and states (the LQR problem). Product quality is not explicitly part of the performance criteria when designing this controller. Good quality is achieved because a process expert has given the control engineer a set of physical specifications which

the expert knows will yield good product. A good controller will make a process less susceptible to variations, but this is only the first step towards good quality product. The automatic quality control approach goes one step further by making the achievement of *good* quality the criteria by which controller performance is measured.

The multiple-objective optimization approach is also unique. When controlling quality, this approach is necessary when one considers the nature of quality. Ultimately, only one operating point is used, and therefore tradeoffs among quality characteristics are made. One approach to optimizing quality would be to form an aggregate cost measure and use this to determine optimal operating points. One weakness of aggregation is that tradeoffs are made before optimization takes place. It may also be necessary to choose individual quality measures to accommodate the aggregate cost function (again, consider the requirements when designing using the LQR formulation). By leaving the cost function as a vector, the system retains flexibility by allowing the operator to make tradeoff decisions based on current needs and goals.

1.3 Application to the Injection Molding Process

The automatic quality controller will be implemented on the injection molding process. Plastic injection molding is a cyclical process used to make complexly shaped plastic parts. The goal of the injection molding machine operator is to consistently produce parts which meet some predefined quality specifications. Part quality can be measured in terms of suitable material properties such as strength, surface finish, and shape. The key word above is *consistently*. The time-varying nature of the process and the complex relationships between molding conditions and part properties make this a difficult task.

Each cycle of the process has four phases, which occur in the following order:

1. *Plastication* - the polymer is melted and deposited in front of an injecting mechanism. This continues until the required amount of polymer has been accumulated.

2. *Injection* - the molten polymer is forced into the mold under high pressure. This continues until the mold is completely full.

3. *Holding* - the molten polymer is held in the mold, under pressure, to ensure complete mold filling. Cooling of the part begins at this point. Holding continues until the gate has frozen, at which point no more polymer can enter the mold cavity.

4. *Cooling* - the polymer is allowed to cool in the mold. This continues until the temperature of the formed part is low enough so that it will retain its shape once ejected from the mold.

A cut-away view of a reciprocating screw injection molding machine is shown in Figure [1.1]. In this type of machine, the screw is used to both *melt* the polymer and *pump* it into the nozzle area. This melting and pumping is accomplished by rotation of the screw. As the melting and pumping occurs, the screw retracts, creating the volume occupied by the polymer melt. Injection and holding are accomplished by forcing the screw forward, like a piston. (A check valve at the screw tip prevents a flow of polymer back into the screw.) Once the polymer is in the mold, cooling begins, and at a predetermined time, the mold opens and the solidified part is ejected. There are other types of injection molding machines which accomplish these four phases via different mechanisms, but the phases and sequencing are essentially the same [2].

A sequential block diagram of the molding cycle is shown in Figure [1.2]. Interactions between these phases occur when one phase establishes the initial conditions for a subsequent phase [3]. It is the complexity of these interactions which motivates

the need for control of the process. These interactions are shown in Figure [1.3] and are discussed in detail in Appendix A.

1.4 Organization of the Thesis

In Chapter 2, more background on the injection molding process is presented. This discussion focuses on process modeling and the relationship between inputs, processing conditions, and quality. Experimental data from the Advanced Manufacturing Laboratory's Arburg injection molding machine is presented. These results motivate the need for control of quality, and not just of processing conditions.

In Chapter 3, the quality control problem will be defined rigorously. The quality control policy will be defined in terms of solving a multiple-objective optimization problem. This chapter also includes a brief review of multiobjective optimization. Finally, an algorithm for automatic quality control is presented.

As discussed previously, the multiobjective algorithm will be used for quality control in an injection molding process. Details of the experimental equipment are contained in Chapter 4.

Chapter 5 contains the discussion of the results of controller tuning on the injection molding machine using the multiobjective optimization algorithm. Three different case studies are presented: tuning of a plastication phase controller in simulation, on–line tuning of this same controller, and the on–line tuning of an injection phase controller.

Presentation of the quality control results is done in Chapter 6. Quality control for two different molds was investigated. Three quality criteria are defined for a spiral mold: spiral length, cycle time, and flashing. For an ASTM four cavity test specimen mold, the following criteria are used: flashing, underfill, and cycle time. For both studies, process setpoints are automatically adjusted to optimize quality.

Conclusions and suggestions for future research are presented in Chapter 7.

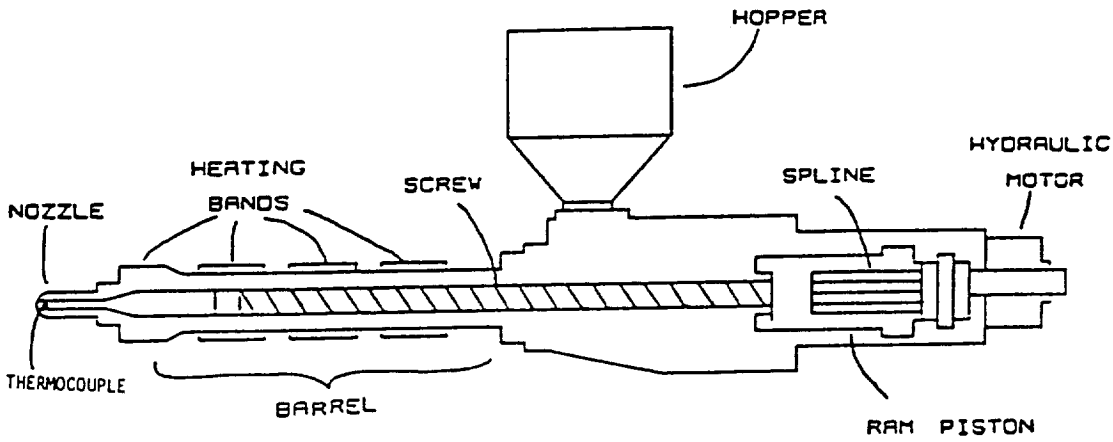

Figure 1.1: Cut-away View of a Reciprocating Screw Injection Molding Machine

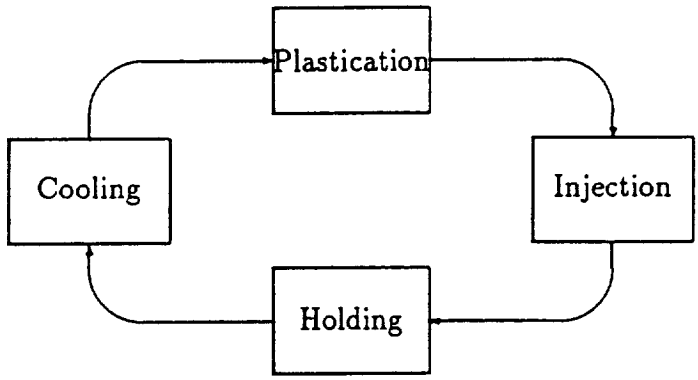

Figure 1.2: The Injection Molding Phases

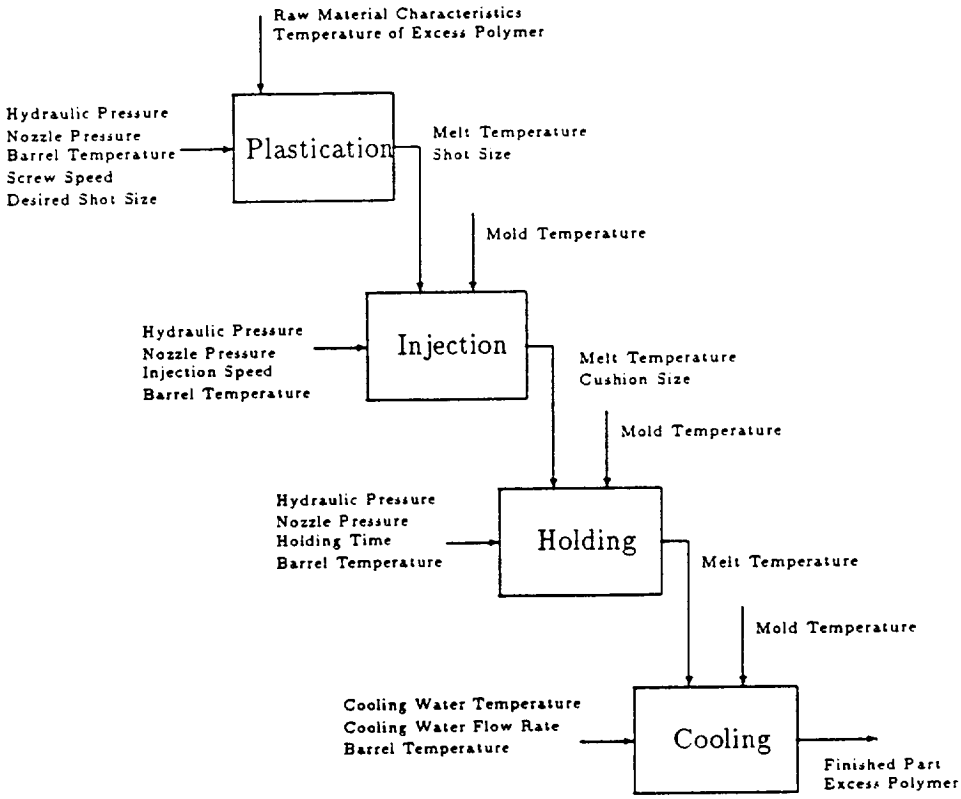

Figure 1.3: Injection Molding Phases and Interactions

Several appendices are included which will provide more extensive background material. The modeling of the injection molding process and a derivation of models specific to the Arburg machine can be found in Appendix A. A guide to the notation used in the equations throughout this thesis is found in Appendix B. In Appendix C, a simple example is presented which highlights aspects of the mulitobjective optimization algorithm.

2. The Injection Molding Process and Part Quality

2.1 Introduction

Part quality is easy to label but difficult to define quantitatively. There are two reasons for this: quality is specific to the function of the part, and not all quality measures have a quantitative basis [1]. Quality "measures" for an injection molded part can fall into one of the following categories [33]:

- Mechanical Properties;
- Dimensions;
- Appearance.

Each of these categories is made up of various components and each component measures a different aspect of the part. These components come in two distinct forms, *variables* or *attributes* [34]. A variable is a characteristic or property which can be measured on a continuous scale. Part dimension is a variable, measured in some appropriate unit of length. An attribute is a characteristic which the part does or does not have. The presence of streaks in a colored part is an attribute: either coloration is solid or it is not. Quality categories and typical components are listed in Table 2.1. This is by no means an exhaustive list because with every new design there will be some new measure of quality.

Kamal and Bata, in [35], stated

> *The thermo-mechanical history experienced by the polymer during flow and solidification results in the development of microstructure (morphology, crystallinity, and orientation) in the manufactured article. The ultimate properties of the article are closely related to the microstructure.*

Therefore, the control of the process and product quality must be based on an understanding of the interactions between resin properties, equipment design, operating conditions, thermo-mechanical history, microstructure, and ultimate properties.

Polymer research involves elucidating the following three relationships:

- f_p - the relationship between processing and microstructure;
- f_m - the relationship between microstructure and part mechanical properties;
- $f_p \cdot f_m$ - the overall relationship between processing and mechanical properties.

Experiments with the goal of determining f_m (or "f_m-research") would be considered fundamental research. The goal is to determine the relationship between the microstructure and aggregate mechanical properties of the material, regardless of the process which is used to produce the material. Experiments with the goal of determining $f_p \cdot f_m$ ("$f_p \cdot f_m$ research") is more oriented to manufacturing and control. Here, the goal is to determine the relationship between processing and mechanical properties, one application being to determine the limits of a particular processing technique with respect to properties of the final product.

2.2 A Review of Injection Molding Research

In this section, the literature pertinent to the plastic injection molding process is reviewed. The research described in this work draws from three fields: injection molding, process control, and multiple objective optimization. Because of this diversity, there is no single literature review chapter. Literature pertinent to the process modeling, control, and quality specifically for injection molding will be reviewed here. Otherwise, when appropriate, the literature on process control and multiobjective optimization will be discussed.

Table 2.1: Quality Categories and Components

Categories	Components
Mechanical Properties	Tensile Strength Flexural Strength Izod Impact Strength Ball-drop Impact Strength Stiffness Modulus Elastic Modulus
Dimensions	Absolute Size Shrinkage Part Weight Part Density Part Volume
Appearance	Color Color Mixing Surface Smoothness Clarity

2.2.1 Research in Modeling and Control

Dynamic studies of the process can be divided into two categories: those which attempt to build a *mechanistic* model of the process and those which attempt to build an *empirical* model of the process. A mechanistic model typically consists of partial differential equations based on mass balance, momentum balance, and energy balance equations. To these equations, the boundary conditions describing a particular situation must be applied [4]. An empirical model of the process is an attempt to form a rough approximation of the behavior of the process using simple models. Empirical models are based on some convenient form such as a linear static model or a transfer function [5, 6]. It must be emphasized that the best mechanistic or empirical models will only provide a *general clue* to the input and output relationships in the process, the actual situation is much more complex.

The foundation work on polymer processing was done by Spencer and Gilmore in [7, 8]. Their goal was to determine the role of pressure, temperature, and time in

injection molding. The result was an empirical equation of state for polymers.

The most comprehensive work on *plastication* was done by Donovan in [9, 10, 11]. The emphasis of this work was on melting due to conduction heating. The result was a theoretical model which would predict the melt profile and temperature profile. Donovan, Thomas, and Leverson then verified this using the *cooling experiment.* Details of this can be found in [9]. Lipshitz, Lavie, and Tadmor [12] developed a melting model which took viscous dissipation into affect. This was modeled as a periodic step disturbance that occurred during screw rotation. Raimund used these results as the foundation for his experimental work [13]. His experiments, and a new analysis using his data, will be presented in Section 2.3.1. Kamal, Gomes, and Patterson, in a series of papers [14, 15], studied the dynamics and control of melt and barrel temperature. The focus of this work was on the implementation and effect of feedback controllers. They developed empirical transfer functions of melt temperature as a function of barrel temperature and melt pressure. They then used these transfer functions to evaluate different control strategies.

The mechanistic modeling of injection has centered around the modeling of the unsteady flow of a hot, non-Newtonian polymer melt into a cold cavity. These dynamics are modeled as unsteady-state free surface flow coupled with transient cooling and are described by the basic equations of change. Tadmor, Boyer, and Gutfinger, in [16], modeled the flow of polymer into the melt cavity using the finite element method. Experimental work by Kreuger and Tadmor [17] validated this modeling method by studying the injection of polymer into a rectangular cavity with various obstructions. One result was that it could be assumed that molten polymer behaved as a Newtonian fluid, which simplified modeling and simulation. Later work by Mavridis, Hyrmak, and Vlachopoulos [18], Kamal, *et al.* [19], and Gogos, Huang, and Schmidt [20] included fountain flow in the modeling. Understanding of the injection phase is important because shear and elongational stresses cause high

orientation of the polymer. Polymer which contacts the mold surface has frozen-in orientation while the remainder of the polymer has time to relax while cooling. These microstructure differences within a part, due to the injection process, can have a great impact on quality properties such as strength and warpage.

Empirical modeling of the injection phase has focused on control of injection velocity or pressure. In a sense, the filling of the mold can be considered "open-loop" once the plastic leaves the nozzle. Shankar in [21] and Shankar and Paul in [22] developed a lumped-parameter model for the injection process and evaluated a state-space approach to injection control. Costin, Okonski, and Ulicny in [23] examined adaptive control of the injection process and compared it to the performance of a PI controller for hydraulic pressure control. Haber and Kamal [24] and Kamal *et al.* [25] also studied the control of pressure during injection.

Ma presented one of the first analyses of the injection molding process as a system [3]. The fundamental result is that no portion of the cycle exists in isolation. This was further emphasized by the experimental work of Whelan in [29], Sanschagrin in [30], and the discussion of the importance of cycle-to-cycle control in the survey of injection molding machine control done by Agrawal, Pandelidis, and Pecht in [31].

All of the process control techniques have one thing in common: the process is controlled but the effect of the control has not been related to the quality of the product. *The operator has been given new, more complicated knobs, but has not been told where or how to set them.* Product quality must be used to provide feedback on the dynamic performance of the process.

2.2.2 Research on the Effect of Microstructure on Mechanical Properties

Much research has been done in trying to determine the relationship between microstructure and mechanical properties ("f_m research"). One example is the study of the influence of orientation on the mechanical properties of injection molded polystyrene, by Hoare and Hull [26], Bayer *et al.* in [27], and Lopez Cabarcos, *et al.* in [28]. Other studies compare the affect of crystallinity, the type of polymer, the length of the polymer chains, or any number of other characteristics. This research was carried out to understand the correlations between processing and properties. The goal was modeling, and not control.

A common thread through all of these types of studies is the approach. The experimenters select a particular aspect of the microstructure to examine, find a process which will produce this aspect preferentially, and allow some method to vary the aspect over a defined range to determine the effect on the properties under study. In the case of Hoare and Hull, two different processes were used to produce specimens with orientation. One set was made by using a well-characterized polystyrene sheet and a second set was made by injection molding thin plaques. Molecular orientation was varied by cutting specimens out of the sheet or plaques in different directions, as shown in Figure [2.1]. The processing to achieve the orientation was not examined, indeed the goal was to produce sheet or plaques as identical as possible. The differences in microstructure in the specimens were induced by the method of sample preparation, the result being that the relationship between microstructure and properties can be studied independently of the process.

2.2.3 Research on the Effect of Processing on Mechanical Properties

Once the relationship between the microstructure and mechanical properties is understood, it is possible to determine the right plastic for the job. Given the

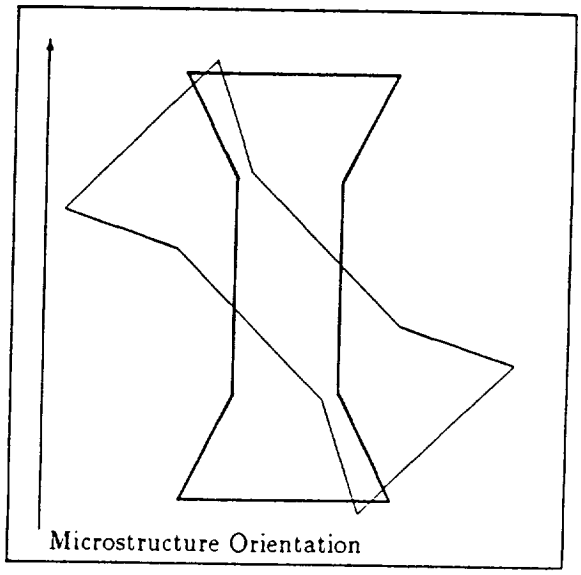

Figure 2.1: Preparation of Specimens with Two Different Orientations

specifications for the part, a plastic is chosen. Now the product must be manufactured. An understanding of the relationship between processing and properties ("$f_p \cdot f_m$-research") must be developed.

Whelan, in [29], showed that by varying the processing conditions, a wide range of mechanical properties could be generated in the product. Similar conclusions were reached by Lopez Cabarcos, *et al.* in [28]. The reason for this was that by changing the processing conditions, different microstructures could be developed in the molded part, thereby changing part properties. Whelan also showed that even if machine settings were held strictly constant, there still remained variation in the product due to factors not under direct control (*e.g.* machine dynamic variations and process disturbances).

In a later study [30], Sanschagrin tried to determine those inputs which were most important to the production of consistent parts. He was able to show a substantial reduction in part variation through the implementation of some fairly simple control schemes. More importantly, he showed that the relative importance of the

inputs changed when a different part was being made. Not only is it important to control the process, but it is important to understand how to control the process for any particular product.

Hsieh in [32] applied the Taguchi Method of quality control to injection molding. A new analysis based on this work is discussed extensively in Section 2.3.2. All three of these works specifically address quality as a performance measure, but none of them take advantage of feedback control.

2.3 Preliminary Experiments

As an illustration of the different types of investigations that can be carried out on an injection molding machine and some of the problems with the process, three different experimental programs will be discussed. These programs were carried out on the AML Arburg injection molding machine, the target machine of the research in this thesis.

The first set of experiments to be discussed was research conducted by Raimund [13]. The purpose of these experiments was to develop an understanding of the relationship between the three different inputs which affect melt temperature and the melt temperature that was actually generated. With respect to the research described earlier this could be considered "f_p-research". The second set of experiments was research conducted by Hsieh [32] to develop an understanding between process input and quality. This would be "$f_p \cdot f_m$-research". Finally, experiments were carried out by Redlitz [36] to develop an understanding of the problem of process control for the injection molding process. This would also be "f_p-research". These research programs will be described in the following sections.

2.3.1 Sensitivity of Melt Temperature to Plastication Inputs

In the research conducted by Raimund [13], the goal was to determine the relative importance, to melt temperature, of conduction heating and shear heating during plastication. Experiments were carried out in which the different plastication inputs were manipulated and the resulting effect on melt temperature was recorded. Conduction heating was directly affected by the setting of the barrel temperature. Shear heating was directly affected by setting the screw speed and the melt pressure. Screw speed set the initial shear rate on the polymer and melt pressure controlled the time under which shearing took place by lengthening the plastication time.

Raimund carried out the following type of experiment: a $5cm$ shot size was generated under preset and constant barrel temperature, screw speed, and melt (back) pressure conditions. Barrel temperature was set by selecting the desired temperature on the barrel heater controllers. Screw speed was set using an adjustable hydraulic valve. Melt pressure was set by adjusting a hydraulic pressure relief valve (in steady-state, melt pressure and hydraulic pressure are linearly related). During plastication, melt pressure and recovery rate data were collected using the microprocessor controller [36, 37]. Finally, to collect temperature data, the molten polymer was slowly injected, past the nozzle thermocouple, into the air (no mold was used). A slow injection rate was used (approximately $0.1 cm/s$) so that polymer temperature could be measured without inducing any shear heating errors at the thermocouple tip. These experiments were carried out on two different polymers: polystyrene and polypropylene. In recording the data for these experiments, the average of each process variable value over one cycle was used.

In the following section, a new analysis, based on the data collected by Raimund will be presented and discussed.

2.3.1.1 Analysis of Polystyrene Data

Experiments on polystyrene were carried out using three different barrel temperatures and seven different melt pressures. Screw speed was not varied, possibly because of difficulties in processing polystyrene. A constant value of $300RPM$ was used for all experiments. In Figure [2.2], melt temperature is plotted with respect to back pressure. Notice that melt temperature increases with both increasing back pressure and increasing barrel temperature. To get a better feel for the shape of the relationship, melt pressure versus the difference between barrel temperature and actual melt temperature (or "melt temperature error") is shown in Figure [2.3]. In this figure, the nonlinear relationship between pressure and temperature is much more evident. Also, notice that as the barrel temperature gets higher, the error decreases. This suggests that the dependence on shear heating decreases with increasing temperature. This is reasonable since the amount of shear heating that takes place within the polymer is dependent upon the viscosity of the melt, and this viscosity decreases as temperature increases.

Figure [2.4] shows melt temperature versus recovery rate. This figure is very similar to Figure [2.2]. The reason for this is that recovery rate is linearly dependent on melt pressure and screw speed. This dependence is readily seen in Figure [2.5], where recovery rate is shown versus melt pressure. Again, notice that this relationship is temperature dependent for this polymer.

In Figure [2.6], melt temperature is plotted versus plastication time. The same data is used in Figure [2.7], except that melt temperature error is used. This figure suggests a linear relationship between melt temperature and plastication time. (Compare this with Figures [2.3] and [2.4].) This is further justified when one considers the approximate relationship between plastication time, shot size, and recovery rate:

$$\text{Plastication Time} = \frac{\text{Shot Size}}{\text{Recovery Rate}}$$

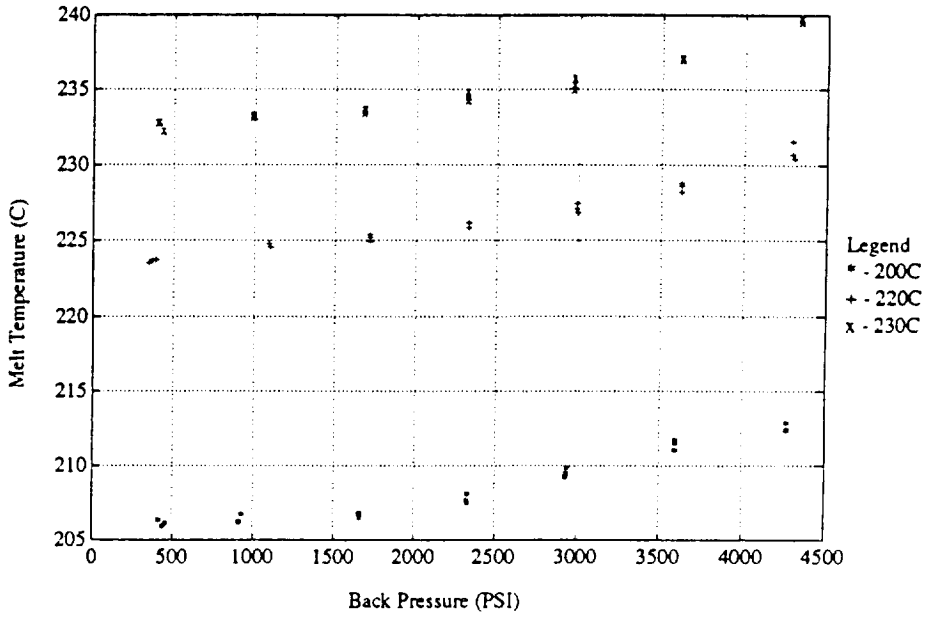

Figure 2.2: The relationship between melt temperature and melt pressure, at three different barrel temperatures, for polystyrene.

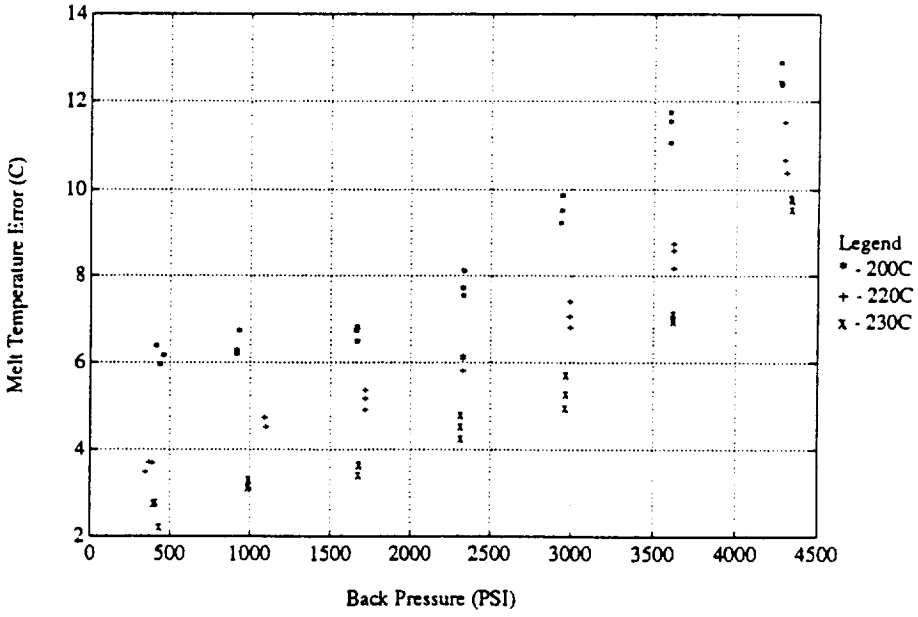

Figure 2.3: Melt temperature error versus melt pressure for polystyrene.

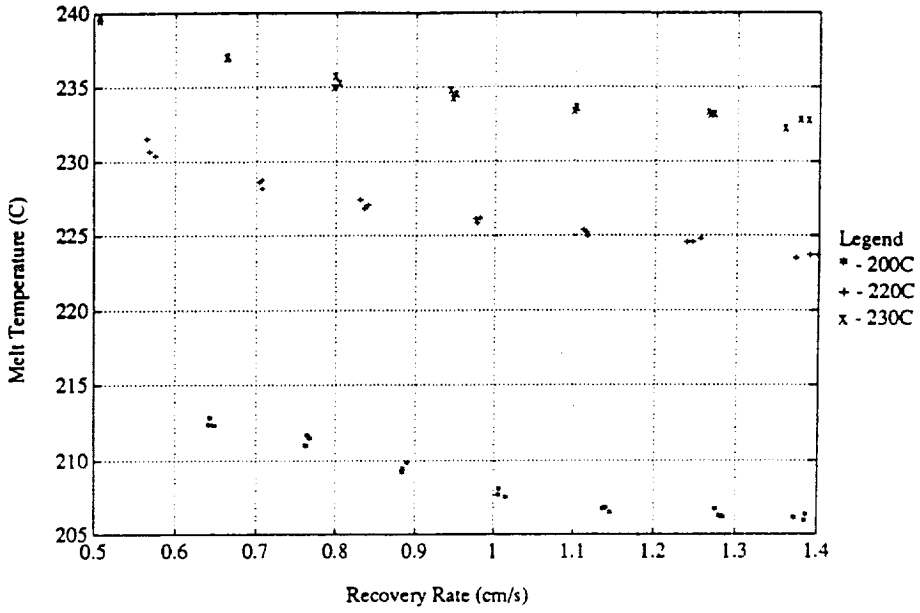

Figure 2.4: Melt temperature versus recovery rate for polystyrene.

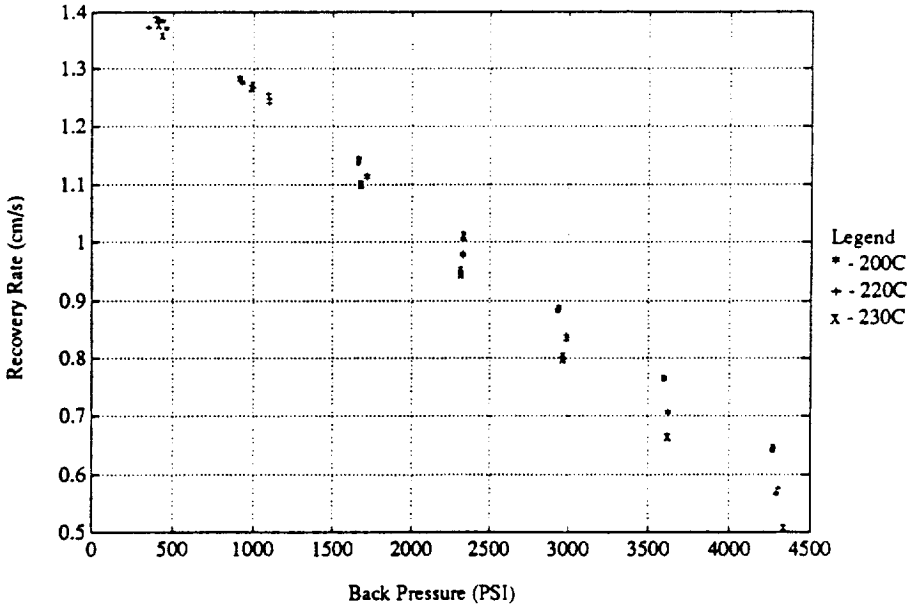

Figure 2.5: Recovery rate versus back pressure for polystyrene.

A linear regression was performed on the data shown in Figure [2.6]. The three input variables were barrel temperature (T_b), plastication time (t_P), and screw speed (ω). The output was average melt temperature (T_m). The resulting model was:

$$T_m = 0.857T_b + 1.317t_P + 0.103\omega \tag{2.1}$$

This equation was used to estimate the melt temperature, given the same input used in the regression. Estimated versus actual temperature is shown in Figure [2.8]. The fit seems to be quite good, but the following should be kept in mind:

1. There are only three different levels for the barrel temperature. Because of this, it is difficult to judge how "linear" the response of melt temperature to barrel temperature really is.

2. Only one value of recovery rate (ω) was used in generating Equation (2.1). Therefore, the model fits three coefficients to data with only two degrees of freedom.

3. There is enough data that linearity with respect to plastication time seems to be justified.

A second linear regression was run on the data, this time using only two parameters: barrel temperature and plastication time coefficients. The resulting model was

$$T_m = 0.996T_b + 1.360t_P \tag{2.2}$$

Estimated versus actual temperature is shown in Figure [2.9]. As expected, the fit is much worse (although not bad). If the plastication time coefficient is compared between the two equations, it will be noticed that it is practically the same. In comparison, the barrel temperature coefficient changes by about 15%. This supports the linearity assumption for plastication time and also underscores the need for more barrel temperature and screw speed data.

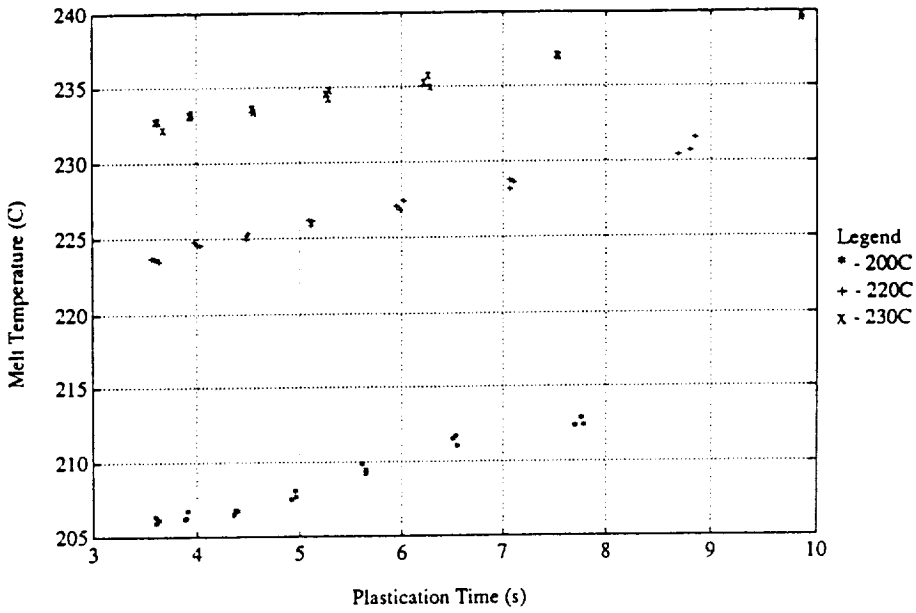

Figure 2.6: Melt temperature versus plastication time for polystyrene.

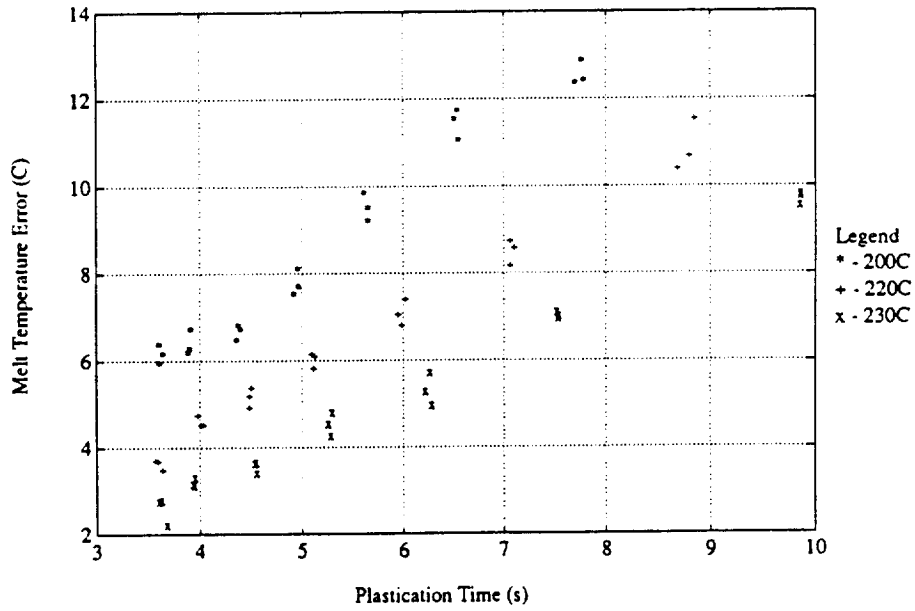

Figure 2.7: Melt temperature error versus plastication time for polystyrene.

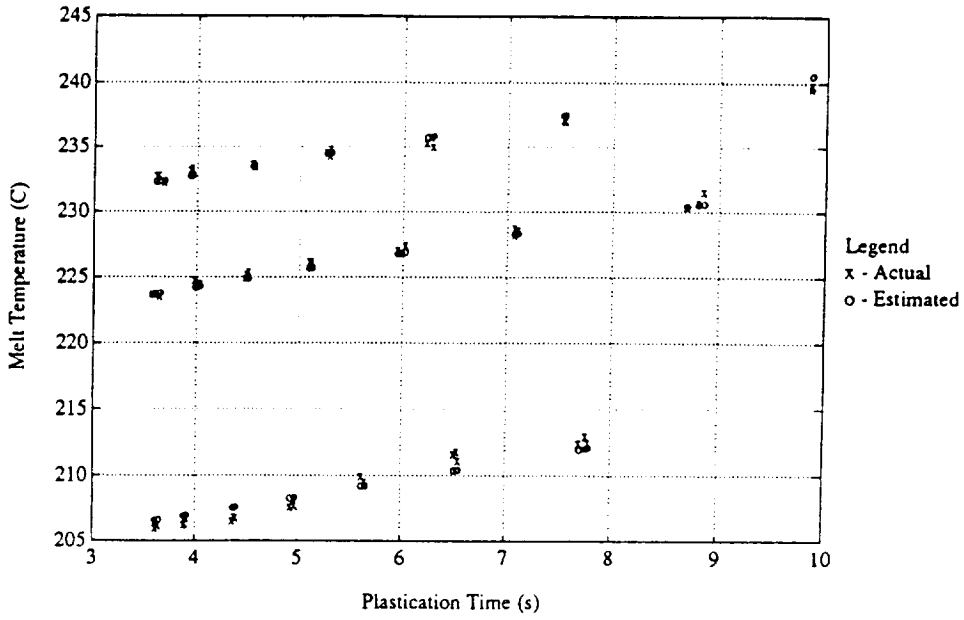

Figure 2.8: Estimated versus actual temperature using a three parameter model.

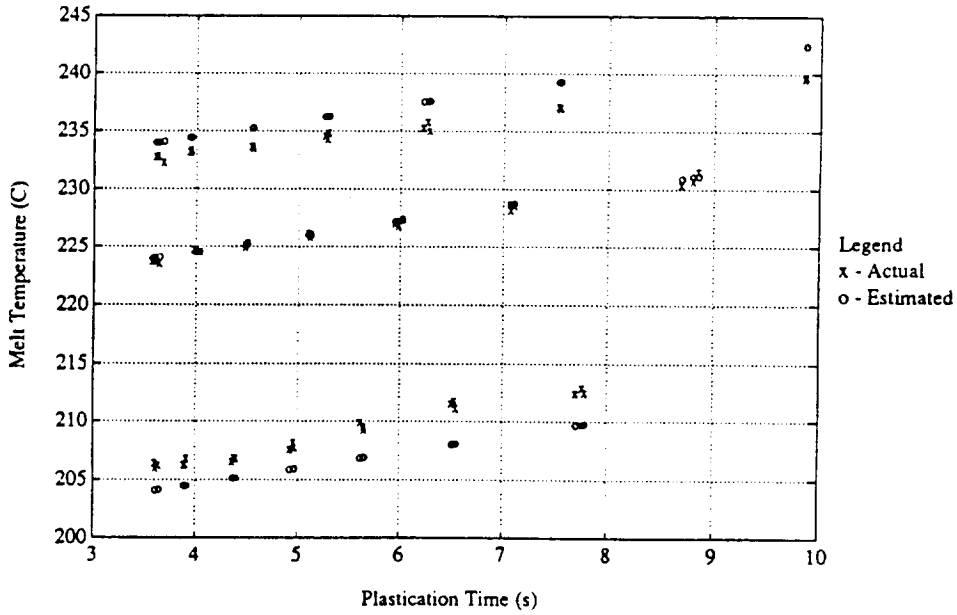

Figure 2.9: Estimated versus actual temperature using a two parameter model.

2.3.1.2 Analysis of Polypropylene Data

An identical set of experiments were run, this time using polypropylene. The dependence of machine processing conditions on the type of polymer used could be studied. In these experiments, it was possible to use two different screw speeds: $200RPM$ and $300RPM$. For polypropylene, four barrel temperature levels were used: $180C$, $190C$, $200C$, and $210C$. The hydraulic pressure levels were kept the same as in the previous experiments.

Melt temperature versus back pressure data for the two different screw speeds are shown in Figures [2.10] and [2.11]. If these figures are compared with Figure [2.2] it can be immediately seen that there is much less of a temperature dependence on back pressure in polypropylene than in polystyrene. This is due mainly to the difference in apparent viscosity of the two polymers.

In Figures [2.12] and [2.13], this same data is plotted with respect to melt temperature error. If these figures are compared with Figure [2.3], it is immediately seen that there is much less of a dependence on barrel temperature. Also, this data is not nearly as "orderly" as that for polystyrene.

One hypothesis is that apparent viscosity of polypropylene is less dependent upon the operating conditions investigated. This is supported by a plot of recovery rate versus back pressure (Figure [2.14]). The linear relationship doesn't change as a function of barrel temperature, only with screw speed. (Compare this with Figure [2.5].) As is discussed in more detail in Appendix A, recovery rate, V_s, is assumed to be linearly related to screw speed, ω, and melt pressure, P_m, in steady-state. Figure [2.14] justifies this assumption, over the range of temperatures tested.

In Figures [2.15] and [2.16], temperature is plotted versus plastication time. Again, the phenomena of melt temperature being linearly related to plastication time is evident.

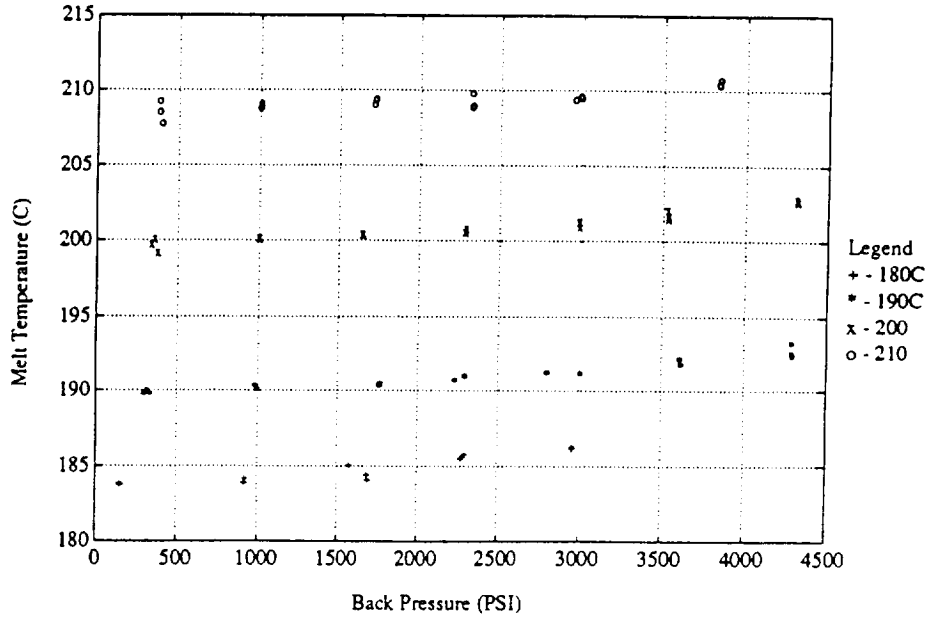

Figure 2.10: Polypropylene melt temperature versus back pressure for four different barrel temperatures at 200RPM.

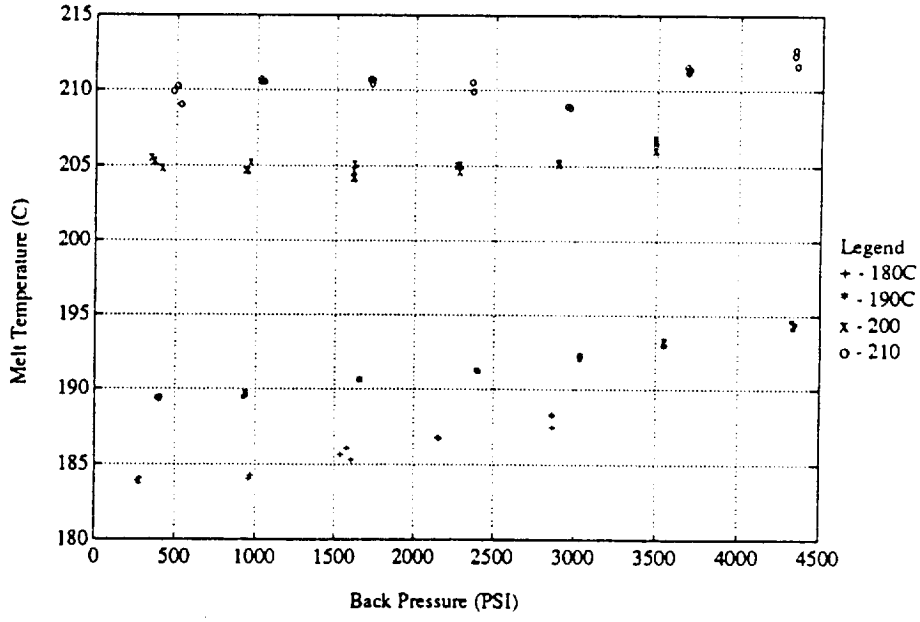

Figure 2.11: Polypropylene melt temperature versus back pressure for four different barrel temperatures at 300RPM.

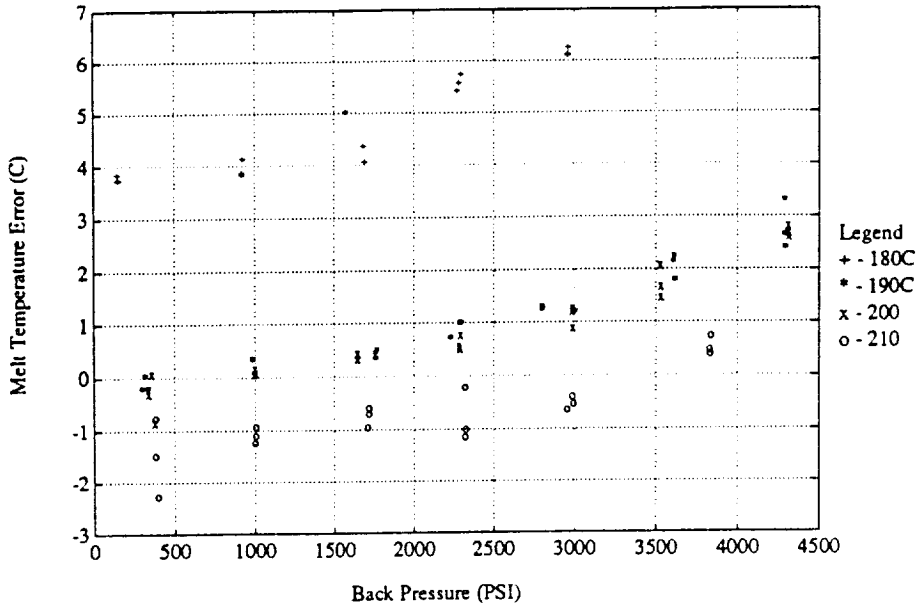

Figure 2.12: Polypropylene melt temperature error versus back pressure for four different barrel temperatures at 200RPM.

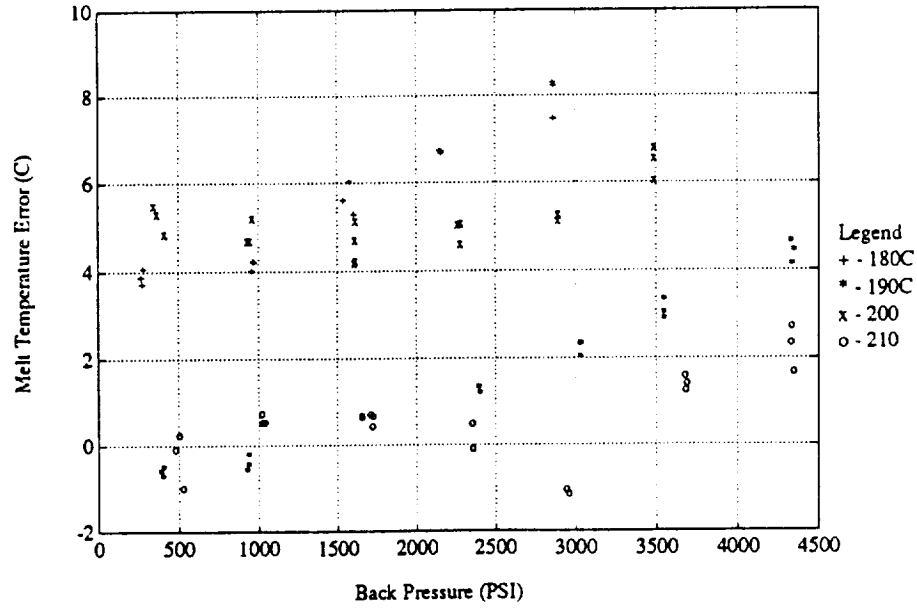

Figure 2.13: Polypropylene melt temperature error versus back pressure for four different barrel temperatures at 300RPM.

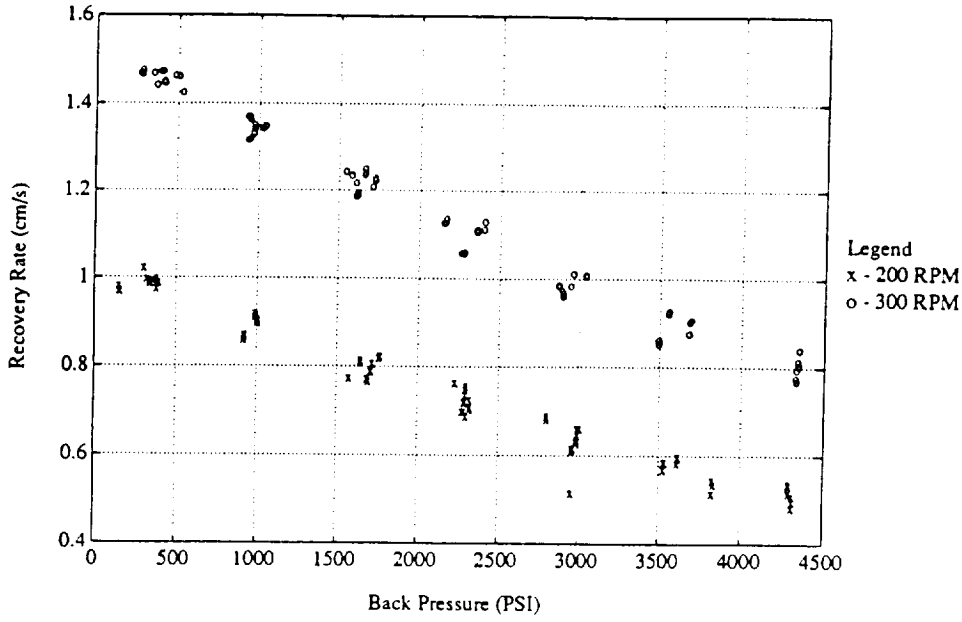

Figure 2.14: Back pressure versus recovery rate: upper curve is $300RPM$ data, lower curve is $200RPM$ data.

As for the polystyrene data, a linear regression was run on the data shown in Figures [2.15] and [2.16] to find a model relating melt temperature to plastication time, barrel temperature, and screw speed. For this model, two levels of screw speed data were available, so a full three parameter fit could be done. For polypropylene, the model was

$$T_m = 0.9284T_b + 0.9136t_P + 0.04373\omega \tag{2.3}$$

Estimated versus actual temperature is shown in Figure [2.17] and [2.18]. The fit is reasonably good although not as good as that for polystyrene. Again, the correct trend is predicted by plastication time, but the prediction of absolute temperature is not as good. Also, the model fits the data better at $200RPM$ than at $300RPM$.

2.3.1.3 Different Processing for Different Materials

Now that the experimental data for processing two different polymers has been presented, some of the differences in the effect of processing on these materials can be discussed.

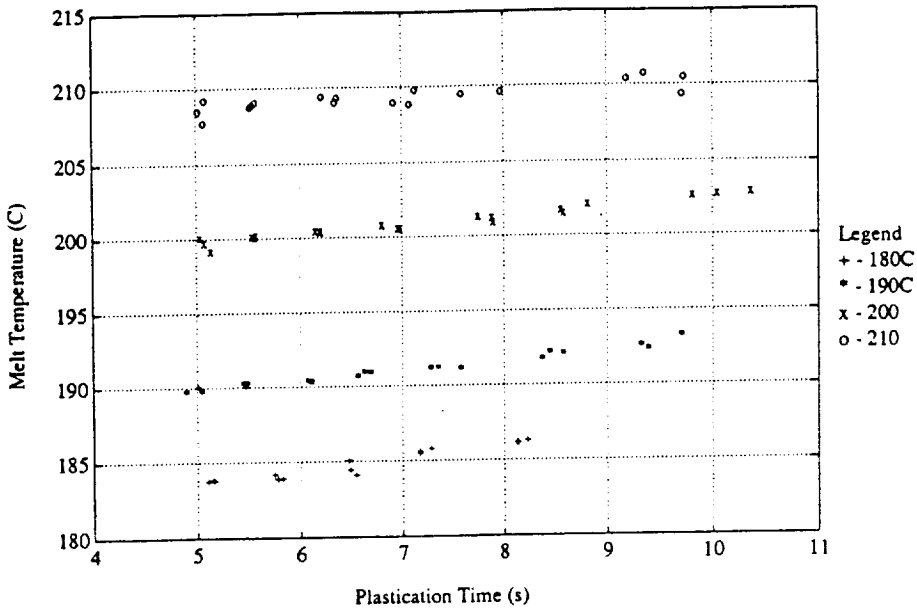

Figure 2.15: Polypropylene melt temperature versus plastication time for four different barrel temperatures at 200RPM.

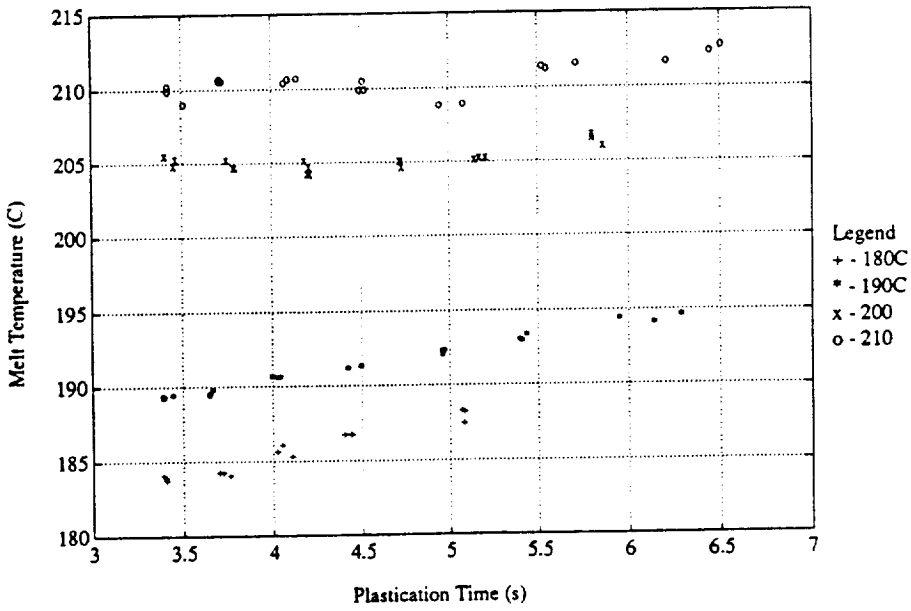

Figure 2.16: Polypropylene melt temperature versus plastication time for four different barrel temperatures at 300RPM.

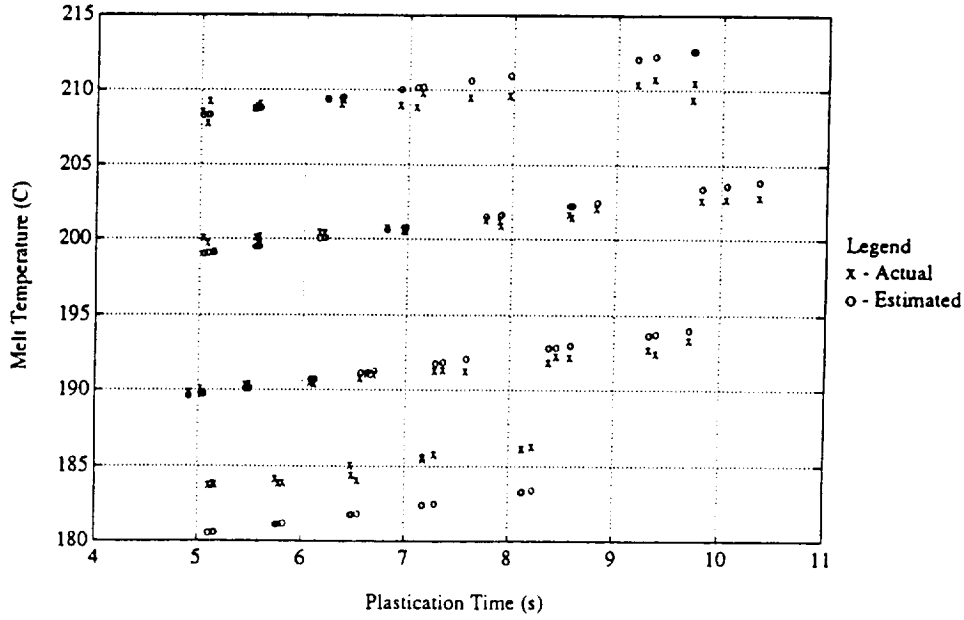

Figure 2.17: Estimated temperature versus actual temperature for polypropylene at 200RPM.

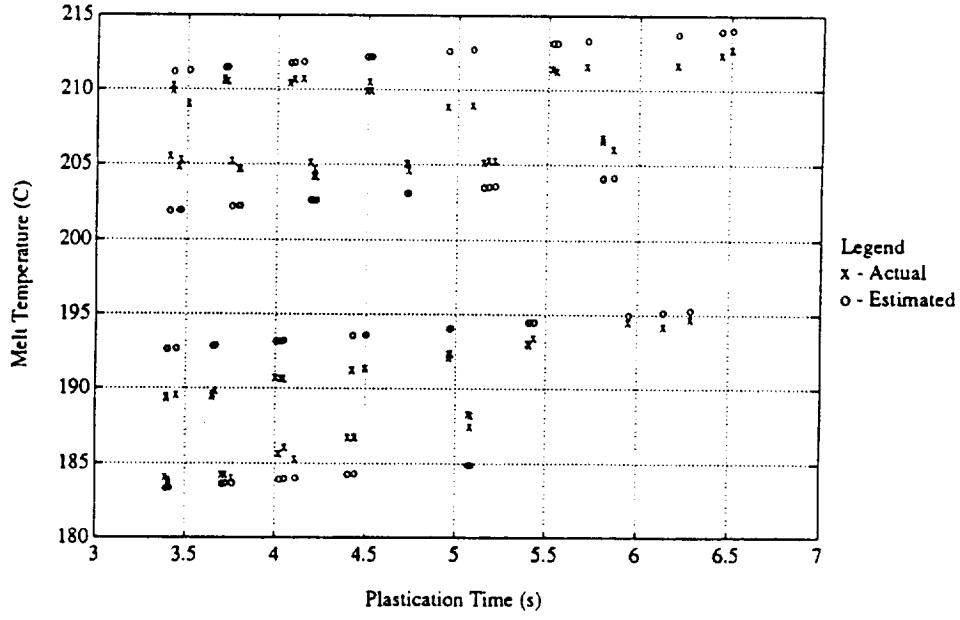

Figure 2.18: Estimated temperature versus actual temperature for polypropylene at 300RPM.

First, consider barrel temperature. For polystyrene, the relationship between barrel temperature and melt temperature is well-behaved. This is not the case for polypropylene. For this polymer, there are some instances where the melt temperature is less than the barrel temperature and in other cases it is higher, but for the most part, barrel and melt temperature are equal. This cannot be easily explained in terms of the processing conditions and must be related to some property of polypropylene.

For both polymers, there is a strong linear relationship between melt temperature and plastication time. Since plastication time is the time under which viscous heating of the polymer takes place, the importance of viscous heating to melt temperature is emphasized.

Finally, all of the analysis carried out on this experimental data was done using statistics of the data. Plastication time was computed using average recovery rate. The heater bands had a $\pm 5C$ error band on the setpoint so only average barrel temperature was known. In Figure [2.19], the melt temperature profile versus linear position in the melt is shown (for polystyrene) for four different back pressures. Notice that every one of the profiles has a hot spot, and this peak temperature spot is at a different position in the melt. Since final molecular orientation is dependent upon the polymer temperature, this profile will result in a part with different molecular orientation depending upon the location in the mold.

All of these observations only underscore the fact that the conditions of the polymer must be measured and controlled, not just machine operating points.

2.3.2 Part Quality Dependence upon Processing Conditions

In the research conducted by Hsieh [32], the goal was to determine optimal machine operating conditions for a particular part. Three different quality characteristics were defined for this part, and a series of experiments were run to determine

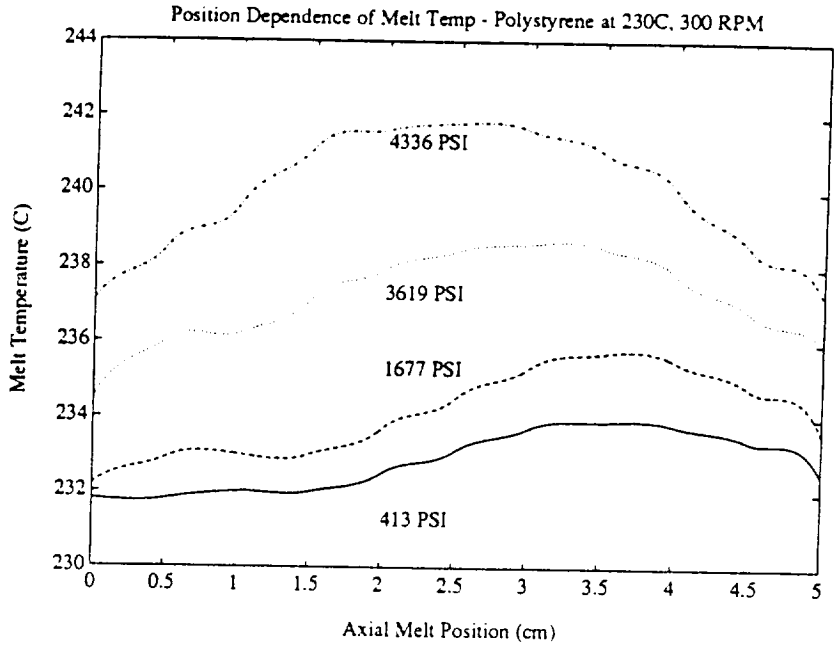

Figure 2.19: Melt temperature versus position in the shot at four different back pressures.

the sensitivities of these characteristics to the different machine inputs.

Hsieh used the *Taguchi method* [38] to determine the optimal input settings using experimental data. This method is essentially a three step process. First, a set of initial experiments are run to determine typical machine output under standard operating practice. This establishes a performance benchmark for the process. Next, a set of inputs are chosen to be manipulated. The choice depends on the experimenter's *a priori* knowledge as to which inputs will be *important*. An orthogonal array, appropriate for the number of inputs and levels to be manipulated, is chosen and a series of experiments are run as prescribed by this orthogonal array. The use of this orthogonal array allows the maximum amount of information to be gotten out of the minimum number of experimental runs. Analysis of this data will yield the sensitivities of the outputs to the different inputs. When using the Taguchi method, standard functions are established to determine optimums, such as *nominal is best* or *larger is best*. These functions provide standard rules of thumb for the process designer. For instance, *nominal is best* simply corresponds to

minimizing the squared error about the desired mean. When used properly, these optimizing functions will determine a set of input conditions which reduce process variance as well as achieve the target quality output. These optimizing functions are considered separately for each input/output pair. It is up to the experimenter to determine trade-offs between competing optimums, as was the case with this series of experiments [32]. A thorough explanation of the Taguchi method can be found in [38].

2.3.2.1 Operating Conditions for the Experiments

For these experiments, a test specimen mold was used (Figure [2.20]). This mold has four cavities: two tensile test specimens, an impact strength test disc, and a weld line test bar. Two *quality variable* outputs were defined for the part: the dimension parallel to flow and the dimension perpendicular to flow on the large tensile test specimen. These dimensions were measured by the experimenter to an accuracy of 0.0001 inches. One *quality attribute* was defined: the amount of flashing that occurs on the whole part. To make analysis easier, Hsieh designated four categories of flash,

- Class I – No flash – part is acceptable;
- Class II – Minimal flash – part is acceptable;
- Class III – Medium flash – part is unacceptable;
- Class IV – Severe flash – part is unacceptable;

The definition of these classes gave the attribute some "variable-like" properties for use in analyzing the data.

Seven inputs were manipulated with three different levels assigned to each input. They were as follows:

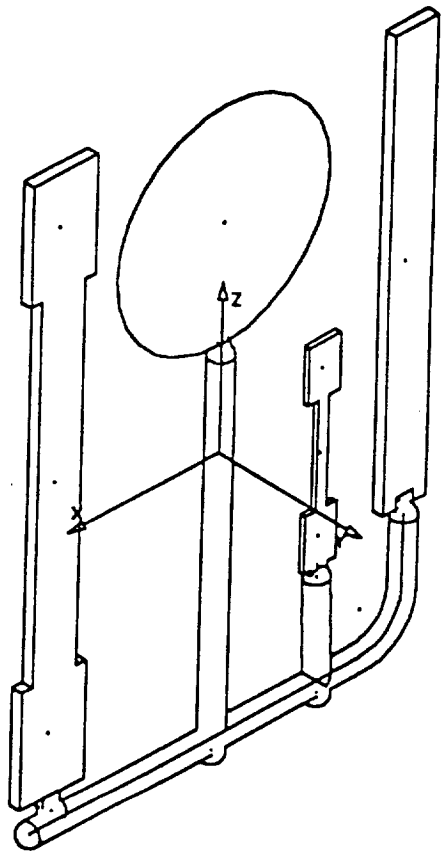

Figure 2.20: ASTM test mold used for part quality experiments.

- Heater Base Temperature – this is the reference temperature at which the barrel was kept. Three levels were used: $210C$, $240C$, and $270C$.

- Heater Temperature Profile – there are four heaters which can be set independently, starting at the nozzle. Two different temperature profiles were investigated:

  1. Uniform – all heaters at the base temperature (T_{ref})
  2. Varied – the heaters are adjusted according to the following schedule:
     (a) $H_1 = T_{ref} - 40C$,
     (b) $H_2 = T_{ref} + 10C$,
     (c) $H_3 = T_{ref}$,
     (d) $H_4 = T_{ref} - 10C$,

- Injection Velocity Profile – this is the velocity trajectory that was used to inject the shot. Three levels were used: 20% uniform, 20% ramped, and 40% ramped. These profiles are shown in Figure [2.21].

- Injection Pressure – this is the maximum hydraulic pressure used during injection. The following levels were used: $900PSI$, $1200PSI$, and $1500PSI$.

- Injection Time – this is the amount of time allowed for injection of the part. The levels used were $1s$, $2s$, and $3s$.

- Holding Pressure – this is the hydraulic pressure applied during holding. The following levels were used: $50PSI$, $120PSI$, and $200PSI$.

- Hold/Cool Time – the ratio of hold time to cool time was varied. Since cooling takes place throughout the entire period, this amount was kept to $25s$. The following hold/cool ratios were investigated: $5/20s$, $10/15s$, and $15/10s$.

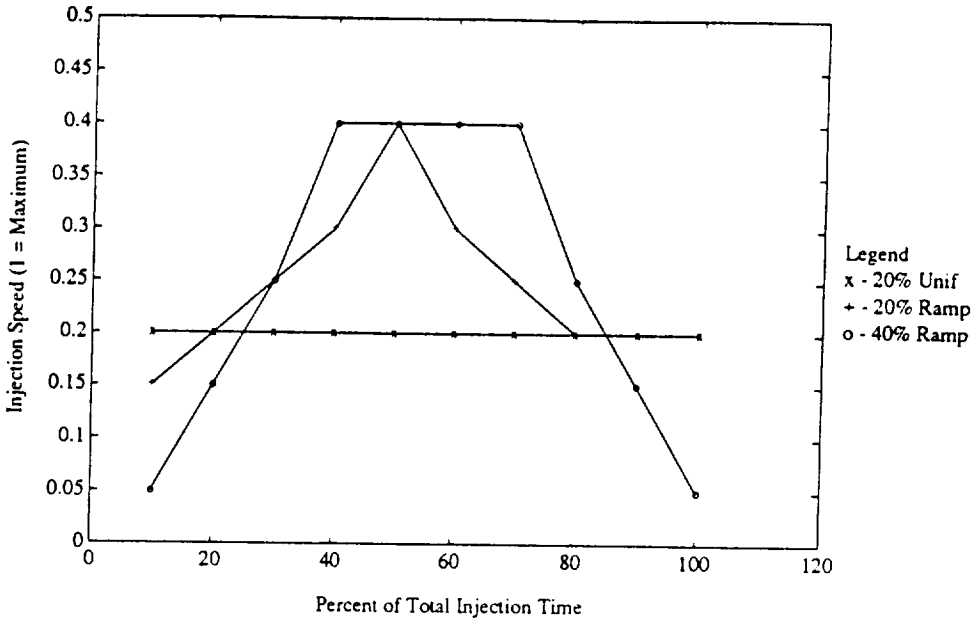

Figure 2.21: Velocity profiles used in the quality experiments.

The following should be noted about the experiments that were run. The injection molding machine was treated as a black box. No attempt was made to choose inputs based on the expected effect on the polymer. This is evident by the fact that *injection time* was chosen as an input to be manipulated. The end of injection is usually signaled by the screw reaching a specific forward position, commonly referred to as the cushion setting. The use of injection time as the terminating signal in these experiments can have two possible consequences: either the injection time will be too short, possibly causing an incomplete fill, or injection time is too long and the plastic undergoes what is essentially a "high-pressure" holding phase before actual holding commences, increasing the probability of flashing. One other problem was that injection pressure and velocity were both used as control inputs. These inputs are actually dynamically coupled and it is not possible to control both at the same time. Finally, inputs were set but the actual machine response was not measured. It is not known whether the machine actually achieved the desired setpoints and profiles and therefore whether the input manipulated has any effect upon the process.

2.3.2.2 Initial Machine Performance Experiments

Since the goal of these experiments is to determine optimal settings for machine inputs, a set of experiments were run at commonly used machine settings to establish a performance benchmark. Twenty-six experiments were run at the input settings shown in the first column of Table [2.2].

In Figures [2.22] and [2.23], histograms of the dimensional data are shown. The parts were grouped in 0.0002 inch increments for these charts. These histograms roughly approximate the probability distribution of the random process affecting these quality measurements. The mean and variance of the dimensional output are given in Table [2.3]. The number of parts in each of the classes of flash is given in Table [2.4]

As can be seen from the histograms, the distribution is somewhat spread out. Also the means are off of the target values of 5.0 inches for the parallel dimension and 0.5 inches for the perpendicular dimension.

2.3.2.3 Machine Performance Using *Optimal* Inputs

A set of experiments was performed, based on an orthogonal array experimental design, and the resulting part data was analyzed, as prescribed by the Taguchi method [38]. The method predicted that the process inputs shown in Table [2.2] would optimize the outputs according to the *nominal is better* criteria for the dimensional outputs and the *larger is better* criteria for the flash output (*larger is better* refers to the number of parts in Class I).

Twenty-nine experiments were then performed at the new settings and the parts were measured. Histograms of the dimensional outputs are shown in Figures [2.24] and [2.25]. Both part quality distributions are much "tighter" than those of the initial experiment and the product is definitely closer to the target. The mean and standard deviation of the parts for this experimental run are given in

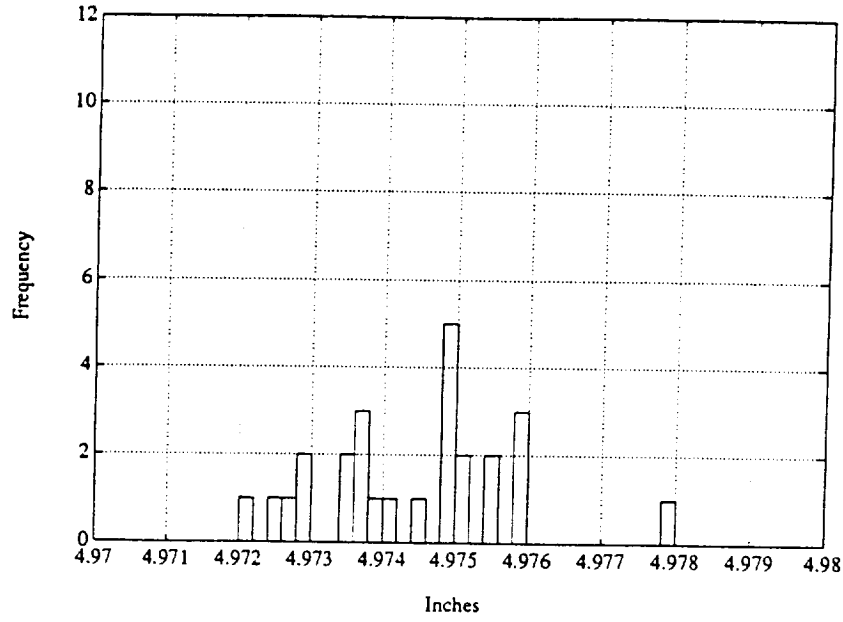

Figure 2.22: Histogram of parallel dimension data for the initial experiment

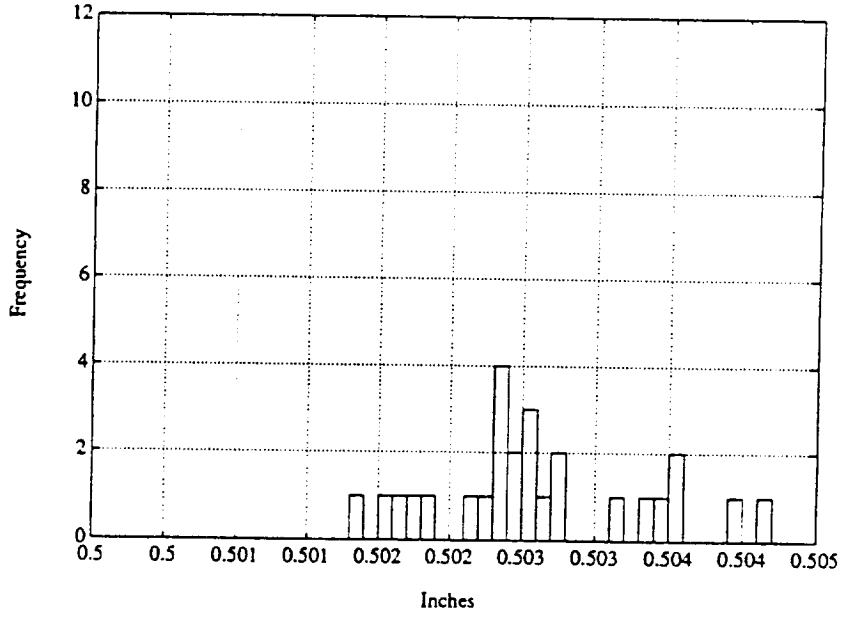

Figure 2.23: Histogram of perpendicular dimension data for the initial experiment

Table [2.3].

Also, according to Table [2.4], all of the parts are acceptable. This is deceiving in that some of the parts were actually incomplete. This quality characteristic is important, but was not measured by any of the three quality characteristics actually used, so it could not be taken into account in the choice of inputs.

2.3.2.4 Machine Performance Using *Adjusted Optimal* Inputs

A third set of experiments were run using the inputs shown in the last column of Table [2.2]. These inputs were adjusted, based on recommendations from an experienced operator, to compensate for the incomplete fill problem. For this run, all parts were acceptable, although most parts fell into flash output Class II, rather than Class I. Histograms of the part data from this run are shown in Figures [2.26] and [2.27], and the statistics are shown in Table [2.3]. From these figures, it can be seen that machine performance is still quite good, as compared with Figures [2.22] and [2.23].

2.3.2.5 Analyzing the Quality Control Approach

In these experiments, the relationship between part quality and operating conditions of the injection molding conditions was investigated. Using a methodical approach, it was quite easy to find new operating conditions for the injection molding machine which significantly increased the quality of the product. In fact, if the different sets of inputs are compared (Table [2.2]), large improvements in quality can be achieved with relatively small adjustments in the operating conditions.

One problem with this approach is that it is somewhat open-loop. The process variables are not measured, so the actual machine response to the chosen inputs is not known. Also, since the process variables are not measured, if variations occur in the output, it may not be possible to immediately determine which input should

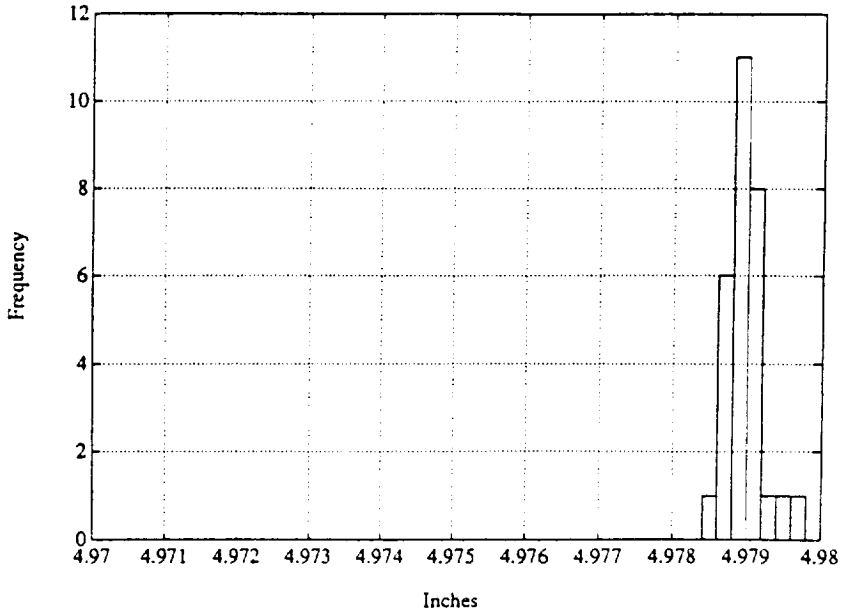

Figure 2.24: Histogram of parallel dimension data for the experiment using the optimal inputs.

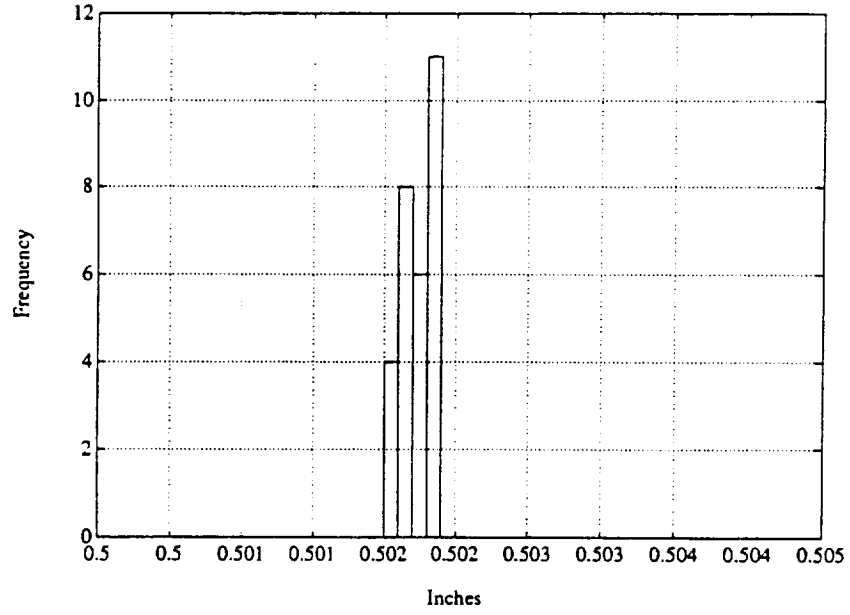

Figure 2.25: Histogram of perpendicular dimension data for the experiment using the optimal inputs.

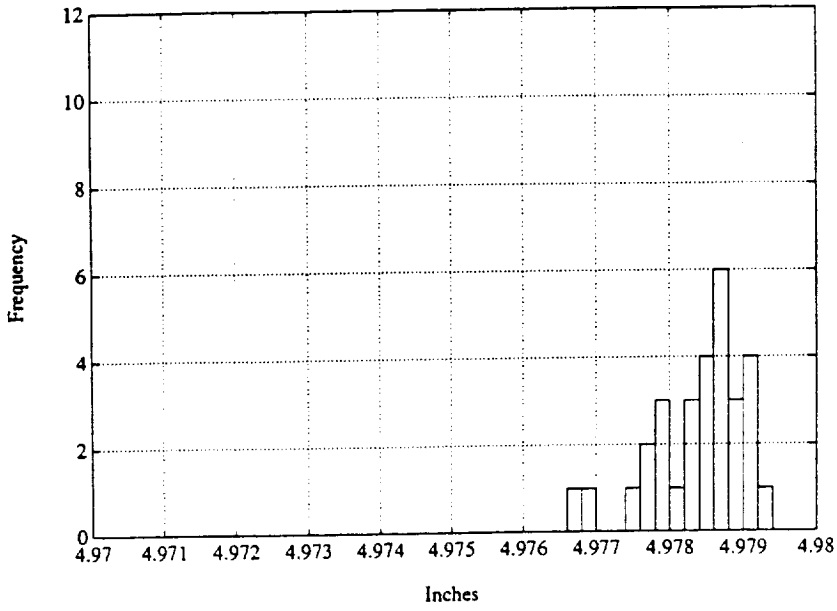

Figure 2.26: Histogram of parallel dimension data for the experiment using the adjusted optimal inputs.

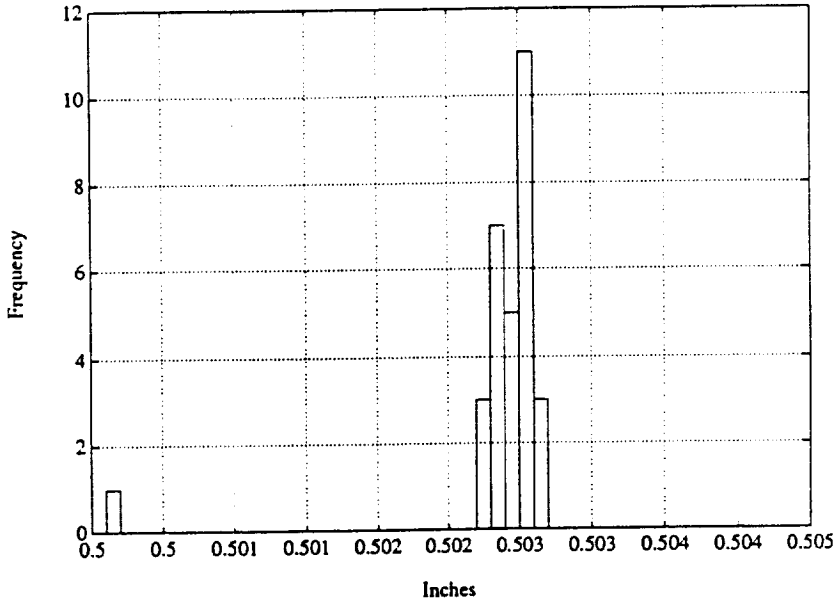

Figure 2.27: Histogram of perpendicular dimension data for the experiment using the adjusted optimal inputs.

Table 2.2: Inputs used for the quality experiments.

Inputs	Initial	Optimal	Adjusted
Heater Base Temperature	$215C$	$210C$	$220C$
Heater Temperature Profile	Uniform	Uniform	Uniform
Injection Velocity Profile	20% Ramped	20% Ramped	20% Uniform
Injection Pressure	$1100PSI$	$900PSI$	$900PSI$
Injection Time	$1s$	$1s$	$1s$
Holding Pressure	$100PSI$	$50PSI$	$50PSI$
Hold/Cool Time	$5/19s$	$15/10s$	$15/10s$

Table 2.3: Statistics of the dimensional data.

	Initial		Optimal		Adjusted	
Dimension	Mean	St. Dev.	Mean	St. Dev.	Mean	St. Dev.
Parallel	4.9745	$1.292 \cdot 10^{-3}$	4.9790	$0.255 \cdot 10^{-3}$	4.9784	$0.619 \cdot 10^{-3}$
Perpendicular	0.5032	$7.137 \cdot 10^{-3}$	0.5023	$0.109 \cdot 10^{-3}$	0.5029	$0.536 \cdot 10^{-3}$

Table 2.4: Flash output for the three experiments.

Flash	Initial	Optimal	Adjusted
Class I	7	29	4
Class II	17	0	26
Class III	2	0	0
Class IV	0	0	0
Total	26	29	30

be changed to correct the problem. In the next section, a feedback approach to machine control will be discussed.

2.3.3 Sensitivity of Machine Dynamics to Operating Conditions

In research conducted by Redlitz [36, 37], a microprocessor controller was implemented to study control of the plastic injection molding process. Redlitz implemented PID control of melt pressure with a hydraulic servovalve as the actuator. The gains of the controller were tuned heuristically, and repeatability of the process and sensitivity of the process dynamics to the operating point was investigated.

First, the process controller was tuned using a melt pressure setpoint of $5000 PSI$. After several iterations, it was found that the best attainable process response was that shown in Figure [2.28]. This run had the following statistics:

- Overshoot = 6.4%
- Undershoot = 7.1%
- Steady-state Error = 0.47%

Using the same controller gains, runs were made at two other setpoints: $7000 PSI$ and $3000 PSI$. The melt pressure responses for these runs are shown in Figures [2.29] and [2.30]. These figures show that the dynamic characteristics change drastically with the operating point. It is important, therefore, to understand the range of operating points for which a particular set of gains produces a good output, and possibly to determine a gain schedule for different setpoints.

The controller was tuned on an injection molding machine that had already been warmed up. A set of experiments were run to see how this warming affected the machine dynamics. Fifty runs were made, using a setpoint of $5000 PSI$ and the previously determined controller gains. Two statistics were collected from each of

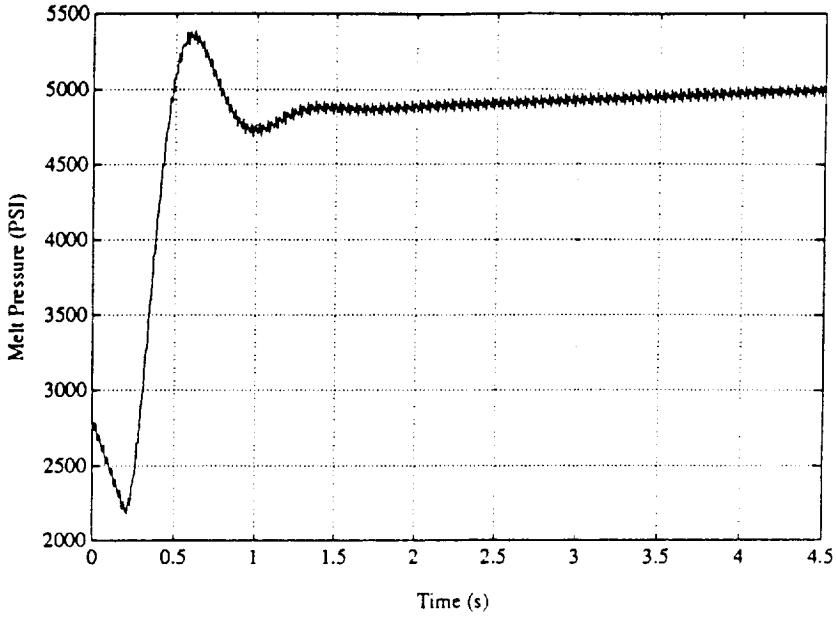

Figure 2.28: Melt pressure response at a setpoint of 5000*PSI*.

the runs: overshoot and steady state error. The results of this experiment are shown in Figures [2.31] and [2.32].

Both performance measures showed some stochastic behavior over the fifty runs. The variance of steady-state error decreased somewhat, but not significantly over the course of the experiments. The variance of the overshoot decreased significantly by the end of the experiments. This is an indication that the transient dynamics of the system are affected by thermal transients in the machine.

Other experiments have shown that there is a marked change in performance which is dependent upon the hydraulic oil temperature. Cooling of the injection molding machine hydraulic oil was done using tap water and thermostat control. The heat transfer is dependent upon the ambient temperature of the water, but the temperature at which the thermostat opens is fixed. When ambient water temperature is very low, approximately $18C$, the thermostat control system enters a limit cycle with an amplitude of $6C$ and a period of about 30 minutes. This had a remarkable effect on the system dynamics as can be seen in Figures [2.33 and [2.34]. Figure [2.33] shows the evolution of the plastication step response over time. As the

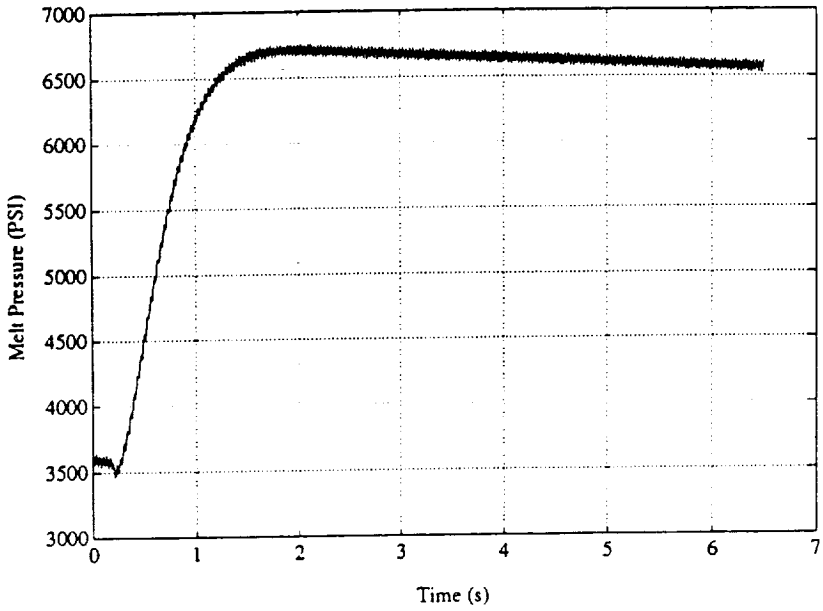

Figure 2.29: Melt pressure response at a setpoint of 7000*PSI*.

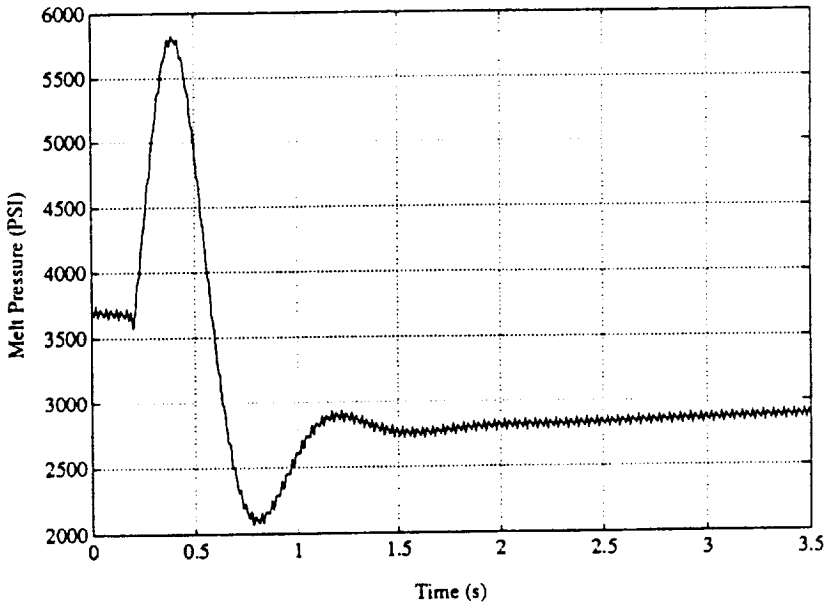

Figure 2.30: Melt pressure response at a setpoint of 3000*PSI*.

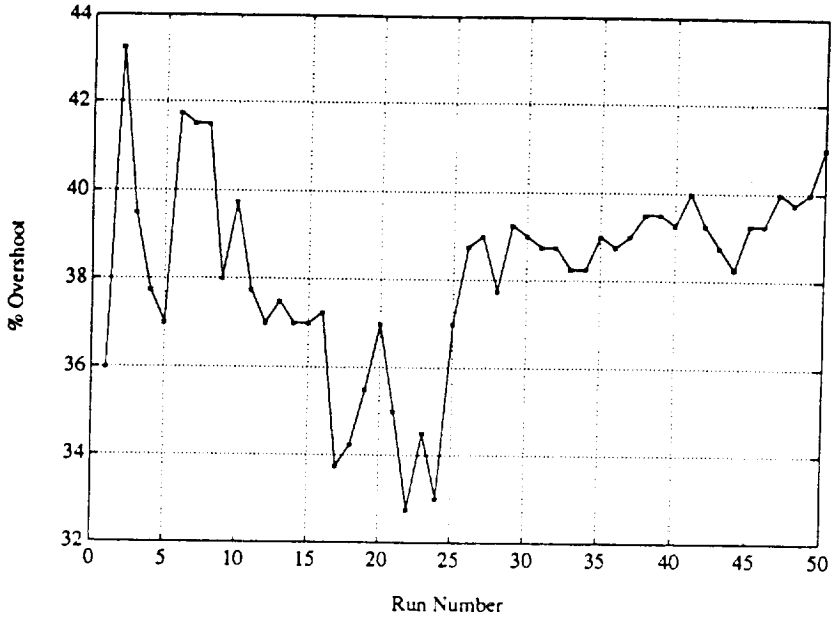

Figure 2.31: Overshoot for 50 runs from a cold start.

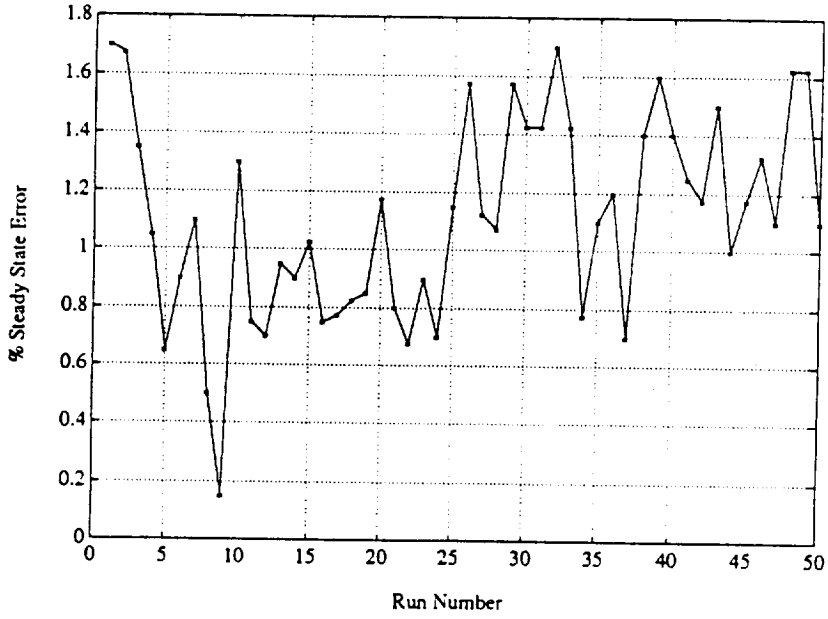

Figure 2.32: Steady-state error for 50 runs from a cold start.

runs progress, the system changes from one with a very large, fast overshoot to one with a very slow, long overshoot. This change is directly attributable to the change in hydraulic oil viscosity caused by the limit cycle in the hydraulic oil temperature controller.

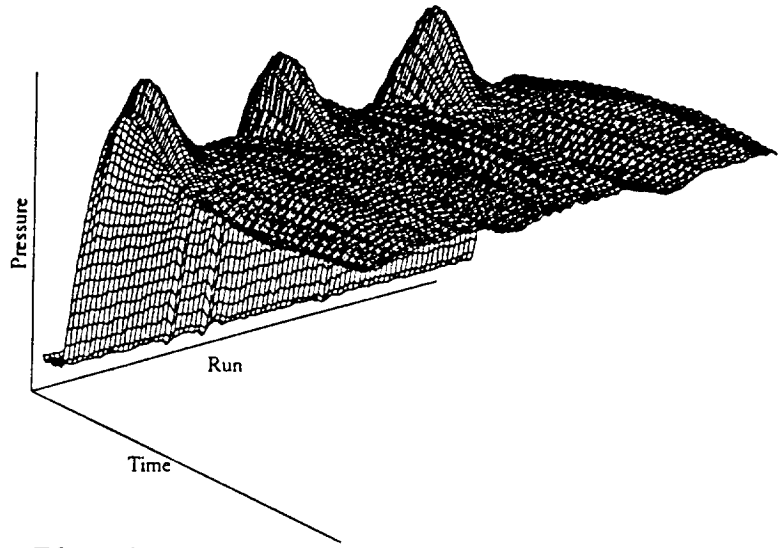

Figure 2.33: Plot showing evolution of the plastication step response over time.

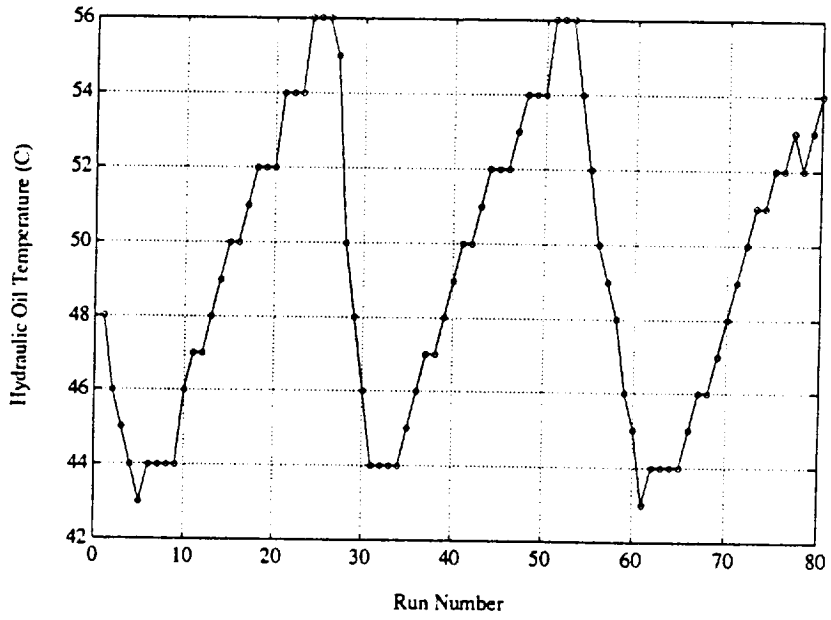

Figure 2.34: Plot showing evolution of the hydraulic oil temperature over time.

2.4 Summary

In this chapter, different measures of part quality and the factors that affect this quality were explored. The ultimate properties of any part are dependent upon both the material that makes up the part as well as the processing conditions under which the part was made. These factors must be controlled if high quality parts are to be made consistently.

Three different aspects of control of the injection molding machine were discussed, based on research performed on the Arburg machine. These experiments highlighted various aspects of the machine control problem. The research by Raimund showed the importance of measuring process variables and how process dynamics are material dependent. The research by Hsieh showed the importance of considering part quality in choosing the operating point for the machine. The research by Redlitz showed how control performance depends upon the machine operating point and that the dynamics are time-varying. One conclusion is that an automatic quality control system must incorporate process measurements and quality measurements with a controller that can adapt to changing process conditions. A method to provide this automatic quality control will be explored in Chapter 3.

3. Proposed Approach to the Quality Control Problem

3.1 Introduction

In this chapter, the quality control problem is posed as a multiple–objective optimization problem. Multi–objective optimization provides a framework for understanding the relationships between the various performance criteria and allows the operator to make decisions on how to trade off amongst the quality objectives to achieve performance *he* defines as *best*. It is an inherently interactive process, with the operator constantly making decisions.

In Section 3.2, the quality control problem is defined with respect to the components of a control system [39]. It is shown that a quality control policy can be formulated as a multiobjective optimization problem.

In Section 3.3, the important concepts of multiple–objective optimization are reviewed. When there are multiple objective functions, a tradeoff surface can be defined. This surface corresponds to those points where performance in one objective must be given up in order to achieve better performance in another. Any point on this surface represents a *nondominated* or *Pareto optimal* operating point. It can be shown that this point is an extremum of a, probably unknown, scalar function of the objective functions [40]. The most important result is that if the current operating point is *dominated*, it is possible to determine how the inputs should be changed so that the process moves towards the tradeoff surface. This *feasible direction* can be found as the solution of a linear program, which is presented in Section 3.4.

In Section 3.5, an algorithm is presented which will automatically find operating points on the tradeoff surface given an initial feasible starting point. This algorithm incorporates the feasible direction finding linear program with a line search,

so that the process will automatically iterate towards the tradeoff surface. An important feature of the algorithm is that the decision maker is a fundamental part of the algorithm. Interaction of the decision maker with the algorithm allows the operator to "steer" the system towards the operating point he defines as best. In subsequent chapters, case studies are presented which demonstrate the operation of this algorithm in the face of conflicting objectives.

3.2 Definition of the Quality Control Problem

In this work, a systems approach to quality control was taken. Any control system has the following four components:

- Performance Criteria
- Inputs and Outputs
- The System Model
- The Control Policy

The control problem is to define a control policy such that the inputs, acting through the model, cause the outputs to meet the performance criteria. In the following sections, each of these components is defined with respect to the problem of controlling quality.

3.2.1 Performance Criteria

The performance criteria for control of product quality can be summarized quite simply: *consistently maintain optimal product quality.* Product quality is measured in terms of meeting objectives with respect to many different part characteristics which must be satisfied simultaneously [1, 41]. Hsieh [32], for her experiments, defined three different quality characteristics: minimize the deviations

from the desired length in the "parallel to flow" direction, minimize the deviations from the desired length in the "perpendicular to flow" direction, and minimize the number of parts with flashing. Two different performance objectives could be associated with the each of the dimension criteria: minimize the mean deviation from the desired length and minimize the variation of the actual length.

The goal is to find and maintain operating conditions which produce the best quality. When only one objective must be met, *best* corresponds to the minimization of an objective function. Because of the *multiple* objective nature of the problem, *best* cannot be defined as easily. Improving one objective function may only be done at the expense of another; a tradeoff must be made. In multiobjective optimization, those points at which a tradeoff decision *must* be made are known as *nondominated* points. The locus of these points forms a tradeoff surface. It can be shown that any point on this surface is optimal in *some sense* in that there exists a coordinate–wise increasing function for which this point is an optimum (see [40], p 148). Any point on the tradeoff surface will be preferable to a point off of the surface (a dominated point). If the experiments described in Section 2.3.2 are considered, the operating conditions used in the *optimal* and *adjusted* runs provide performance which dominates that of the *initial* run. (See Figures [2.22] to [2.27] in Section 2.3.2.) They can be considered to lie on the tradeoff surface. Of these two sets, the experimenter preferred the third because it was felt that a loss in meeting the dimensional criteria was worth the gain in the flashing criteria. Here, the experimenter was optimizing an implicit function, her *utility function*, in choosing the best operating conditions from the nondominated set. More detailed discussion on the formulation of multiobjective optimization problems and approaches to the solution of these problems can be found in Cohon [42], Zeleny [43], Steuer [40], and Sawaragi, *et al.* [44].

In this work, it will be assumed that quality is defined as a set of objective

functions which are to be minimized simultaneously. The solution to this multiple objective optimization problem exists and is represented as a tradeoff surface corresponding to the achievable quality. The automatic quality controller has two functions. First, map enough of this tradeoff surface so that the operator can choose a feasible point at which the process can operate. This is the point that has been previously described as *best*. It will be called the *quality setpoint*. Second, continually monitor the process to make sure that the quality setpoint is being maintained *and* that it still lies on the tradeoff surface.

3.2.2 Inputs and Outputs

An open-loop manufacturing system is depicted in Figure [3.1]. For injection molding manufacturing, the process is the injection molding machine dynamics. The inputs would be any machine settings, such as desired injection velocity, melt pressure, holding time, *etc.* (see Figure [1.3]). The outputs are process variables such as screw velocities, pressures, and temperatures during the molding cycle. The material relation represents the relationship between the processing and part quality. In a sense, the process variables over the manufacturing cycle are the inputs and part quality is the output.

One aspect of this "cascaded" system which must be taken into account in controller design is that there are two time scales. Quality information is only available once per cycle (and possibly delayed due to measurement complexity) whereas process data is available throughout the cycle (at the speed of the computer sampling rate). In modeling the system and designing the control, there must be some way to integrate the information between these two subsystems. The method of integration will be suggested through the definition of the inputs and outputs.

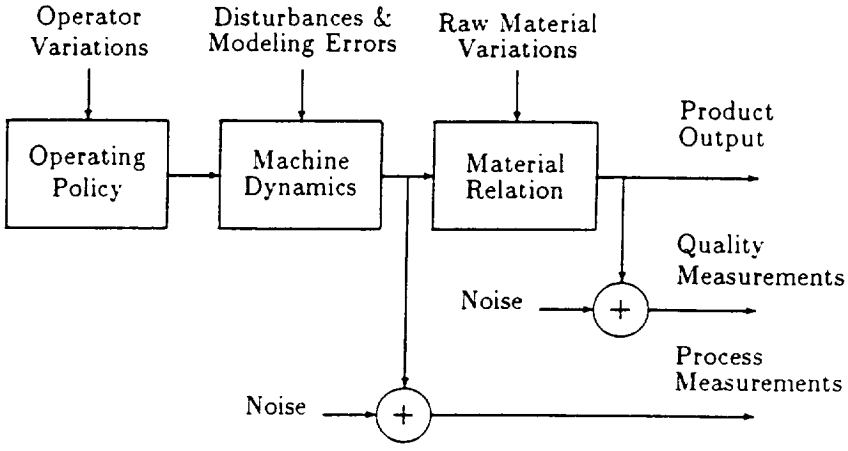

Figure 3.1: Variability in the Manufacturing Process

3.2.2.1 Inputs

For purposes of control, the inputs to the manufacturing system can be any mechanism that can affect the part quality. These inputs can be qualitative or quantitative [45]. Qualitative inputs are those inputs that can only take on a value from a unique set. The particular batch of polymer used, or a particular mold geometry are examples of these. Quantitative inputs are those that can take on any value over a continuous range. Since qualitative inputs usually can be changed only when the process stops, they will not be considered for the purpose of automatic quality control.

The set of quantitative inputs can be partitioned into two sets: those which are command signals of the process, $u_i(t)$; and those which are controller parameters, $k_i(t)$. In an abstract sense, there is no difference in these types of inputs. They can be treated in the same manner when determining the quality control policy. The partitioning is done simply to emphasize the different entry points of these inputs.

In general, all inputs will be continuous functions over the interval $[0, t_c]$. Since the injection molding process is cyclical, it will be convenient to define these inputs

with respect to the cycle number. Therefore, the m_{I_s} command signals and the m_{I_p} parameters during the l^{th} cycle are defined as

$$u_i^l(t) \in \mathcal{U} \subset \mathcal{C}[0, t_c(l)] \qquad i = 1, 2, \cdots, m_{I_s} \tag{3.1}$$

$$k_i^l(t) \in \mathcal{K} \subset \mathcal{C}[0, t_c(l)] \qquad i = 1, 2, \cdots, m_{I_p} \tag{3.2}$$

In this definition, $\mathcal{U}$ is the set of command signals, constrained by any actuator limits and $\mathcal{K}$ is the set of controller parameters, only constrained by the necessity for closed-loop stability. $t_c(l)$ is the length of the l^{th} cycle.

3.2.2.2 Outputs

There are three different types of measured outputs in the system; quality variables, quality attributes, and process variables. Of these, the quality variables and attributes are more important, since this is what the performance criteria are based on. The process variables are important because they are a direct measure of the processing conditions, which ultimately determine part quality. Each of these measured outputs and their relationship to the manufacturing model will be discussed in the following sections.

Quality Variables

Quality variables are measured on a continuous scale. Examples of these might be product dimensions or weight. If there are p_V different quality variables which are measured, the *quality variable space* can be defined by

$$q_V(l) \in \mathcal{Q}_V \subset \mathcal{R}^{p_V} \tag{3.3}$$

$q_V(l)$ is the measurement of quality for the l^{th} part (or cycle) and the set $\mathcal{Q}_V$ is the set of all achievable part qualities (good or bad). Note that each of these quality variables can only be measured once per cycle.

The performance criteria are specified in terms of the optimization of each of these quality variables (*i.e.* , each element of the vector $q_v(l)$). Also, a constraint region can be defined which represents the minimal requirements for a part to be good.

$$\mathcal{Q}_{V_T} \subset \mathcal{Q}_V \tag{3.4}$$

$\mathcal{Q}_{V_T}$ is the *quality variable target set.*

When considered as an optimization problem, $\mathcal{Q}_{V_T}$ represents the constraint set for quality. Over this constraint set, the continuous functions representing quality, $q_V(l)$, are optimized.

Quality Attributes

Quality attributes are measured on a discrete scale. This is a property which a part either *does* or *does not* have. In this sense each quality attribute can take on a discrete value. If p_A quality attributes are measured,

$$q_{A_i}(l) \in \mathcal{Q}_A = \{0, 1\} \qquad i = 1, 2, \cdots, p_A \tag{3.5}$$

In this equation, $q_{A_i}(l)$ is the quality attribute function for the i^{th} attribute. This function will only take on the value 0 or 1. The quality attribute also can only be measured once per cycle.

For part l to be good, all elements of $q_A(l)$ must equal zero. Therefore, the *quality attribute target set* is:

$$\mathcal{Q}_{A_T} = \{0\} \tag{3.6}$$

If

$$q_{A_i}(l) \in \mathcal{Q}_{A_T} \qquad \forall i \tag{3.7}$$

then the l^{th} part is said to satisfy quality with respect to the quality attributes.

When considered as an optimization problem, $\mathcal{Q}_{A_T}$ represents the constraint set for quality with respect to attributes. Since the quality attribute functions have

a discrete range, an optimum within this range cannot be defined. $q_A(l)$ are equality constraints for the optimization problem.

Process Variables

The third type of measurable output of the manufacturing system is the process variable. This is an "intermediate" variable because it is the output of the processing equipment but is the input to the material relationship (or *quality model*). These measurements, more than the machine inputs, describe what was done to the material to produce the final part properties, and ultimately the quality.

Process variables can be measured continuously through the processing cycle. Each output can be defined as a continuous function over the processing cycle, as was the input. Therefore, the m_V process variables during the l^{th} cycle are defined as

$$y_i^l(t) \in \mathcal{Y} \subset C[0, t_c(l)] \qquad i = 1, 2, \cdots, m_V \tag{3.8}$$

In the above definition, $\mathcal{Y}$ is the set of measurable outputs during the l^{th} cycle. The only constraints associated with the process variables will be those due to equipment limitations.

These variables can be measured continuously (up to the sampling rate of the available data acquisition system), so are available much more frequently than quality measurements. This allows some process variations to be detected more quickly than if only quality measurements are used.

3.2.3 The System Model

The system consists of the cascade of a process model and a material relation (or quality model). The process model is the mathematical description of the material processing, most often represented as dynamic equations (as shown in

Appendix A). The quality model is used to describe the relationship between processing and part quality. There is no strict form for this relation, due mostly to the situation-dependent definition of quality, which is the system output. An important consequence of this is that the model largely determines the type and capabilities of the control system which can be implemented.

3.2.3.1 The Process Model

For the purpose of integrating process inputs and outputs with the quality control system, this process must be modeled with respect to machine cycles. This model can be defined as

$$y^l(t) = H[u^l(t), k^l(t)] \tag{3.9}$$

where

$$H : \mathcal{U} \times \mathcal{K} \rightarrow \mathcal{Y} \tag{3.10}$$

In the above equation, H represents the system dynamics, including any automatic controllers.

3.2.3.2 The Quality Models

There are two types of quality measurements, so two models will be defined. For quality variables

$$q_V(l) = G_V[y^l(t)] \tag{3.11}$$

where

$$G_V : \mathcal{Y} \rightarrow \mathcal{R}^{p_V} \tag{3.12}$$

G_V is a vector function which maps the process variables into a real number, for each of the quality variables.

The model for quality attributes is defined as

$$q_A(l) = G_A[y^l(t)] \tag{3.13}$$

where

$$G_{A_i} : \mathcal{Y} \rightarrow \{0,1\} \qquad i = 1, \cdots, p_A \tag{3.14}$$

G_{A_i} maps the process variables into a discrete set depending upon whether the part is good (0) or bad (1), with respect to each of the quality attributes.

It should be noted that both G_A and G_V map a continuous function space into the reals. Both of these functions perform *feature detection* on the output. The feature detection will have to be determined ahead of time. Examples of this would be to relate quality to the average value or peak value of a process variable.

3.2.4 A Control Policy

The control policy is driven by the performance criteria which are defined for the system. For quality control, two performance criteria were defined:

- Find the tradeoff surface so that the operator can choose a quality setpoint.
- Ensure that the quality setpoint is on the tradeoff surface.

Given the objectives and constraints, it is possible to find a suitable control policy by solving the following multiobjective optimization problem:

$$\min_{u(t),k(t)} q_{V_i} = G_{V_i}\left[H(u(t),k(t))\right] \qquad i = 1, \cdots, p_V \tag{3.15}$$

subject to

$$G_V\left[H(u(t),k(t))\right] \in \mathcal{Q}_{V_T} \tag{3.16}$$

$$G_A\left[H(u(t),k(t))\right] \in \mathcal{Q}_{A_T} \tag{3.17}$$

$$u(t) \in \mathcal{U} \qquad k(t) \in \mathcal{K} \tag{3.18}$$

To solve this problem the p_V different functions described by Equation (3.15) must be minimized simultaneously. As was discussed before, this is the point where

the operator or process engineer must make tradeoffs in satisfying the different performance criteria. Ultimately, the solution will be a set of command signals, $u^*(t)$, and controller parameters, $k^*(t)$, which yield quality performance that lies on the tradeoff surface.

The advantage of the multiobjective formulation of the problem is that a utility function (or preference function) is not an explicit part of the solution. As will be seen when implementation is discussed, since experimental data is generated during algorithm iterations, decisions made by the operator when evaluating different operating points can be based on actual performance, not on some *a' priori* aggregate cost function. This also allows the operator to dynamically change the quality goal, as manufacturing priorities change.

3.3 A Review of the Multiple-Objective Optimization Problem

In this section, a review of multiple-objective optimization will be provided. This will include a general problem statement and pertinent definitions. The Kuhn-Tucker Conditions for Nondominance will be presented. These conditions form the basis of a feasible directions algorithm used to find the nondominated points of a given multiobjective optimization problem.

The general form of the constrained, nonlinear, multiple-objective optimization problem can be stated as follows:

$$\min_x z_i = f_i(x) \qquad i = 1, \cdots, p \tag{3.19}$$

subject to

$$x \in X = \{x \in \mathcal{R}^n \mid g_j(x) \leq 0, j = 1, \cdots, m\} \tag{3.20}$$

There are p objective functions which must be minimized simultaneously. The minimization takes place over X which is a subset of $\mathcal{R}^n$. This subset reflects the m functional constraints, $g_j(x)$, on the decision variables $x \in X$. The p objectives define a mapping from decision space into objective space, Z.

In the case of a single objective function, the decision space is mapped to the real line. Since the real line can be considered an *ordered* space, there will be only one global minimum point in objective space. There may be many solutions, *i.e.* , many points in decision space that achieve the same value in objective space, but there will be only one global minimum value of the objective function. When there are multiple objective functions, the objective space is multi-dimensional. Only a *partial* ordering can be imparted. The point in decision space, x_i^*, that minimizes $f_i(x)$ may not minimize the other $p-1$ objective functions. The concept of *nondominance* must be introduced (Note: The following definitions are due to Steuer in [40].)

> **Definition:** Let $z^* \in Z$. Then z^* is *nondominated* (*Pareto optimal*) iff there does not exist another $z \in Z$ such that[1] $z \leq z^*$ and $z \neq z^*$. Otherwise, z^* is a dominated criterion vector.

All nondominated points in Z form the nondominated set $\mathcal{N}$ (also known as the *tradeoff surface*).

Nondominance refers to objective space. The points in decision space which are mapped into the nondominated set are known as efficient points. This can be defined as follows:

> **Definition:** Let $f : X \rightarrow Z$. A point $x^* \in X$ is *efficient* if $f(x^*) \in \mathcal{N}$. Otherwise x^* is *inefficient.*

In other words, an *efficient* point is a solution to the multiobjective optimization problem which produces a *nondominated* vector of objective function values.

Even though there is typically no single point in decision space which simultaneously minimizes all of the objective functions, only one solution can be chosen. The decision maker must weigh the various options (those points on the tradeoff

[1]Let $u, v \in \mathcal{R}^n$ and $i = 1, \cdots, n$. Then $u > v$ implies that $u_i > v_i \ \forall i$; $u \geq v$ implies that $u_i \geq v_i \ \forall i$.

surface) and choose the *best* one. In multiobjective optimization problems, it is assumed that the decision maker has a *utility function* which maps the criterion vectors to a scalar *figure of merit* but the mathematical form of this function is not known. Define the utility function, U, as

$$U : \mathcal{R}^p \rightarrow \mathcal{R} \tag{3.21}$$

The following two theorems, presented without proof, show the relationship between the utility function and the nondominated points [40].

Theorem: Let U be coordinate-wise increasing.[2] Then, if z^* is optimal for U it is nondominated.

Theorem: Let z^* be nondominated. Then, there exists a coordinate-wise increasing utility function, U, such that z^* is optimal.

Therefore, if the objective function values are represented by z, and $N \subset \mathcal{R}^p$ is the nondominated set, the *best* operating point can be defined as z^* such that

$$z^* = \max_{z \in N} U(z) \tag{3.22}$$

where U is the operator's utility function. Any, not necessarily unique, decision variable x^*, which is the inverse image of z^*, is a best operating input for the process.

Typically, U is unknown and so multiobjective optimization algorithms rely on interaction with the decision maker in order to proceed. Interactive algorithms use two different general methods for finding the minimum of U: either the gradient of U is determined locally such as with the STEM algorithm [46] or the Geoffrion-Dyer-Feinberg (GDF) algorithm [47]; or the nondominated set is iteratively reduced

[2] A coordinate-wise increasing utility function implies that all objective are in maximization form and that monotonicity holds for each of them [40].

to exclude regions where the best point does *not* lie, such as in the tradeoff cut approach of Musselman and Talavage [48], the Zionts–Wallenius (Z–W) method [49], or the interactive weighted sums algorithm of Steuer [40]. Convergence of these types of algorithms can be shown under certain assumptions such as the constraint set being closed, convex, and bounded, and the utility function being continuous, differentiable, and concave. These assumptions may not hold globally and the operator must be relied upon to steer the algorithm clear of undesirable regions, such as those which appear to be Pareto optimal but are not, as in the case of local optima [50].

3.3.1 Finding the Tradeoff Surface

The Kuhn-Tucker conditions for nondominance [51] are a generalization of the necessary conditions for optimality of a scalar function. They can be stated as follows.

> **The Kuhn-Tucker conditions for nondominance**: Given the multiobjective optimization problem as stated in Equations (3.19) and (3.20), with the f_i and g_j being continuous functions, if a solution $z^* = f(x^*)$, is nondominated, then there exist multipliers $w_i \geq 0$, $i = 1, \cdots, p$ and $u_j \geq 0$, $j = 1, \cdots, m$ such that, when

$$x^* \in X \subset \mathcal{R}^n, \tag{3.23}$$

$$\sum_{i=1}^{p} w_i \nabla f_i(x^*) + \sum_{j=1}^{m} u_j \nabla g_j(x^*) = 0 \tag{3.24}$$

$$u_j g_j(x^*) = 0 \qquad j = 1, \cdots, m \tag{3.25}$$

These conditions are necessary for nondominance. They are also sufficient if all of the objective functions are convex, if X is convex, and if $w_i > 0$ for all i [42].

The above conditions are used to *detect* a nondominated point. What is needed is an algorithm which *iterates towards* a nondominated point. This can be developed using the Kuhn-Tucker conditions, *Gordan's Theorem of the Alternative*[52], and duality. First, intuitive arguments will be given to justify this.

Consider the following. If x is a dominated point, then there is at least one objective function that can be strictly decreased without increasing any of the other objective functions (or violating any constraints). Therefore, a direction vector, $d \in \mathcal{R}^n$ exists such that for $x + \epsilon d \in X$ and for some $\epsilon > 0$,

$$f_i(x + \epsilon d) < f_i(x) \tag{3.26}$$

This will hold for any d such that

$$\nabla f_i(x) \cdot d < 0 \tag{3.27}$$

When the above inner product is minimized with respect to any *one* objective function, d represents the *steepest descent* direction for that objective function. If a d exists which satisfies (3.27) for all of the objective functions, this represents a feasible direction to reduce all of the objective functions simultaneously.

This intuition can be verified rigorously using Gordan's Theorem of the Alternative. The theorem, the proof of which can be found in Mangasarian [52], is stated as follows:

Gordan's Theorem of the Alternative: For a given matrix $A \in \mathcal{R}^{m \times n}$, $x \in \mathcal{R}^n$, $y \in \mathcal{R}^m$, with $y \geq 0$, $y \neq 0$ either

$$Ax < 0 \tag{3.28}$$

has a solution, x, or

$$A^T y = 0 \tag{3.29}$$

has a solution, y, but not both.

Gordan's Theorem can be related to the Kuhn-Tucker conditions in the following way. Let A be the Jacobian matrix of the objective functions and active constraints. Let x be the direction vector, and let y be a vector with the first p-rows being the w_i's of (3.24) and the remaining $m - p$ rows are the u_j's of the active constraints. If there is no solution for (3.24), (or (3.29)), then there is a solution for (3.28) and x is the direction vector specified by (3.27).

3.4 A Linear Programming Solution to Multiobjective Quality Control

Using the arguments in the previous section, the following nonlinear program can be used to determine the descent direction:

$$\max_{d} P_o \tag{3.30}$$

subject to

$$P_o + \alpha_i(\tilde{x})\nabla f_i(\tilde{x}) \cdot d \leq 0 \qquad i = 1, \cdots, p \tag{3.31}$$

$$-\beta_j(\tilde{x})\nabla g_j(\tilde{x}) \cdot d \leq 0 \qquad j \in [1, m] \ni g_j(\tilde{x}) = 0 \tag{3.32}$$

$$P_o \geq 0 \tag{3.33}$$

$$\sum_{k=1}^{n} d_k^2 = 1 \tag{3.34}$$

where α_i and β_i are chosen to satisfy the following:

$$\alpha_i(\tilde{x})^2 \nabla f_i(\tilde{x}) \cdot \nabla f_i(\tilde{x}) = 1 \qquad i = 1, \cdots, p \tag{3.35}$$

$$\beta_j(\tilde{x})^2 \nabla g_j(\tilde{x}) \cdot \nabla g_j(\tilde{x}) = 1 \qquad j \in [1, m] \ni g_j(\tilde{x}) = 0 \tag{3.36}$$

In this nonlinear program, the direction d which maximizes P_o is the direction which is a compromise descent direction among all of the objective functions. P_o is the inner product representing the projection of d on each of the objective function

gradients. Equation (3.32) ensures that no constraint of the original problem will be violated. Equations (3.35) and (3.36) ensure that the direction chosen is not affected by the magnitude of any of the gradients. Equation (3.34) ensures that the magnitude of d does not affect the result and that the solution is always bounded.

This optimization problem was approximated by a linear program when Equation (3.34) was replaced by the following equations:

$$-1 \leq d_k \leq 1 \qquad k = 1, \cdots, n \tag{3.37}$$

$$\sum_{k=1}^{n} |d_k| \leq \sqrt{n} \tag{3.38}$$

$$(1 + \sum_{k=1}^{n} \tilde{d}_k^2) = \sum_{k=1}^{n} 2\tilde{d}_k d_k \tag{3.39}$$

These linear constraints represent a faceted approximation of the hypersphere constraint of Equation (3.34). This is illustrated by a two dimensional example of this approximation, pictured in Figure [3.2]. Equation (3.39), represented by the dashed line in the figure, is the Taylor Series expansion of Equation (3.34) around the approximate direction $\tilde{d}$. This direction is determined by first solving the linear program without Equation (3.39) then resolving the larger program using the previous result as $\tilde{d}$. This technique is called approximation programming and is described in [53]. An example is presented in Appendix C which shows the necessity of the approximation given by Equation (3.39). One point that should be emphasized is that the only constraint which is *necessary* is Equation (3.37), which ensures that the solution is *bounded* and is a *feasible direction.*

The solution to this linear program is the direction which *maximizes* the minimum gain in all objectives. This can be thought of as a "best compromise" direction. Implicitly, this formulation gives all objectives equal weight. It is possible to conceive other formulations of this linear program, through the use of additional constraints or weights such that certain objectives are minimized preferentially. It is the authors' contention that this "steering" be done interactively by the decision

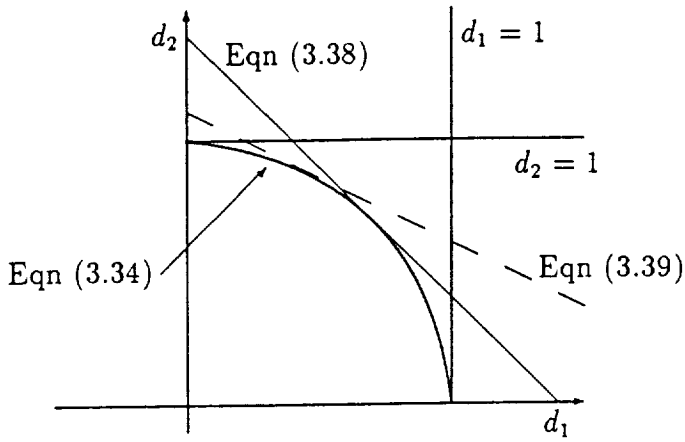

Figure 3.2: Two dimensional example showing how the quadratic constraint is approximated by linear constraints.

maker (DM). The DM will be better prepared to steer the algorithm while observing the process, rather than choosing weights *a' priori.*

In the following section, this linear program will be incorporated into an algorithm which directs the system towards the tradeoff surface.

3.5 Implementation of Automatic Quality Control

Implementation of automatic quality control can be formulated as the iterative solution of the multiobjective optimization problem described in the previous section. This tuning must be done in two phases. In the first phase, a feasible operating condition is given and the system will iteratively track to a point on the tradeoff surface. In the second phase, the *best* operating point, as defined by the decision maker, will be chosen from among the available points on the tradeoff surface. The best operating point can be defined as the one that minimizes or maximizes the utility function.

Direct implementation of the direction finding linear program requires that there be a functional description of both the objective functions and the constraints. It is proposed that objective gradient information be obtained experimentally from

the system. A set of planned perturbations of the operating point from some reference can be used, such as in evolutionary operation [54]. These data can be used to fit empirical models to the process in a region about the design center. This will provide the necessary gradient information for the objective functions. Two different types of constraints will be considered. As shown in Equation (3.16), there may be hard constraints on the objective functions which must be met by the system, corresponding to the limits on the quality variables as defined by the quality variable target set. The same experiments which provide estimates of the objective function gradients will provide estimates of these constraint gradients. There may also be other constraints not directly related to quality. One such constraint could be limits on the controller gains which correspond to the stability of the closed-loop system or of the ability of the actuator to output the required control signal. These types of constraints do not appear in the statement of the original optimization problem (Equations (3.15) to (3.18)). It may be argued that these types of constraints are artificial since they do not correspond to product quality directly, but from a practical point of view, they exclude undesirable operating conditions, reducing the region of decision space which must be searched. Any constraints of this type must have a functional description.

3.5.1 A Proposed Feasible Directions Algorithm

The following algorithm is proposed as a feasible directions method for detecting the tradeoff surface.

Step 1: Choose an initial starting point, $\hat{x}$. Form a set, $\mathcal{N}$, of known nondominated operating conditions and associated efficient points. If this information does not exist, then $\mathcal{N} = \phi$, the null set. Values for the following three parameters must also be chosen: δ, the criterion which is used to measure how well Equation (3.39) must be approximated (used in

Step 4), ϵ, the stopping threshold which detects "nearness" to the tradeoff surface (used in Step 7), and η, the step size of the line search (used in Step 9).

Step 2: Generate data around $\hat{x}$ using designed experiments. Use this data to find the gradient for each objective function. Normalize each of the estimated gradients, *i.e.* , satisfy Equations (3.35) and (3.36).

Step 3: Formulate and solve the linear program, specified by Equations (3.30) to (3.33) and Equations (3.37) and (3.38). This yields the initial feasible direction, d.

Step 4: If $\sum_{k=1}^{n} d_k^2 < 1 + \delta$, then the constraint of Equation (3.34) is approximately satisfied; go to Step 7. Otherwise continue.

Step 5: If $\|d\| > \|\tilde{d}\|$, the approximating program is diverging. Set $d = \tilde{d}$ and go to Step 7. Otherwise continue.

Step 6: Set $\tilde{d} = d$. Solve the linear program, specified by Equations (3.30) to (3.33) and Equations (3.37) to (3.39). Go to Step 4.

Step 7: Analyze the linear program solution. If $P_o > \epsilon$, a stopping threshold, use the feasible direction to find a new nondominated point; go to Step 8. If $P_o \leq \epsilon$, $\hat{x}$ is on the tradeoff surface; go to Step 11.

Step 8: Let $\tilde{x} = \hat{x}$.

Step 9: $\tilde{x} = \tilde{x} + P_o \eta d$. Run an experiment with $\tilde{x}$ as the operating condition and generate $\tilde{q}$, the value of the performance objectives at $\tilde{x}$. η is a step-size for the search algorithm chosen by the experimenter. P_o, the projection value returned by the linear program, is used to adjust the step size so that smaller steps are taken as the algorithm gets nearer the

tradeoff surface, when objectives begin to conflict. (In this step, any line search algorithm may be employed. This method was chosen because it is relatively conservative.)

Step 10: Evaluate the performance of $(\tilde{x}, \tilde{q})$ against $\mathcal{N}$. If $\tilde{q}$ is nondominated, update $\mathcal{N}$ and keep searching along d (go to Step 9); otherwise continue.

Step 11: If $\tilde{q}$ is dominated by an alternative in $\mathcal{N}$, stop the line search. Allow the decision maker to evaluate the alternatives within $\mathcal{N}$ and determine a new operating point to begin evaluations from. Call this desired operating point $\hat{x}$. The algorithm can be stopped at this point or a new iteration can be started by returning to Step 2.

The algorithm has three phases. In the first phase, Steps 1 and 2, the system is probed to generate a model of the response surface. This model provides the objective function gradients which are used to determine the search direction. In the second phase, Steps 3 to 7, a linear program is solved which determines the feasible direction, d, which will move the system to an operating point on the tradeoff surface. In the third phase, Steps 8 to 11, a line search is carried out until no improvement in performance is obtained in the current feasible direction. Once the feasible direction has been exhausted, the operator can be presented with the current set of nondominated operating conditions. The operator will choose one, and a new iteration can begin or the algorithm can be stopped if the operator is satisfied with system performance.

There are three parameters, selected by the user, which control the performance of the algorithm. δ is used as a threshold for determining when Equation (3.34) is approximately satisfied. For the case studies in this thesis, it was found that $\delta = 0.05$ worked well. ϵ was used as a "conflict threshold". P_o is the minimum value of the projection of the feasible direction vector onto each of the

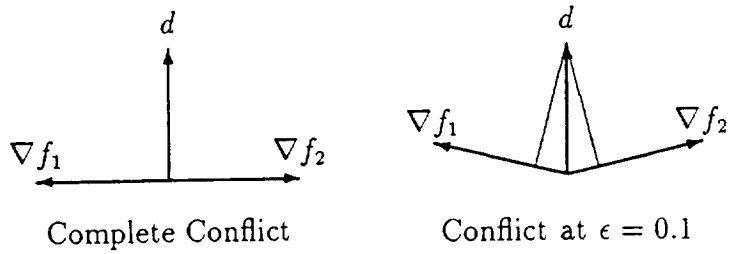

Figure 3.3: Complete conflict versus ϵ–conflict.

gradients. The magnitude of P_o is a function of the magnitude of the most conflicting gradients, the magnitude of d, and the cosine of the angle between them. If two objectives are in complete conflict, *i.e.* , they are 180^o apart, then $P_o = 0$. Assuming the magnitude of all vectors is unity, $P_o \leq 0.1$ signifies that the angle between the feasible direction and the most conflicting gradients is greater than 85^o, or the two most conflicting gradients are greater than 170^o apart. This is illustrated in Figure [3.3]. In this thesis $\epsilon = 0.1$ was used as the threshold for detecting the tradeoff surface. Finally, η is selected to determine the step–size taken during the line search. Selection of this parameter is completely dependent upon the particular problem. It should be of the same order of magnitude as the perturbations used in the experiments of Step 2.

There are two different methods by which the operator *steers* the algorithm, both of which occur during Step 11. The simplest form is that the operator request that certain regions of the tradeoff surface be explored by his choice $\hat{x}$ and a selection of the step–size for input perturbations and the line search. The algorithm will explore the local region around $\hat{x}$, with "local" being determined by the step–size chosen.

The second method of steering is where the operator selectively disables objective functions. If it is found that a region is encountered where the marginal value of a gain in one objective function is very low, this objective can be disabled and all optimization will take place with respect to the others. The objective will continue to be monitored, and it's behavior will be reflected in the construction of the nondominated set, but it will not affect the feasible direction of search. This is implemented directly via the α_i weighting factors. If $\alpha_i = 0$, the i^{th} objective function is removed from the linear program.

3.5.2 Properties of this Algorithm

This algorithm is a primal method, in that it directly attacks the original problem, in this case through the application of the Kuhn-Tucker conditions. One advantage of primal methods, that is particularly useful in this research, is that each point generated in the search process is feasible. Therefore if the search is terminated early, a feasible and probably nearly optimal operating point is still obtained [55]. When the implementation is to be in real-time, feasibility is especially important because an *infeasible* point may correspond to an operating condition that is dangerous to the operator or the equipment. Operator safety is of paramount importance.

One performance measure typically applied to optimization algorithms is that of convergence. Insight into the convergence properties of the multiobjective optimization algorithm can be gained by considering the following theorem[55]:

Global Convergence Theorem: Let A be an algorithm on X, and suppose that given x_0 the sequence $\{x_k\}_{k=0}^{\infty}$ is generated satisfying

$$x_{k+1} \in A(x_k)$$

Let a solution set $\Gamma \subset X$ be given, and suppose,

(i) all points x_k are contained in a compact set $S \subset X$

(ii) there is a continuous function f on X such that

(a) if $x \notin \Gamma$, then $f(y) < f(x) \forall y \in A(x)$

(b) if $x \in \Gamma$, then $f(y) \leq f(x) \forall y \in A(x)$

(iii) the mapping A is closed at points outside Γ.

Then the limit of any convergent subsequence of $\{x_k\}$ is a solution.

Condition (iii), requiring closedness of the map, is the most important. The multi-objective algorithm can be considered as the sequential application of two maps, a direction finding map and a line search map. Closedness of the algorithm requires that *both* of these maps be closed. Closedness is defined as follows:

Definition: A point–to–set mapping $A : X \rightarrow Y$ is said to be *closed* at x if the assumptions

(i) $x_k \rightarrow x, x_k \in X$

(ii) $y_k \rightarrow y, y_k \in A(x_k)$

imply

(iii) $y \in A(x)$

If A is a direction finding algorithm, then the sequence $\{y_k\}$ represents the direction vector for each of the points in the sequence $\{x_k\}$. For A to be closed, it is required that the direction not change suddenly. This cannot be guaranteed by the linear program of Equations (3.30) to (3.33) and (3.37) to (3.39). The direction *will* change suddenly when a formerly inactive constraint becomes active. For this reason, this algorithm is not globally convergent. (This problem is inherent to feasible direction algorithms in general.)

When the constraints are analytic, this "direction change" problem can be avoided by implementing penalty or barrier functions as part of the objective. This

way, gradient information from inactive constraints is included. This is not possible for this problem due to the nature of the quality attribute constraints, as described by Equation (3.17).

Convergence is really important only if an optimization algorithm is running autonomously. For the multiobjective optimization algorithm, a decision maker is an intrinsic part of the algorithm, (*i.e.* , Step 11). In this case, the decision maker provides the "steering" that would be provided by the penalty or barrier function in other feasible direction implementations. Furthermore, it is the decision maker who decides when the algorithm is to stop iterating. When the algorithm "finds" a point that the decision maker considers *best*, iterations are stopped and the process is run using the associated operating conditions. The operator ultimately decides when the optimization has converged.

3.5.3 Example

The following example is intended to show the use of the algorithm on a very simple example. The objective functions are analytic, so there is no need to generate gradients experimentally.

Consider the following optimization problem:

$$\min_{x \in \mathcal{R}^2} f(x) = \begin{bmatrix} (x_1 - 1)^2 + (x_2 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 \end{bmatrix} \tag{3.40}$$

The Jacobian of $f(x)$, evaluated at $\tilde{x}$ and then normalized such that Equation (3.35) holds, is

$$F(x) = \begin{bmatrix} \alpha_1 2(x_1 - 1) & \alpha_1 2(x_2 - 1) \\ \alpha_2 2(x_1 + 1) & \alpha_2 2(x_2 + 1) \end{bmatrix} = \begin{bmatrix} \alpha_1 \nabla f_1(x) \\ \alpha_2 \nabla f_2(x) \end{bmatrix} \tag{3.41}$$

The analytic formulation of the objective functions allows Steps 1 and 2 of the algorithm to be skipped. Then, the following tableau represents the linear program which must be solved to determine the direction of the Pareto boundary. (Note: this

tableau represents the initial LP to be solved. The constraint due to Equation (3.39) is not included.)

0	-1	0	0	0	0	0	0	0	0	0	0	0
0	1	$F_{11}(\tilde{x})$	$F_{12}(\tilde{x})$	$-F_{11}(\tilde{x})$	$-F_{12}(\tilde{x})$	1	0	0	0	0	0	0
0	1	$F_{21}(\tilde{x})$	$F_{22}(\tilde{x})$	$-F_{21}(\tilde{x})$	$-F_{22}(\tilde{x})$	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	1	0
$\sqrt{2}$	0	1	1	1	1	0	0	0	0	0	0	1

The objective functions are two paraboloids, with global minima at (1, 1) and (-1, -1), respectively. Due to symmetry, the Pareto optimal boundary is the line connecting them. In terms of the geometry of the objective surface, this line forms the set of all points for which the individual objective gradients point in opposing directions.

The algorithm was run using three different initial points. A plot showing the objective function contours and the iterations is given in Figure [3.4]. The zig-zagging across the Pareto boundary is a result of the algorithm not being closed. This behavior would be detected by the operator, in practice.

3.6 Summary

In this chapter, the automatic quality control problem has been defined formally. This led to the statement of the problem as a multiple objective function optimization problem. The iterative solution to this problem forms the basis of feedback control of part quality.

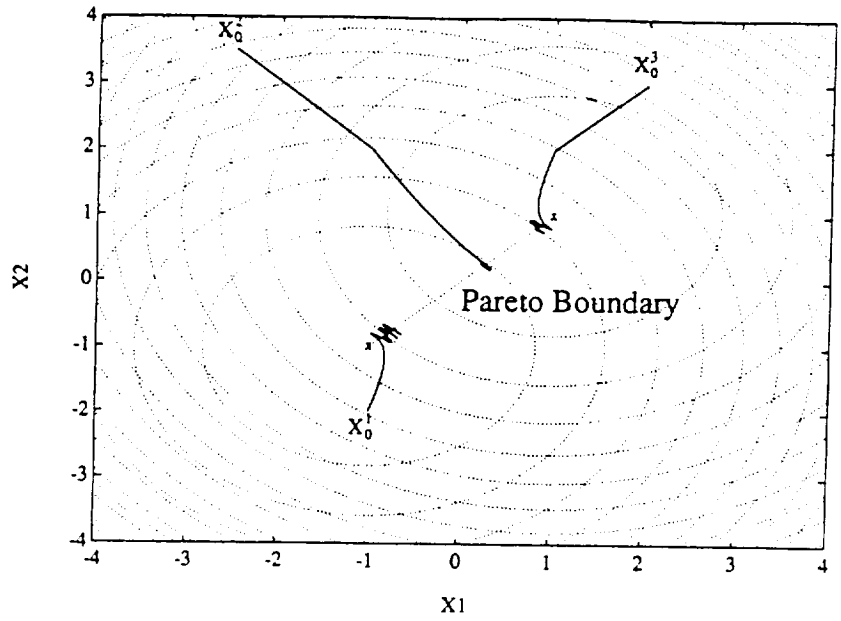

Figure 3.4: Contour plot of objective functions showing algorithm iterations.

A multiobjective algorithm for implementation of quality-based controller tuning was presented. This was a feasible directions algorithm based on the Kuhn-Tucker conditions for nondominance.

Implementation of this algorithm on the injection molding machine is the focus of the rest of this work. In the next chapter, details of the equipment on which the implementation was carried out, are presented. In the two chapters following that, the controller tuning and quality control case studies are presented.

4. Experimental Setup

Any control system will depend, to some degree, on the specific machine being controlled and the equipment used to control this system. In this chapter, specific details of the injection molding machine used, available sensors, actuators, and the control computers will be discussed.

4.1 The Injection Molding Machine

The injection molding machine which will be used is an Arburg Allrounder, Model #221-175-350. The specifications for this machine are given in Table [4.1]. This machine is equipped with actuators and sensors which can be used to monitor and control the process. This equipment will be discussed in the following sections.

4.1.1 Actuators

The actuators available are used for machine sequencing, feedback control, or as process settings. These actuators are the solenoid valves, the barrel heater bands, the nozzle heater band, the hydraulic motor control valve, and the servovalve. The capabilities of each of these actuators will be described below.

Solenoid Valves

Each of the machine phases is initiated by activating or deactivating-activating solenoid controlled valves. These valves "route" hydraulic fluid through the hydraulic system. They are merely switches, and as such, cannot be used for feedback control. Sequencing of the machine through the phases does depend upon the proper switching of these valves. In Table [4.2], these valves are listed showing the phases where they are active (ON) and inactive (OFF).

Table 4.1: Arburg Allrounder Machine Specifications

Shot Size	4.64 ounces	
Clamp Force	40 tons (toggle-type)	
Injection Force	12.2 tons	
Screw Speed	Min 10 RPM	Max 320 RPM
Screw Torque	145 foot-pounds	
Nozzle Pressure	6 tons	
Nozzle Retract	7.08 inches	
Effective Stroke Max	5.7 inches	
Clamp Stroke	Min 2.4 inches	Max 7.9 inches
Open Daylight Max	19.7 inches	
Close Daylight	Min 5.9 inches	Max 11.8 inches
Platen Dimensions	13.5 x 9.8 inches	
Tie Rod Spacing	8.66 inches	
Tie Rod Diameter	1.8 inches	
Max Mold Weight	275 pounds	
Max Movable Mold Weight	165 pounds	
Mold Size	8.66 x 9.8 inches	
Mold Thickness	Min 5.9 inches	Max 11.8 inches
Hydraulic Ejector Force	2.7 tons	
Hydraulic Ejector Stroke	2.36 inches	
Barrel Heater (3)	1750 Watts	
Nozzle Heater	200 Watts	
Total Connected Load	45 Amps	

Table 4.2: Solenoid Valve Activation Table

Valve	Plastication	Injection	Holding	Cooling
System Pressure	OFF	ON	ON	X
Mold Close	ON	ON	ON	ON — OFF
Mold Open	OFF	OFF	OFF	OFF — ON
Holding Pressure	OFF	OFF	ON	OFF
Screw Forward	OFF	ON	ON	OFF
Screw Retract	OFF	OFF	OFF	ON (briefly)
Screw Rotate	ON	OFF	OFF	OFF
Carriage Forward	ON once, to initiate the first cycle			
Carriage Retract	ON once, to end last cycle			

Barrel and Nozzle Heater Bands

The heater bands are used for control of the barrel and nozzle temperature. Each of these heater bands has its own closed-loop control system. Underneath each heater, set in the barrel, is a thermocouple which provides temperature feedback to the associated heater band. Each closed-loop controller provides the operator with a knob to set the desired barrel temperature for that heater band. The controller is an analog control loop, possibly a PID controller (observation of the controller in action shows that there is some anticipatory response to the sensed temperature). The control action is a simple ON/OFF. There is no readout of actual temperature; a simple LED readout indicates when the temperature is within $\pm 5^0 C$ of the set temperature. It is not possible to adjust temperature setpoints automatically, due to the construction of these particular heater band controllers, although this is not a limitation in general. If changes to the temperature setpoints is necessary, this will be accomplished by notifying the operator.

Hydraulic Motor Control Valve

The hydraulic motor control valve is used to control screw rotation speed. This is done by regulating the flow to the hydraulic motor which rotates the screw, using a ball valve. In typical operation, this is set by the operator through a knob and graduated dial. On the dial, 0 corresponds to the minimum screw speed and 5 corresponds to the maximum speed (see Table [4.1]). A stepper motor and gear system has been implemented to allow computer control of screw speed. One limitation of this actuator is that it can only be turned when the screw rotate solenoid valve is off (the mechanical construction of this valve does not allow it to be turned while under pressure). The result is that any feedback control scheme which incorporates changing the screw speed can only make one control action per cycle, when the machine is not rotating.

Servovalve

The servovalve is used to control the flow of hydraulic fluid to the piston actuating the plasticating screw. This actuator controls any linear motion of the screw as well as hydraulic pressure applied by the screw during holding. This actuator is the heart of the injection molding machine control system. During plastication, this valve controls the flow of hydraulic fluid out of the piston chamber, thereby controling the recovery rate of the screw and the pressure developed in the molten polymer. During injection, this valve controls the flow of fluid into the piston chamber, thereby controlling the injection velocity and pressure. During holding, this valve is primarily used to control the pressure on the polymer in the mold. Computer control of this actuator is accomplished through a D/A converter and amplifier from the microprocessor.

4.1.2 Process Sensors

There are five sensors on the injection molding machine: hydraulic pressure, melt pressure, screw position, melt temperature, and screw RPM. Each of these will be discussed below.

Hydraulic Pressure

A hydraulic pressure sensor is located in the hydraulic piston chamber behind the screw. This sensor can be used to record pressure on the screw during plastication, injection, and holding. This sensor has a range of 0-15000 PSI and the output is connected to an A/D converter on the microprocessor.

Melt Pressure

An additional pressure sensor is located in the nozzle of the barrel. This sensor measures the pressure on the molten polymer during plastication, injection,

and holding. The sensor range is 0-3000 PSI and is also connected to an A/D converter on the microprocessor.

Screw Position

A linear potentiometer, attached to the screw, is used to measure screw position. It's range is 0-15 cm and is also connected to an A/D converter.

Melt Temperature

A type-K thermocouple has been mounted in the nozzle for use in measuring temperature of the molten polymer. This temperature will be different from that measured by the thermocouples used by the heater band controllers. This sensor has been used in previous experiments, but is not currently installed.

Screw RPM

Screw RPM is sensed by a magnetic pickup on the screw. This is then fed to an analog gauge to provide the operator with an indication of screw speed (it is not very accurate). Hardware has been designed to feed screw speed, using a similar magnetic pickup, to the control computer. This hardware has not been installed.

4.2 The Control Computer

Control of the plastic injection molding process is done using two computers: an Intel 80286-based personal computer, used for process supervision, and a Motorola 68000-based microprocessor, used for real-time control and data acquisition [36, 37]. Together, these computers comprise the plastic injection molding machine quality evaluation and control system (also known as the *Diatomic Spurtalyzer*).

A typical cycle proceeds as follows:

1. Injection parameters are entered into the PC. They are then passed down to the 68000 via a serial line. There are three types of parameters: termination variables, process setpoints, and controller gains. These will be described later.

2. Upon receipt of the parameters, the 68000 begins the injection cycle. Process data is stored and the process is controlled. For the most part, the PC is dormant during this stage.

3. During the injection cycle, the 68000 can signal the PC to prompt the operator for information. An example of this is information related to the operator's inspection of the molded part. Any interaction will be specific to a particular mold being used.

4. Upon completion of the injection cycle, the 68000 passes the logged process data to the PC. All data is stored in files on the disc, and any data pertinent to the optimization is retained in memory to be operated on by the algorithm. During any operations on the PC, the 68000 is dormant.

Communication between the two processors takes place via a serial line. The protocol is a pure *master/slave* relationship. When one processor is running, the other halts until a message is received, then the roles are reversed. Three types of variables are passed from the PC to the 68000 at the initiation of an injection cycle:

- Termination variables – these are used to end a particular phase. The following five are used:
  - Shotsize – the amount of polymer to be plasticated, in centimeters (cm) of screw displacement.
  - Pullback – the distance the screw is retracted after plastication to allow decompression of the polymer, also in cm.

  - Cushion - the amount of polymer that is to remain in the barrel at the end of injection. During holding, this polymer will be used to fill any volume due to shrinkage. This is also measured in cm of screw displacement.
  - Holding Time - the length of time to maintain holding pressure, measured in seconds.
  - Cooling Time - the length of time after holding, that the part is to remain in the mold before injection, also measured in seconds.
  - Maximum Plastication and Injection Time - These are limits that are used as a fail safe in case a problem occurs during the cycle. They can not be altered interactively.

- Process Setpoints - these are the setpoints that the controllers must maintain during a particular phase. The following are used:
  - Plastication Melt Pressure
  - Injection Melt Pressure
  - Holding Melt Pressure
- Controller Gains - the plastication, injection, and holding phases each are controlled by a PID controller. During each cycle, these gains are passed to the 68000.

During a set of experiments, any one of the above parameters can be changed from one cycle to the next. The parameters which were changed depended upon the particular optimization being done. For example, during plastication controller tuning, only the PID gains were altered, all other parameters were held constant. During quality experiments, only process setpoints were varied.

The machine also had other inputs which were not under computer control, namely screw speed and barrel temperature. These settings are dependent on the

polymer being used. In all of the experiments to be described, this polymer was polypropylene. For these experiments, screw speed was held constant at 380 RPM. The barrel temperature could be set independently in four zones along the barrel. These temperatures were 220 C, 220 C, 220 C, and 205 C, from the nozzle to the hopper respectively.

Programming on the 68000 was done in C. All real-time control code was written on the PC, cross-compiled using Uniware software [56], then downloaded to the 68000. Programming of the PC was done via Matlab [57]. All algorithmic programming was done using custom Matlab functions, as well as those provided within Matlab. Interaction with the injection molding machine, such as serial port communication and control, was done using the Matlab–to–C interface. The user operated the injection molding machine by calling a Matlab function called "inject". Input parameters were those described above, and logged data was returned as Matlab matrices, converted into the appropriate units.

5. Multiobjective Controller Tuning

The algorithm was applied to the tuning of two PID controllers, one used in the plastication phase and one used in the injection phase. These case studies are relevant for two reasons: injection molding requires the operator to control many different parameters to produce a good part [29, 30]; and the process is cyclical, so the operator (or algorithm) has the opportunity to evaluate performance and update operating conditions between cycles. Use of the algorithm allows the loop to be closed around process performance. This is shown in Figure [5.1].

The PID regulator is probably the most widely used controller in the process industry. Different methods have been proposed for tuning these regulators, such as Ziegler–Nichols [58]. Despite this, it is common experience to have the regulator poorly tuned, yielding a less than adequate process response [59]. Also, these tuning methods may optimize some aspect of the response which is not appropriate for a particular application. It is *not* the author's intention to provide "yet another controller tuning tool", but rather to explore the multi–objective nature of process control.

An *anti-reset windup* implementation of this controller was chosen because the control output is limited to $\pm 5V$. Feedback to the integral term is used to "turn off" the integrator when the actuator is in saturation. If this does not occur, when the process has a constant error due to actuator limitations, the integrator "winds up", resulting in poor performance once the system comes out of saturation.

The block diagram for this controller is shown in Figure [5.2]. The controller transfer function is

$$\begin{aligned} V(z) &= K_P \frac{(z-(1-\frac{K_I}{K_P}))}{(z-1)} R(z) \\ &- K_P(1+K_D) \frac{(z-(1-\frac{K_I}{K_P}))(z-\frac{K_D}{1+K_D})}{z(z-1)} P_m(z) \end{aligned} \tag{5.1}$$

In this equation, the operator z is the *z-transform* variable.

Notice that the command input, $R(z)$, does not pass through the differencing term of the controller. This prevents spikes due to discontinuities in the command input from passing through the system. Also, since the limiter is a nonlinear element, it is not represented in the transfer function. It should be noticed that this controller has three adjustable parameters, two zeros and a gain. One zero is dependent only on the ratio of K_I to K_P and the other only on K_D. The controller gain is dependent on the product of K_P and K_D. These relationships will be important when analyzing the results of the multiobjective controller tuning. Another important point is that if control design is thought of in terms of choosing the zeros and gain of the controller, there will be more than one set of PID gains that will achieve the same closed-loop response. (It may be possible to use this "degree of freedom" to optimize an additional objective at no "cost".)

Three different case studies are presented. In the first case study the plastication PID controller was tuned in simulation. An analytical model for plastication has been derived, but this model contains many unknown and processing-dependent parameters. Because of this, a model was identified from input/output data. Details of the modeling and identification are presented in Section 5.1.

Using the identified model, the multiobjective algorithm was applied, in simulation, to the tuning of a PID regulator for control of the plastication phase. Two objective functions were defined; one which was directed towards good trajectory tracking and one which was directed towards avoiding controller saturation. Details

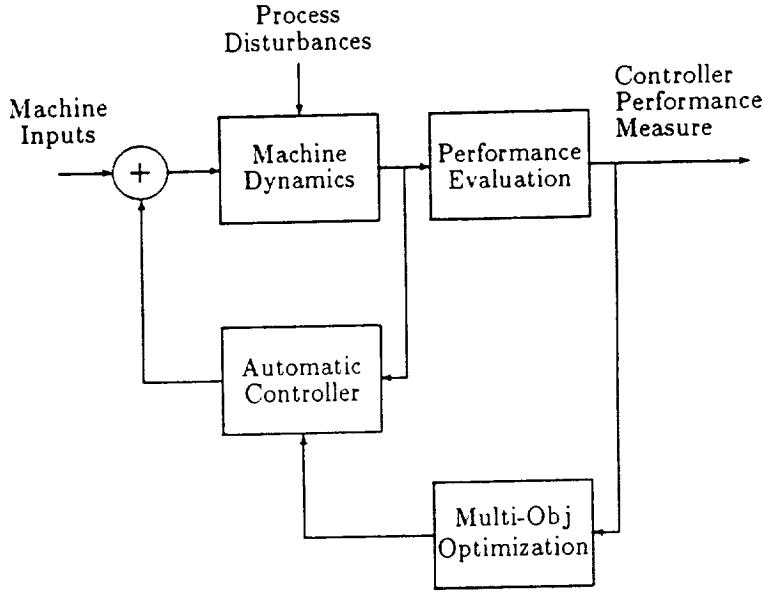

Figure 5.1: The algorithm used for performance monitoring and control.

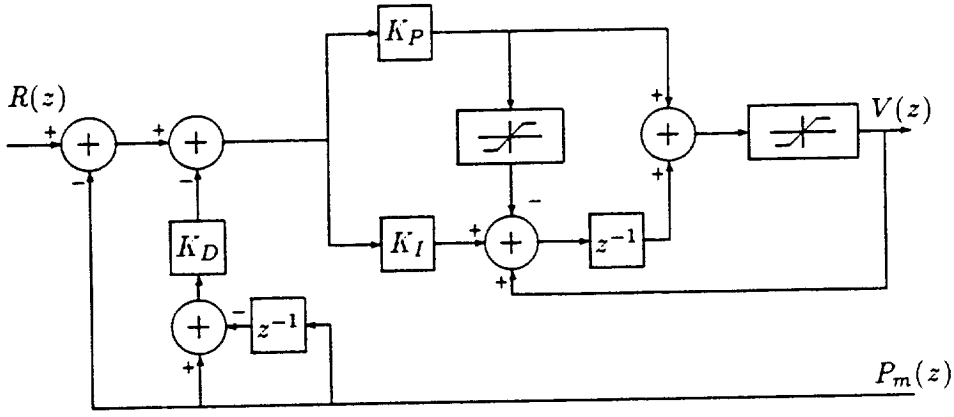

Figure 5.2: PID controller block diagram.

of this simulation, and algorithm results are presented in Section 5.2.

The plastication controller was then tuned *on-line*. Essentially, the simulation of the process was replaced by the Matlab function which controlled the injection molding machine, as discussed in Chapter 4. With respect to the algorithm, there was no difference in the two implementations, other than the fact that the real data was noisy. Details of these results, and a comparison with those obtained from the simulation, are presented in Section 5.3.

Finally, the injection controller was tuned on-line. In the case of injection, the dynamics are very dependent upon the mold, so new gains must be scheduled for every mold. Two different molds were used for the quality control case studies, which are discussed in Chapter 6. For the spiral mold, a slight modification of the plastication gains provided adequate control. For the ASTM four cavity mold, new gains had to be found. Implementation details are no different than that for plastication. Results of this study are presented in Section 5.4.

5.1 Plastication Modeling and Identification

During plastication, a prescribed amount of polymer is melted and deposited in front of an injecting mechanism. Tight control of the processing conditions of the polymer during plastication is very important since this has a large impact on final part quality [29, 30, 35, 26, 28]. In this study, melt pressure is controlled as a single-input single-output system. This affects both how quickly polymer is accumulated as well as the melt temperature.

5.1.1 Analytical Modeling of Plastication

The plastication dynamics consist of the cascade of fluid dynamics associated with the hydraulic portion of the injection system, a mass balance associated with

the motion of the screw, and fluid dynamics associated with the pumping and accumulation of the polymer. The following linear state equation can be used to describe the system:

$$\dot{Z}_p = \begin{bmatrix} -\frac{K_m\beta_m}{V_m} & -\frac{A_m\beta_m}{V_m} & 0 \\ \frac{A_m}{m} & -\frac{b}{m} & -\frac{A_H}{m} \\ 0 & \frac{A_H\beta_H}{V_H} & -\frac{K_{Pp}\beta_H}{V_H} \end{bmatrix} Z_p + \begin{bmatrix} \frac{K_\omega\beta_m}{V_m} & 0 \\ 0 & 0 \\ 0 & -\frac{K_{Xp}\beta_H}{V_H} \end{bmatrix} U_p \qquad (5.2)$$

In the above equation, the states are defined as $Z_{p1} = P_m$, $Z_{p2} = V_s$, and $Z_{p3} = P_H$. The control inputs are defined as $U_{p1} = \omega$ and $U_{p2} = X_v$. (All of the symbols used in this equation are defined in Table [5.1].) An output equation must be defined for this state-space model. If melt pressure is the output, then the output equation is:

$$Y_p = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} Z_p \qquad (5.3)$$

A block diagram of this system is shown in Figure [5.3]. A detailed derivation of this model is contained in Appendix A.

The dynamics of the servovalve are dependent on the internal construction of the valve. On the Arburg machine, this is a three-port valve. An approximate model of the valve dynamics is [61]

$$X_v[s] = \frac{K_v}{s(\tau_v s + 1)} V[s] \qquad (5.4)$$

The input to the servovalve is a voltage, V, and the output is the position of the spool, which regulates hydraulic flow out of the screw piston chamber. The integral term of the transfer function is a result of the fact that there is no internal feedback within the valve because of the solid spool [62]. The first order term, with time constant τ_v, is an approximation of the rest of the valve dynamics, including the windings driving the pilot stage.

The problem with this model is that the parameters are difficult to determine, and can be related non-linearly to system states. For example, both hydraulic oil

Table 5.1: Symbols used in the plastication dynamic equations.

A_H	hydraulic pressure effective area
A_m	melt pressure effective area
b	approximate viscous friction on screw
K_m	melt pressure empirical gain
K_{Xp}	servovalve spool position flow gain
K_{Pp}	servovalve hydraulic pressure flow gain
K_v	servovalve gain (voltage to spool position)
K_ω	screw speed empirical gain
m	approximate mass of screw and polymer system
P_H	hydraulic pressure
P_m	polymer melt pressure
U_p	plastication input
V_H	hydraulic oil volume
V_m	melt volume
V_s	recovery rate (screw linear velocity)
V	servovalve input voltage
X_v	servovalve spool position
Y_p	plastication output
Z_p	plastication state vector
β_m	melt bulk modulus
ω	screw RPM (screw rotational velocity)
β_H	hydraulic oil bulk modulus
τ_v	servovalve time constant

and polymer bulk moduli will be dependent on the overall machine temperature. For this reason, it was decided to develop an empirical transfer function for the system.

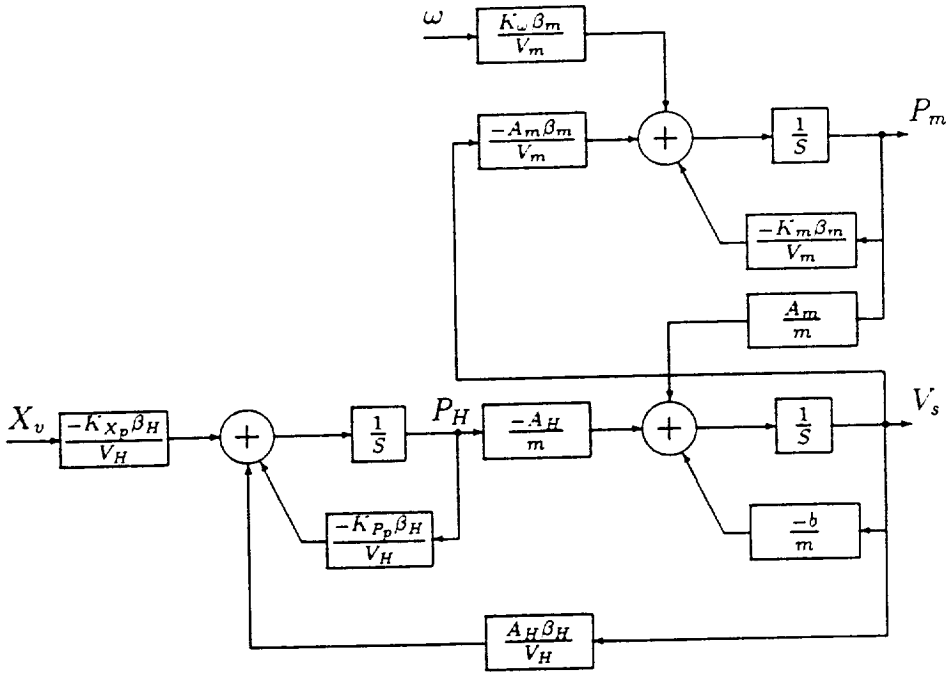

Figure 5.3: Plastication dynamics block diagram.

5.1.2 Identification of a Plastication Model

Initial experiments showed that the open-loop system was marginally stable. Input/output data had to be collected with the system under closed-loop control. This situation can lead to problems with the identification. Specifically, the controller can induce correlation between the output and input, which can adversely affect identification algorithms [63]. To avoid this, a *relay with hysteresis* was used as the stabilizing feedback element. This element will stabilize a class of systems and provides very good data for identification and because the control was *nonlinear*, the correlation problem is less significant. Årzen, in [64], discusses in detail the use of relays as feedback elements. Årzen presents a describing function analysis which shows that the relay will induce limit cycle behavior in the closed loop system. Because of the relay, the input during this limit cycle is a square wave with a period related to the system dynamics.

The nominal melt pressure reference was chosen to be 3000 psi. Other references were tested to determine the sensitivity of the model to this parameter. The *relay with hysteresis* has two parameters which were changed routinely: relay amplitude and hysteresis width. These changes would generate limit cycles with different periods and amplitudes, possibly exciting different dynamics in the system. These parameter changes provided a large ensemble of input/output data sets which were used to evaluate the accuracy of the identified model.

The sampling period used for these experiments was 12.5 ms. This choice was dictated by the ability of the 68000 controller to compute the PID control law within one sampling period.

Input and output data for the injection molding machine, using this feedback are shown in Figures [5.5] and [5.6]. Model identification was done using the Identification Toolbox in Matlab [63]. The following ARMA model was chosen as the

representation.

$$A(q)P_m(k) = B_1(q)V(k) + B_2(q)\omega(k) \tag{5.5}$$

In this equation, q is the unit time delay operator and k is the sample index.

Various model orders were tried. The selection of the final model was done by comparing output from the model, under the same relay feedback as the real process. Also, models which displayed approximate pole–zero cancellation were deemed to be of too high an order. Model identification was done using over 40 data sets. In all cases, the seventh-order model, with respect to the voltage input, proved to be the best. It was possible to identify only one coefficient with respect to screw speed. This input had to be a step, due to physical considerations. Because of this, it did not provide enough excitation for more than one coefficient to be estimated. Also, the poles of the empirical models were practically identical for the different data. The zeros varied quite a bit. In almost all cases, the zeros were non-minimum phase, and they typically appeared as complex-conjugate pairs.

After an analysis of process data, the following two-input, one-output ARMA model was chosen,

$$\begin{aligned} P_m(k) &= 2.380P_m(k-1) - 1.908P_m(k-2) + 0.570P_m(k-3) \\ &+ 0.796P_m(k-4) - 1.914P_m(k-5) + 1.392P_m(k-6) \\ &- 0.316P_m(k-7) + 0.371V(k-1) - 0.974V(k-2) \\ &+ 1.652V(k-3) - 1.616V(k-4) - 0.026V(k-5) \\ &- 0.285V(k-6) + 0.275V(k-7) + 0.00731\omega(k-1) \end{aligned} \tag{5.6}$$

In the above equation, k represents the sample index. The current output, $P_m(k)$, is a function of previous input and output values.

The poles and zeros of this transfer function are shown in Figure [5.4]. There are no zeros associated with screw speed since only one coefficient is estimated for this input. In this figure, confidence intervals representing one standard deviation around each of the poles and zeros are shown. The confidence intervals for the

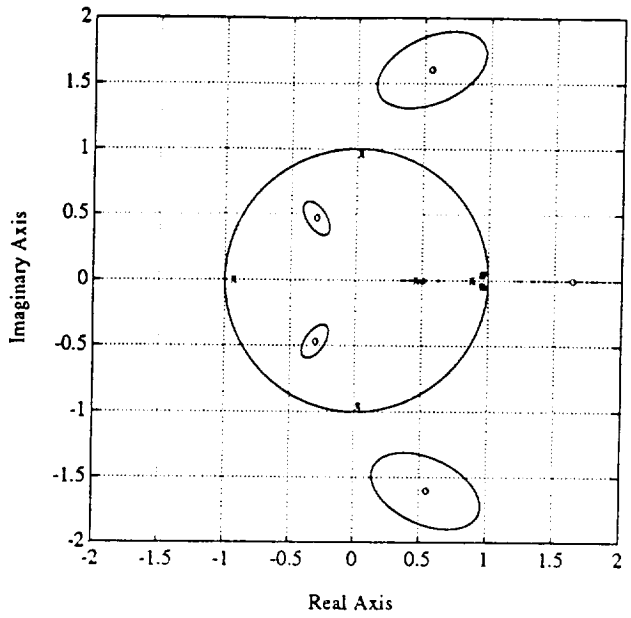

Figure 5.4: Estimated poles and zeros with confidence intervals.

poles are extremely tight, indicating a very good fit of the denominator polynomial. Those for the zeros are not quite as good, especially for the non-minimum phase zeros.

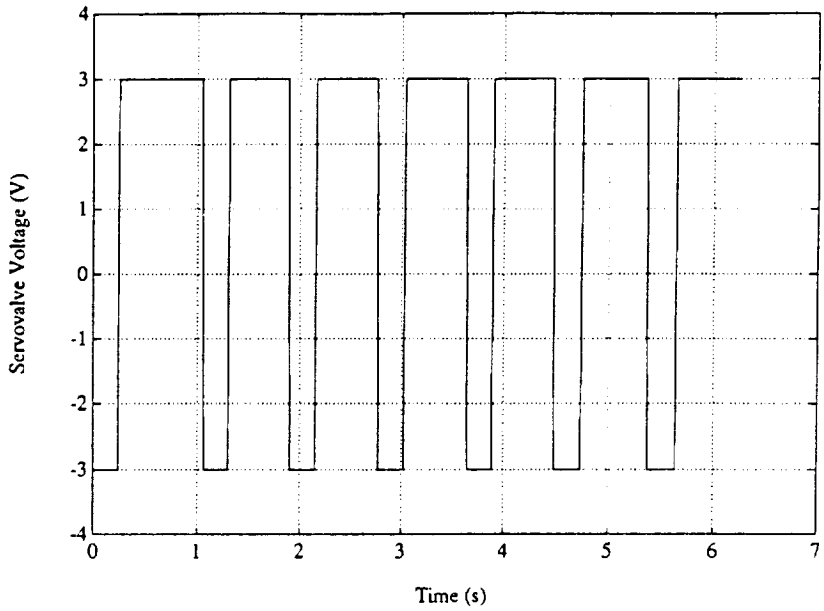

Figure 5.5: Process input produced by relay controller.

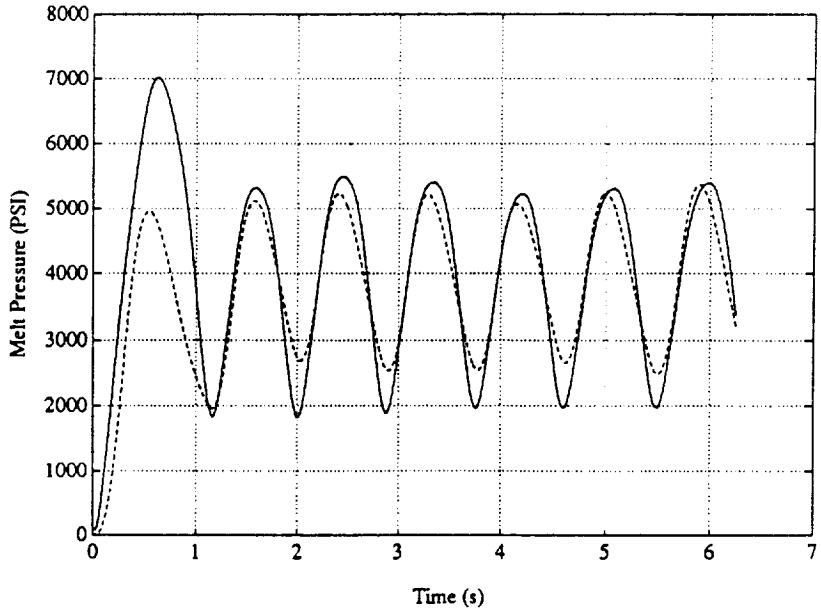

Figure 5.6: Comparison of process output and simulation output (simulation is dashed-line).

5.2 Plastication Controller Tuning: Simulation Results

For this study, two objective functions were chosen: minimize the integral time absolute error [65] and minimize control saturation. The following equations were used in conjunction with a discrete-time simulation of the process:

- Minimize Integral Time Absolute Error (ITAE)

$$\text{ITAE} = \sum_{k=1}^{T_f} k|R(k) - P_m(k)| \tag{5.7}$$

- Minimize Maximum Control Deviation

$$\|V\|_{16} = \left[\sum_{k=1}^{T_f} V(k)^{16}\right]^{1/16} \tag{5.8}$$

Equation (5.8) is the "16–norm" and approximates taking the maximum value of the control voltage over the interval but is a continuous and convex function, which is numerically better conditioned [50].

5.2.1 Simulation Conditions

The simulation scenario was that the closed-loop system consisting of the transfer function, Equation [5.6], and the controller, Equation [5.1], was given a step input of 3000 PSI for 4 seconds (320 samples). The resulting system response was evaluated with respect to the performance objectives described above. A two level factorial design with center point was used for experimental perturbations to determine gradients. For experiments run using this design, each parameter can take on two distinct values, its nominal value *plus* the desired perturbation, and its nominal value *minus* the perturbation. An additional experiment was run with all parameters unperturbed: the centerpoint. This yielded a total of *nine* input level combinations. This particular experimental design allows the estimation of all first-order interactions between inputs. Since only linear terms were to be estimated,

this design also provided enough degrees of freedom so that "goodness-of-fit" of the linear model could be examined. This information would help determine whether a particular gradient estimation is valid [45, 5].

A root-locus design was done to find an initial set of stabilizing gains to be used as an initial feasible point for the multiobjective optimization algorithm. Approximately 100 iterations of the algorithm were performed to map the tradeoff boundary and associated efficient points shown in Figures [5.7], [5.8], [5.9], and [5.10]. Each iteration involved determining the gradients of the objective functions, determining a feasible descent direction using the linear program, and performing a line search in this direction. Each algorithm iteration will evaluate the system over a minimum of ten input level combinations, so many nondominated points can be generated in a single iteration. It is important to note that nothing prevented the algorithm from generating a set of gains which produced an unstable closed-loop system. It relied entirely on the choice of the performance criteria and the fact that small perturbations of gains were used to determine parameter changes.

5.2.2 Discussion of the Simulation Tuning Results

Analyzing the results of the optimization yields some interesting insights. Even though the tradeoff curve is continuous, there seems to be two disconnected efficient regions in decision space. This bifurcation corresponds to the point where ITAE is 500 and the control norm is 0.6. This disconnected behavior of the efficient set occurs in multi-objective optimization problems where the objective functions are functions of ratios of the decision variables [43]. In the two different regions, there is a completely different, parametric relationship between the decision variables and the tradeoff surface. This behavior may not be anticipated by an operator. In each performance region, a parametric relationship seems to exist between performance and controller gains. This could be used to give the operator *one knob* to select

desired performance rather than have him select three gains.

The tradeoff curve shows a lower limit of 200 for the ITAE objective function, no matter how much control is used. This provides a good idea of the marginal value of allowing more control action during any one cycle. At the same time, an upper limit of the control norm is 1.1. Physically, this means that any desired, achievable ITAE performance can be met without any actuator saturation.

In this example, all efficient gains yielded an asymptotically stable closed-loop system.

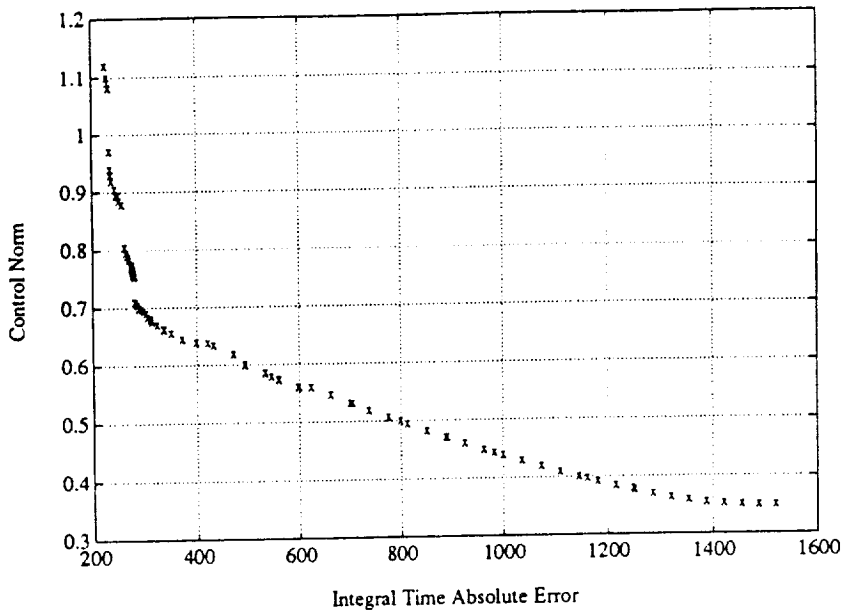

Figure 5.7: The tradeoff boundary (nondominated set) found using the multiobjective algorithm.

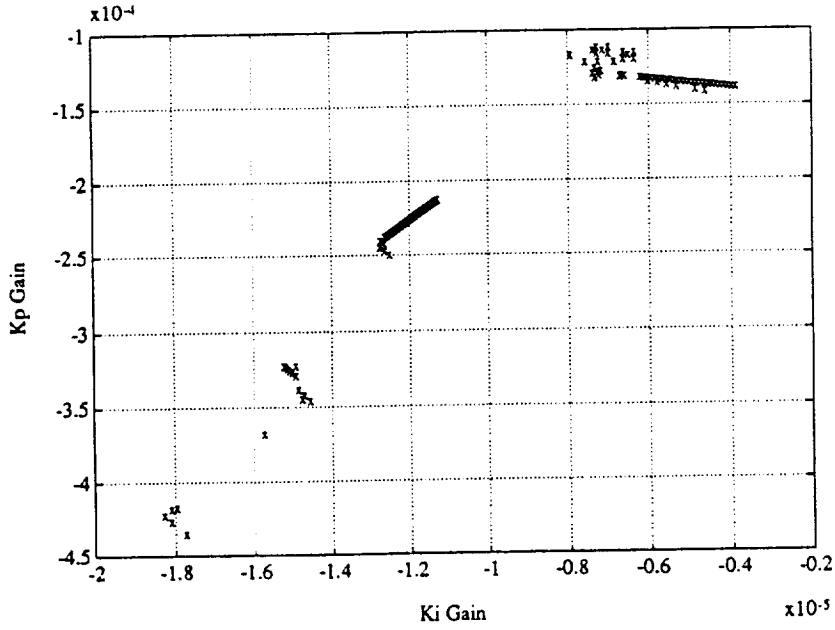

Figure 5.8: Efficient points: K_I vs. K_P.

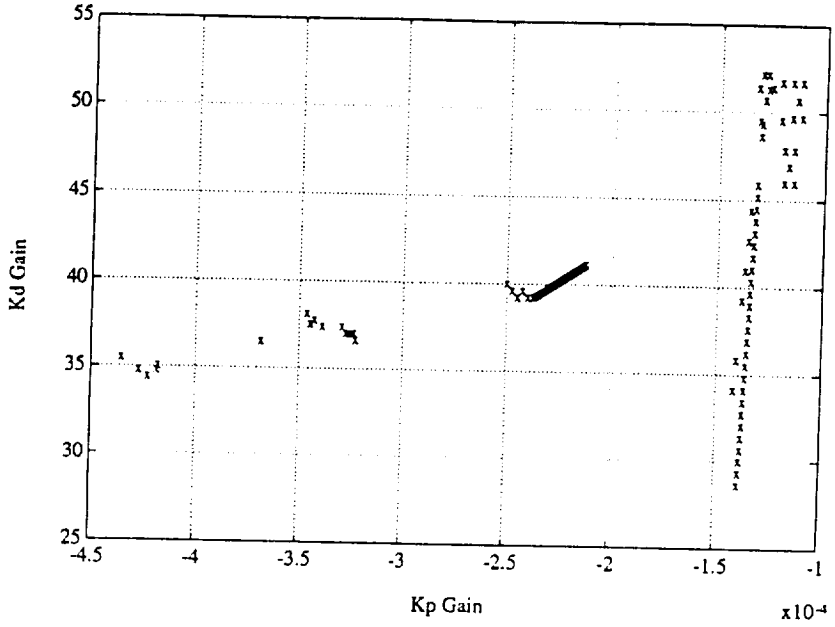

Figure 5.9: Efficient points: K_D vs. K_P.

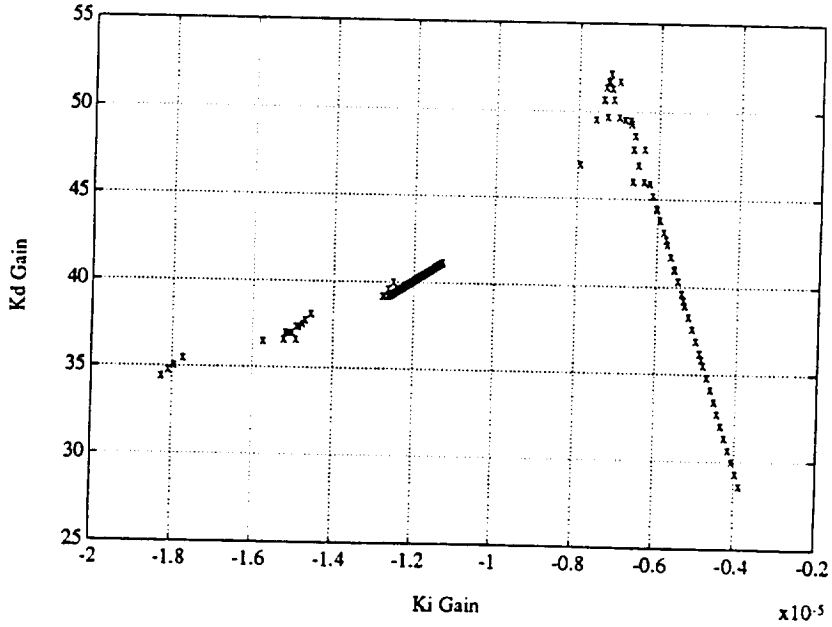

Figure 5.10: Efficient points: K_D vs. K_I.

5.3 Plastication Controller Tuning: Experimental Results

The procedure for experimental tuning of the controller was identical to that of the simulation. In fact, the only implementation difference was that the Matlab function which simulated the system was replaced by the Matlab function which accessed the 68000 microprocessor, as described in Chapter 4.

The machine operating conditions used for these experiments are given in Table 5.2. Plastication was initiated with the nozzle empty and terminated when the desired shotsize was achieved. During plastication, data was collected using a sampling rate of 80 Hz, the same as was used for the identification experiments. The barrel temperature could be set independently in four zones. The temperature setpoints are given in Table 5.2 sequentially from the nozzle to the hopper. All of these parameters were held constant for the experiments. Any one of them could affect the empirically-determined transfer function. These parameter settings represent typical values. In the future, it may be useful to develop a family of transfer functions for different values of these parameters.

A typical injection session proceeds as follows. The operator enters the desired initial operating point. The system then runs a series of injection cycles, with no operator interaction, where this operating point is perturbed according to some experimental design. After this experimentation phase, the performance data is processed to yield the gradient information. The linear program is then solved to determine if the current operating point is nondominated. If not, the algorithm proceeds with the line search, generating new PID gains and running the process. Assuming no problems occur which require operator intervention, the line search continues as long as non-dominated points (with respect to known performance) are generated. Once a dominated point is generated, the operator is presented with information about the known nondominated operating points and associated efficient points. In this example, with only two objective functions, this information

Table 5.2: Nominal Input Settings

Polymer	Polypropylene
Screw Speed	380 RPM
Barrel Temperature	200 C, 200 C, 200 C, 185 C
Shotsize	5.0 cm
Plastication Setpoint	3000 PSI

is presented as a plot of ITAE versus the control effort (such as Figure [5.7] or [5.11]). The operator then selects which of these points is the desired operating point. The associated gains are passed down to the microprocessor controller and the machine continues cycling. He is also given the opportunity to allow the algorithm to improve on this point. If he so chooses, the algorithm will begin the experimentation/line search process again.

In the initial phases of running a particular process, the algorithm is used to map out the tradeoff surface. This experimentation phase is used to gain an understanding of possible machine performance, suitability of the chosen objective functions, suitable experimental perturbations of the inputs, a suitable step–size for the algorithm, and other implementation–specific details. Once the tradeoff surface and efficient points are established, this knowledge provides diagnostic information about the process. For example, if the system can no longer maintain a previously achievable operating point, or if the efficient points change, there may be some underlying physical cause which should be investigated.

5.3.1 Discussion of the Experimental Tuning Results

Figures [5.11] shows both the experimentally determined tradeoff boundary (x) and the appropriate region from the simulation results (o). The process performance used to initialize the algorithm is also shown (*). Figures [5.12] through [5.14] show the associated efficient points and the initial PID gain values.

The experimental session was started by initializing the process with the gains shown by the "*" in Figures [5.12], [5.13], and [5.14]. A set of experiments were run, which were based on a two-level factorial design plus the center point. A level change consisted of a $\pm 5\%$ gain change. The result was that 9 experiments were run, which were then used to evaluate the gradient of the performance measure with respect to the gains. This constituted Steps 1 and 2 of the algorithm described in Section 3.5.1. $\mathcal{N}$ was initialized as ϕ, the null set. Once the initial search direction was found (Steps 3 to 7), the algorithm proceeded with the line search in gain space (Steps 8 through 9). Once the line search failed, the current set of nondominated points was presented to the operator. A new point of exploration was chosen and the algorithm was restarted at Step 2. After approximately 200 injection cycles, the performance tradeoff boundary shown in Figure [5.11] was produced.

In Figures [5.15] through [5.22], the process input and response is shown for the initial point, as well as for three points along the tradeoff surface, marked **A**, **B**, and **C**. [3] From a practical point of view, process performance for each of these points is adequate, so there is no reason not to use the gains that yield the minimum ITAE value.

The following observations can be made about the experimental results:

- The simulation and experimental results are in good agreement. The tradeoff curves and PID gains have the same basic shape and values.
- The experimental results show a lower achievable ITAE value. This is due to the fact that nonlinearities in the real system, which are not modeled by Equation (5.6), allow a much faster rise time.
- In general, the control norm has a higher value. This is a result of the noisiness of the actual signals. This does not affect the ITAE criterion because the

[3]Note: No relationship between run number and performance should be inferred from any of the tradeoff surface plots.

pressure signal has a much better signal-to-noise ratio.

- When compared to the simulation results, much less of the tradeoff surface has been "mapped". This is due more to practical constraints; the low control norm/high ITAE system gains yielded a response that could possibly damage equipment, so these regions were not explored.

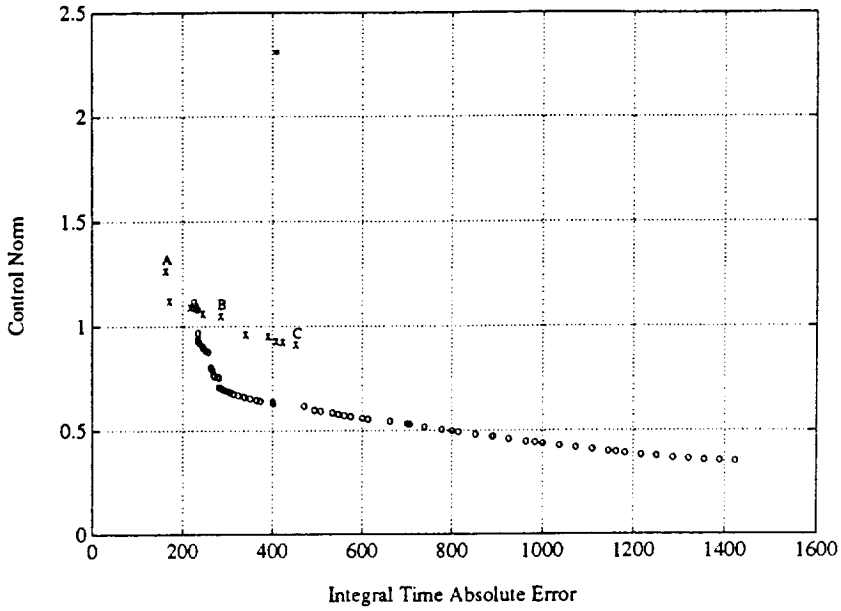

Figure 5.11: A comparison of the experimentally determined tradeoff boundary (x) with the "simulation" boundary (o).

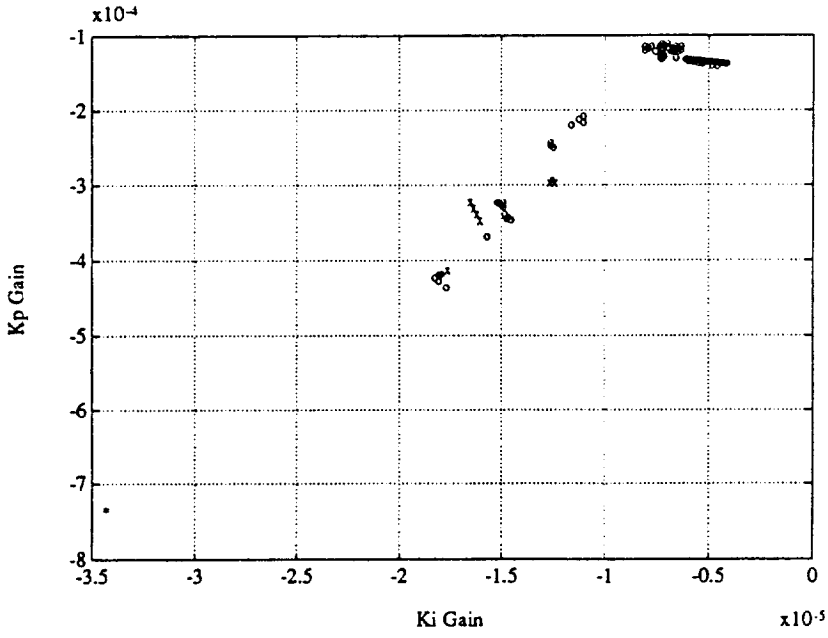

Figure 5.12: Comparison of efficient points: K_I vs. K_P

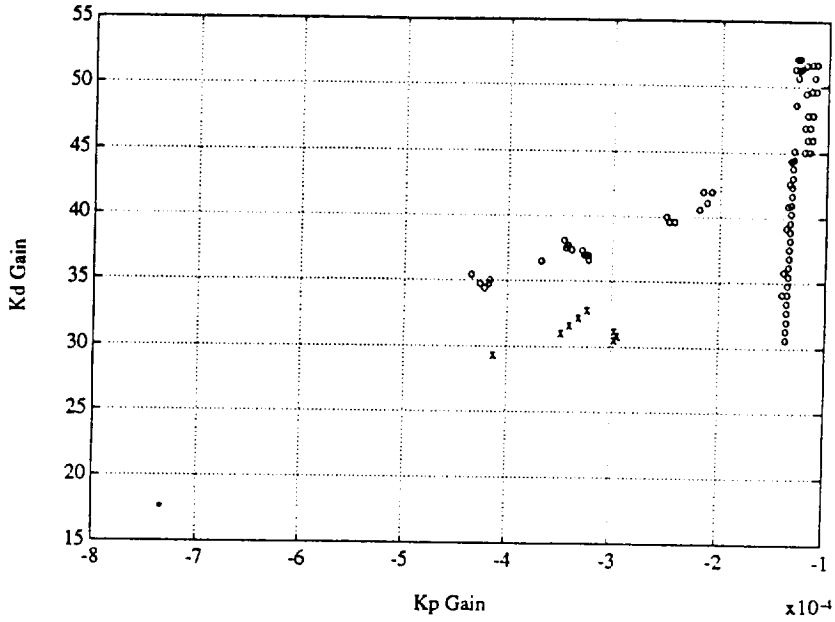

Figure 5.13: Comparison of efficient points: K_P vs. K_D

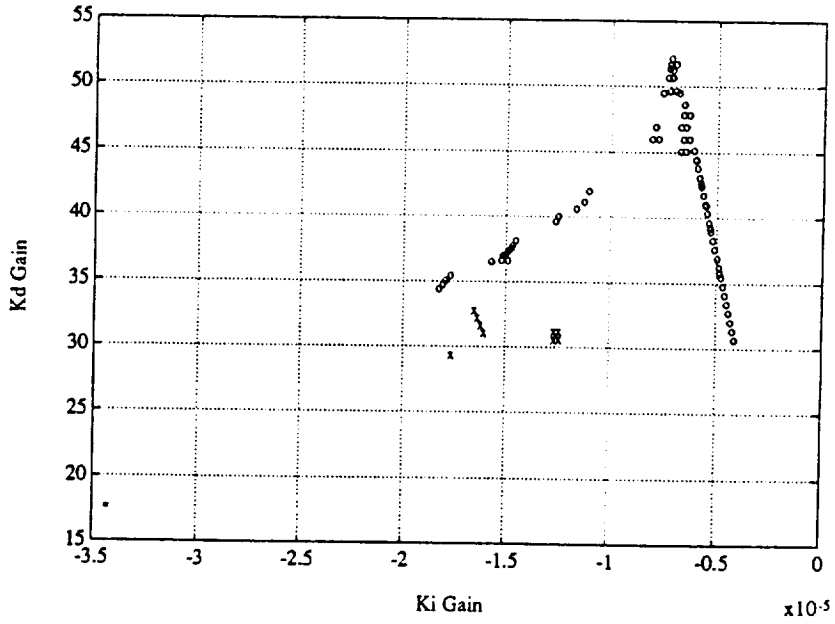

Figure 5.14: Comparison of efficient points: K_I vs. K_D

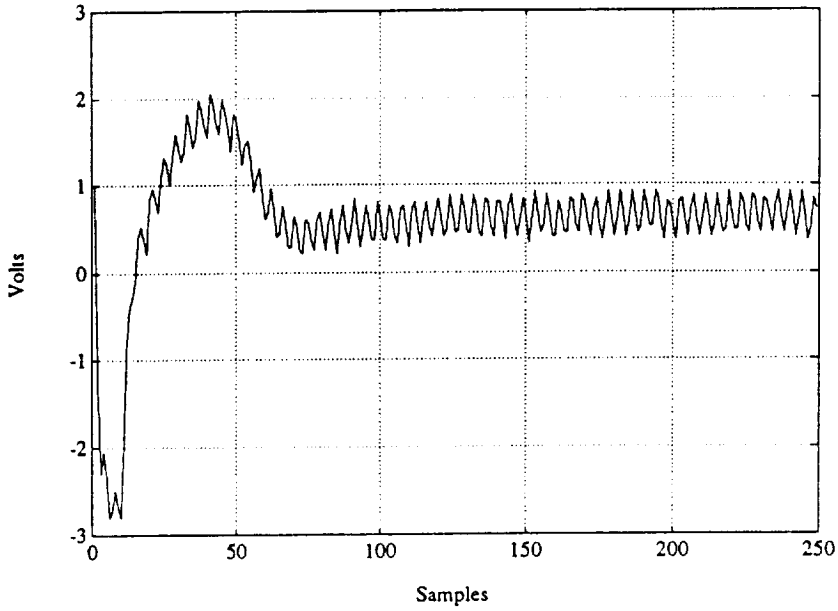

Figure 5.15: Control effort using initial gains.

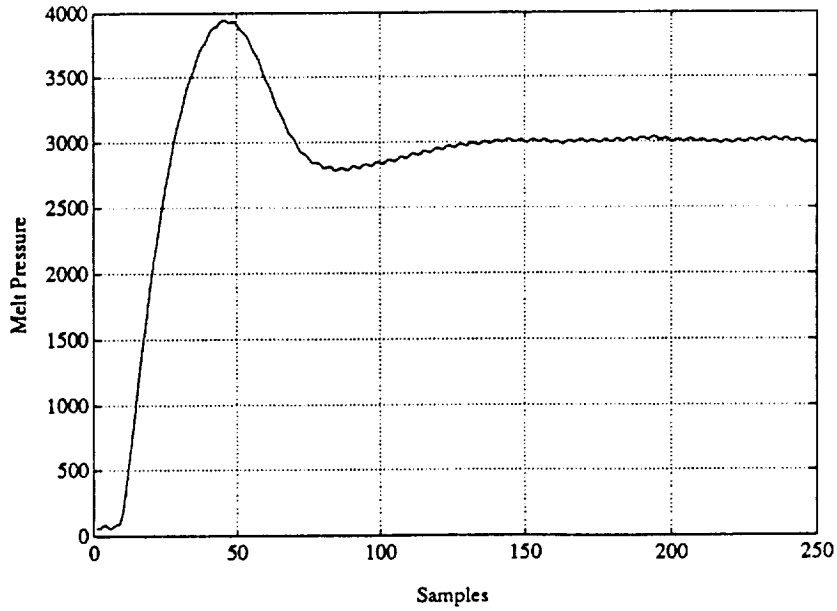

Figure 5.16: Process response using initial gains.

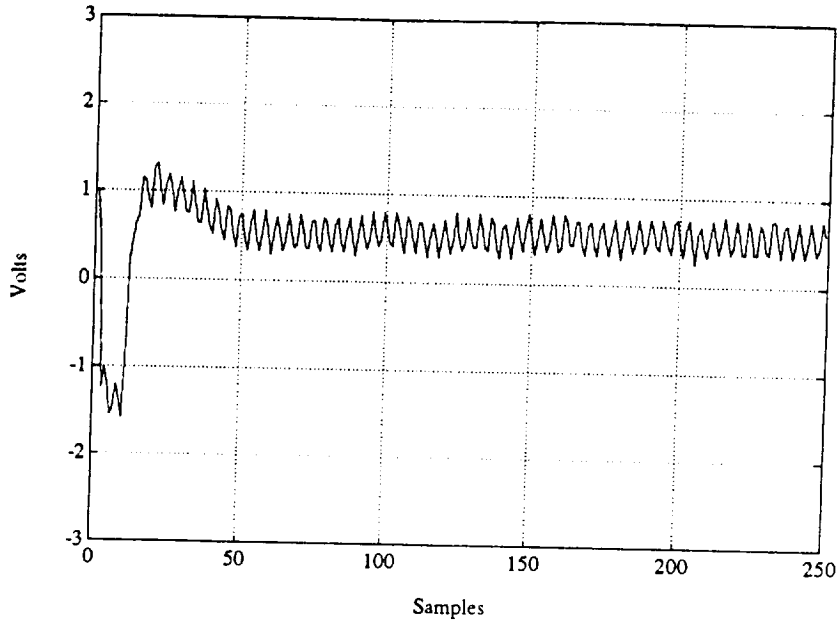

Figure 5.17: Control effort at tradeoff surface point A

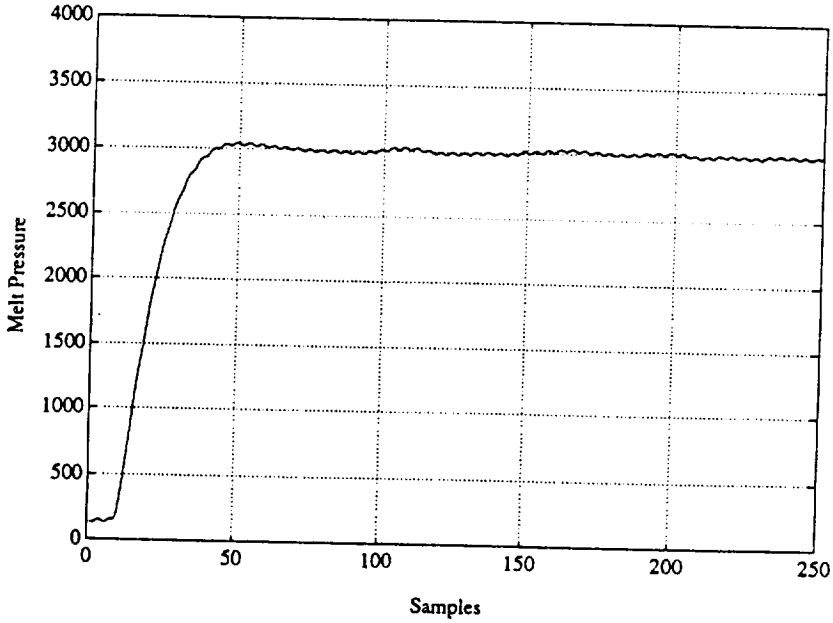

Figure 5.18: Process response at tradeoff surface point A.

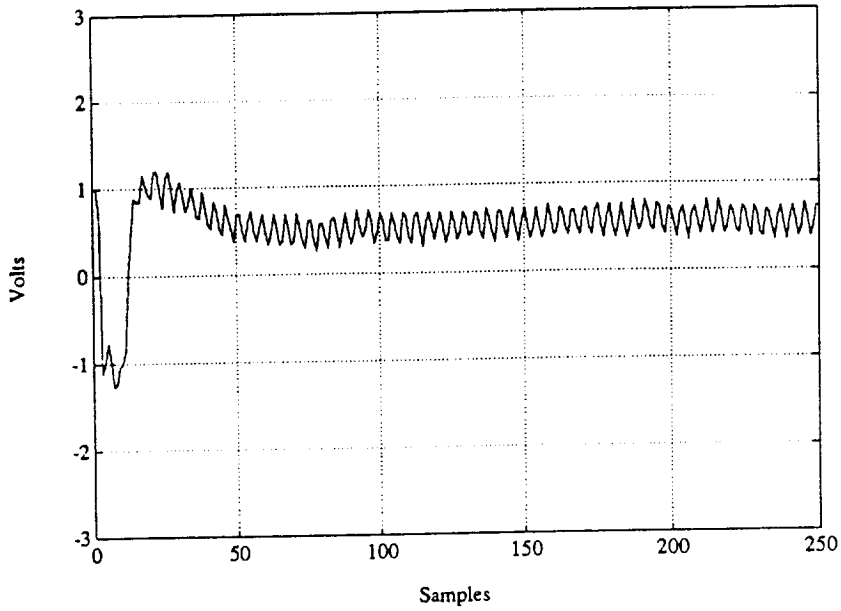

Figure 5.19: Control effort at tradeoff surface point B.

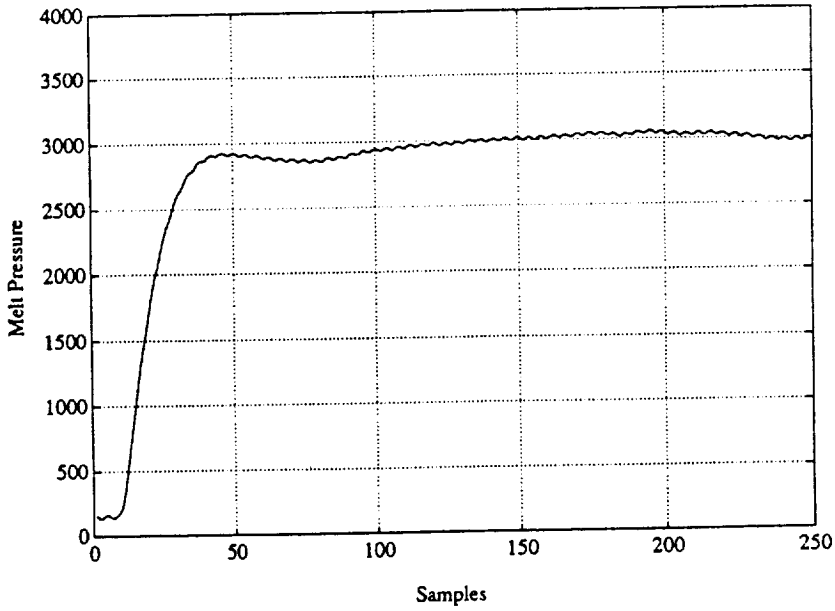

Figure 5.20: Process response at tradeoff surface point B.

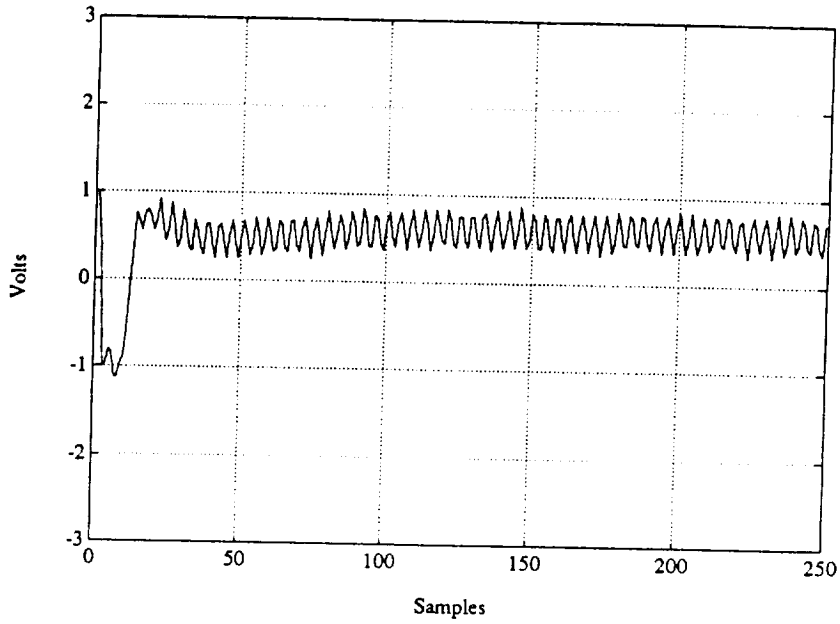

Figure 5.21: Control effort at tradeoff surface point C.

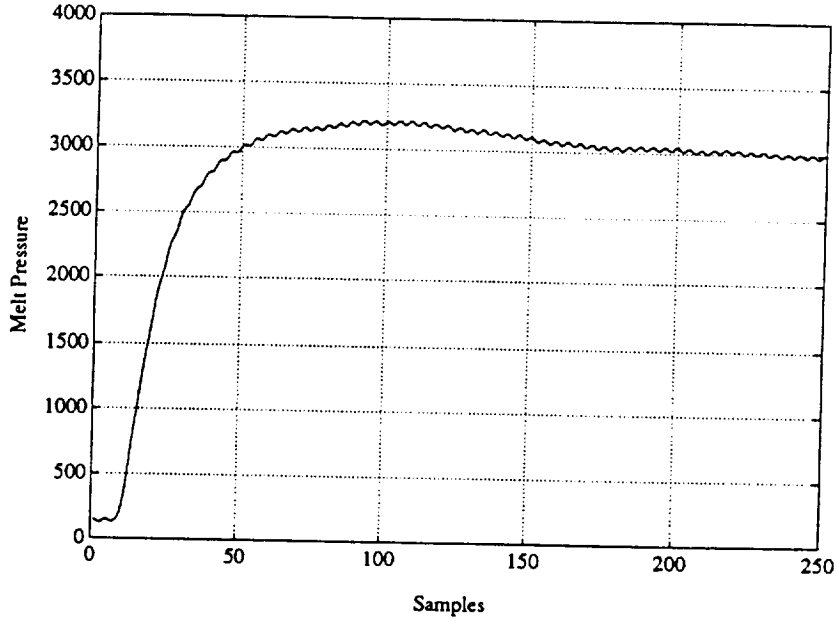

Figure 5.22: Process response at tradeoff surface point C.

5.4 Injection Controller Tuning

During injection, the accumulated molten polymer is injected into the mold. It is during this phase that the part is actually formed. Physical properties of the part, such as flashing, underfill, and initial polymer chain alignment are dependent on how the injection takes place. Injection develops the initial molecular structure of the part. Any final physical properties of the part depend on this initial molecular structure and the thermal history during holding and cooling.

The multiobjective optimization algorithm was used to tune injection PID controller gains, as was done for the plastication phase. For this tuning, three objective functions were chosen:

- Minimize Integral Time Absolute Error (ITAE)

$$\text{ITAE} = \sum_{k=1}^{T_f} k|R(k) - P_m(k)| \tag{5.9}$$

- Minimize Maximum Control Deviation

$$\|V\|_{16} = \left[\sum_{k=1}^{T_f} V(k)^{16}\right]^{1/16} \tag{5.10}$$

- Root Mean Square Error (RMSE)

$$\text{RMSE} = \left(\sum_{k=1}^{T_f} (R(k) - P_m(k))^2\right)^{\frac{1}{2}} \tag{5.11}$$

The first and second of these were the same as was used for the plastication controller tuning. The third objective was added to ensure faster rise time, which is deemphasized by the ITAE because of the time-based weighting.

5.4.1 Analytical Modeling of Injection

The injection dynamics have the same basic subsystems as plastication: the polymer fluid dynamics subsystem; the screw force balance subsystem; and the

hydraulic subsystem. The injection phase dynamics are dependent upon the mold design, as is discussed in more detail in Appendix A. From the formulas derived there, the following state space model can be formulated:

$$\dot{Z}_i = \begin{bmatrix} -\frac{RA_n\beta_m}{V_m} & -\frac{A_m\beta_m}{V_m} & 0 \\ \frac{A_m}{m} & -\frac{b}{m} & -\frac{A_H}{m} \\ 0 & \frac{A_H\beta_H}{V_H} & \frac{K_{P_i}\beta_H}{V_H} \end{bmatrix} Z_i + \begin{bmatrix} 0 \\ 0 \\ \frac{K_{X_i}\beta_H}{V_H} \end{bmatrix} U_i \qquad (5.12)$$

In the above equation, the injection state variable, Z_i, is defined using the same physical variables as Z_p. Again, if the desired output is melt pressure, then

$$Y_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} Z_i \qquad (5.13)$$

A block diagram of this system is shown in Figure [5.23].

Notice that this system only has one input, namely servovalve spool position. The servovalve transfer function is identical for this phase, as for the plastication, namely Equation (5.4).

Because the dynamics are dependent upon the mold, different controller gains must be used for every mold. In the quality experiments, the results of which will be presented in the following chapter, two different molds were used; a spiral mold and a four cavity test specimen mold. Because of this, two different sets of injection controller gains had to be found. For the spiral mold, it was found that a slight modification of the plastication PID gains resulted in gains that provided adequate control for injection. For the four cavity mold, these gains yielded a completely unsatisfactory response.

5.4.2 Description of the Injection Controller Tuning Experiments

All injection controller tuning experiments were run using the ASTM four cavity mold. This was the second mold used for the quality control experiments, which are described in Chapter 6. Process input settings for these experiments are given

Table 5.3: Symbols used in the injection dynamic equations.

A_H	hydraulic pressure effective area
A_m	melt pressure effective area
b	approximate viscous friction on screw
K_m	melt pressure empirical gain
K_{Xi}	servovalve spool position flow gain
K_{Pi}	servovalve hydraulic pressure flow gain
K_v	servovalve gain (voltage to spool position)
K_ω	screw speed empirical gain
m	approximate mass of screw and polymer system
P_H	hydraulic pressure
P_m	polymer melt pressure
R	effective mold flow resistance
U_i	plastication input
V_H	hydraulic oil volume
V_m	melt volume
V_s	recovery rate (screw linear velocity)
V	servovalve input voltage
X_v	servovalve spool position
Y_i	plastication output
Z_i	plastication state vector
β_m	melt bulk modulus
β_H	hydraulic oil bulk modulus
τ_v	servovalve time constant

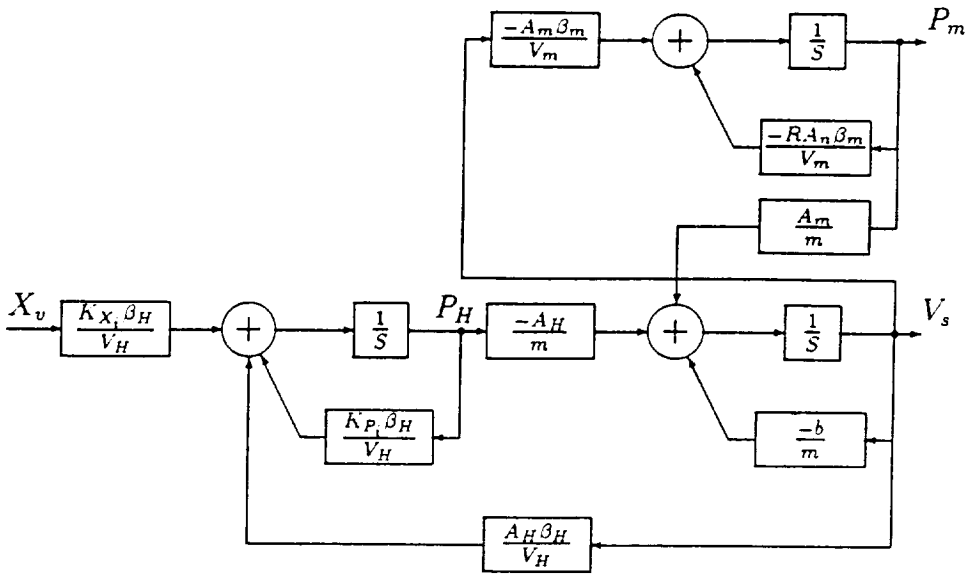

Figure 5.23: Injection dynamics block diagram.

in Table 5.4. These values were held constant during all of the experiments. Injection controller gains and holding controller gains were varied by the multiobjective algorithm. During these experiments, holding gains were set equal to injection gains because holding dynamics will be the same as those at the end of injection. Keeping the gains the same reduced the deterministic disturbances applied to the machine to those only caused by changing the setpoint. It was desirable to minimize these disturbances because anything which caused a pressure spike during injection could lead to flashing of the part. The goal was to make the transfer from injection to holding as "bumpless" as possible.

5.4.3 Experimental Results

After approximately 100 runs, the nondominated sets shown in Figures [5.24], [5.25], and [5.26] were found. The associated efficient points are shown in Figures [5.27], [5.28], and [5.29]. In these figures, the points are lettered **A** through **F**

Table 5.4: Nominal Input Settings

Polymer	Polypropylene
Screw Speed	380 RPM
Barrel Temperature	220 C, 220 C, 220 C, 205 C
Shotsize	3.25 cm
Pullback	0.5 cm
Cushion	0.75 cm
Holding Time	5.0 s
Cooling Time	10.0 s
Plastication Setpoint	3000 PSI
Injection Setpoint	2500 PSI
Holding Setpoint	2000 PSI
Plastication K_P	$-4.13 \cdot 10^{-4}$
Plastication K_I	$-1.76 \cdot 10^{-5}$
Plastication K_D	29.3

for ease of comparison. This lettering does not indicate any of the decision maker's preferences or any time relationship between the different runs. It should only be used to associate nondominated points with the corresponding efficient points.

Since there were three objective functions, it was not possible to plot the tradeoff surface as a smooth curve, as was obtained for the plastication controller tuning. In this case, presentation of the nondominated sets consists of pair–wise plots. Each plot represents the *projection* of the nondominated set into the subspace defined by the axes of the plot. The general shape of the tradeoff surfaces in Figures [5.24], [5.25], and [5.26] lend some insights into the behavior of the system.

During algorithm iterations, the ITAE and RMSE objectives were not in conflict, as can be seen by the shape of the tradeoff surface in Figure [5.24]. If it is possible to get an approximately deadbeat response with the given controller, these two objectives would produce the same tuning results. Any difference would lie in tradeoffs that had to be made of rise time over steady–state error. One big difference in these two objectives was their respective dynamic range. Over the same

responses, ITAE doubled whereas RMSE changed by about 13%. This difference can be attributed to the time-dependent weighting. Over all of the gains found, the transient response was about the same, the system immediately shot up to about 3000 PSI. After this, for half of the trajectories, lots of control energy was used to try and force the system back to the setpoint quickly. For the other half, the system was allowed to slowly settle to the setpoint, but because injection was so short, it was never reached.

By examining the efficient points in Figures [5.27], [5.28], and [5.29], it can be seen that controller saturation is directly tied to the magnitude of K_P. This seems reasonable since the overall controller gain and the position of one of the controller zeros is dependent on K_P (see Equation (5.1). As K_P increases, the bandwidth of the open-loop system is increased, forcing the system to react much more quickly.

Gains defined by point **G** in these figures were used as controller gains for the quality experiments. The only points at which the controller did not saturate were **E**, **F**, and **G**. Point **G** was chosen because these gains gave the least saturation for no meaningful gain in setpoint tracking. Furthermore, if one compares Figures [5.30] with [5.32], the "**G**"-gains yield a response which is much flatter. Since inputs will be tuned in the quality experiments, even though, the setpoint is not tracked, this can be compensated for by adjusting the "applied" setpoint to be lower than the "desired" setpoint. This will be taken care of automatically by the algorithm during the quality tuning since the operator will be controlling *product quality* directly. This is an important feature of this approach to quality control. Specific machine idiosyncrasies are invisible to the operator.

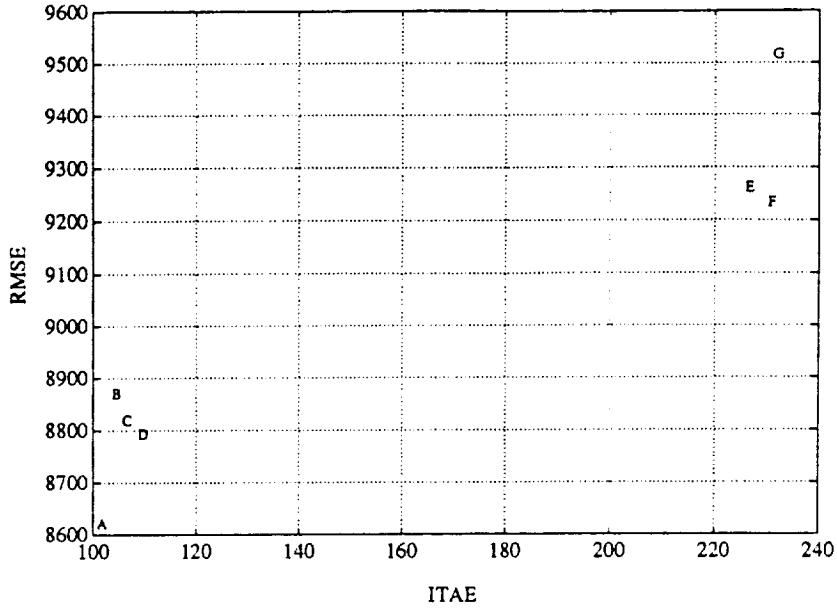

Figure 5.24: Nondominated set of ITAE versus RMSE.

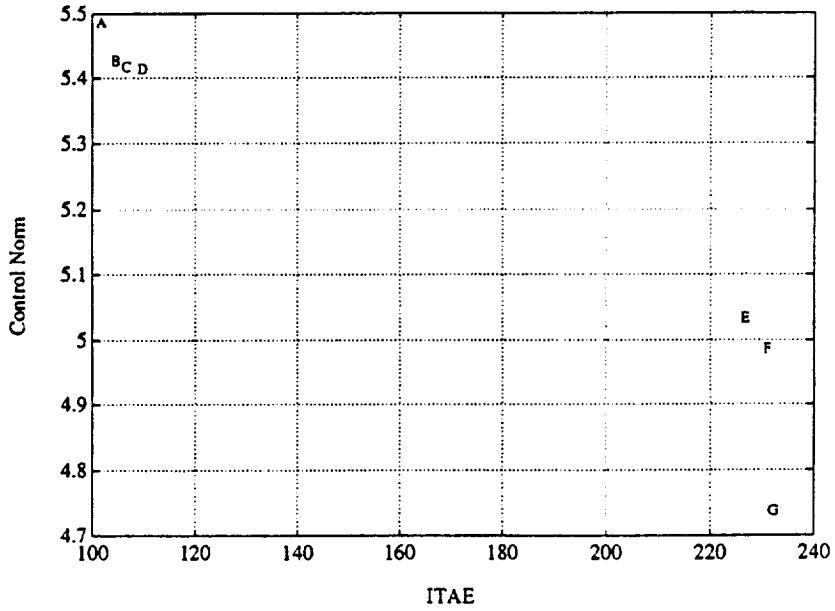

Figure 5.25: Nondominated set of ITAE versus maximum control deviation.

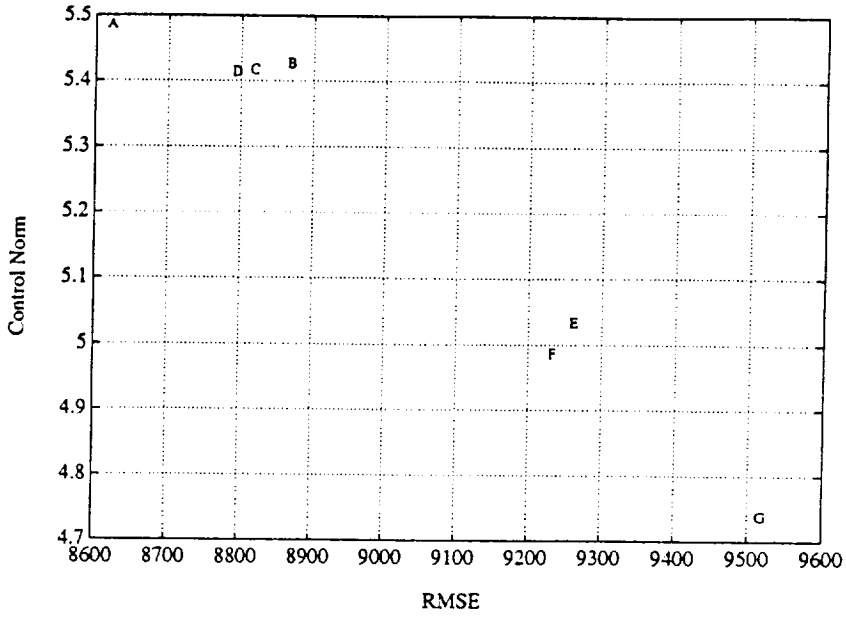

Figure 5.26: Nondominated set of maximum control deviation versus RMSE.

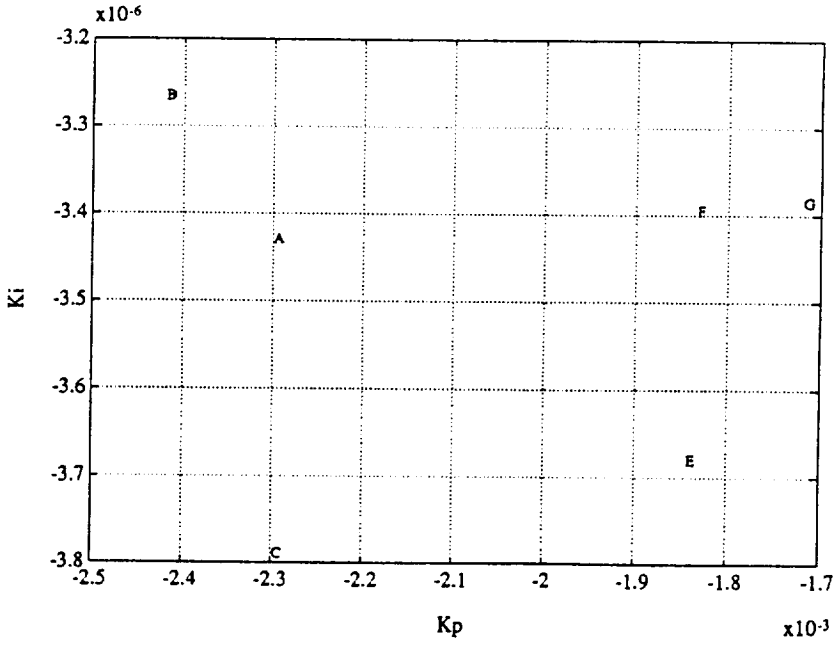

Figure 5.27: Efficient points: K_P versus K_I

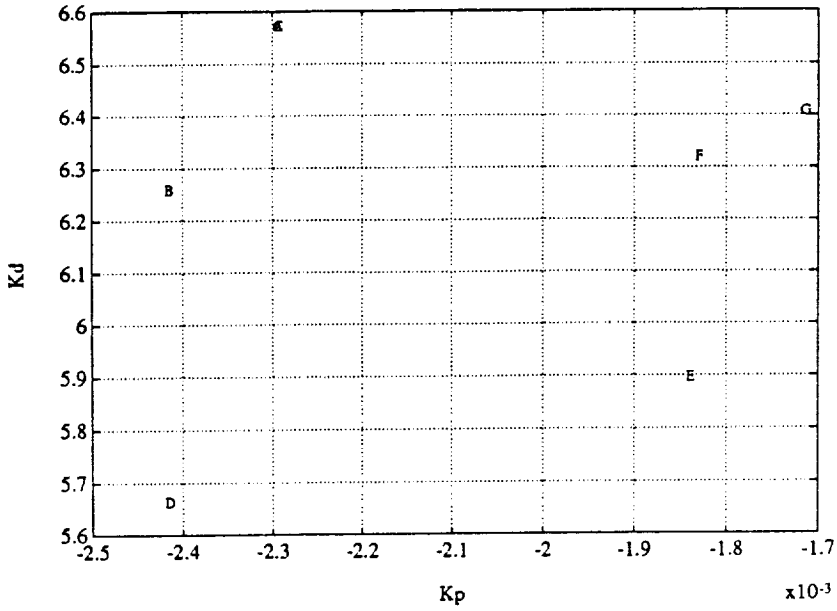

Figure 5.28: Efficient points: K_P versus K_D

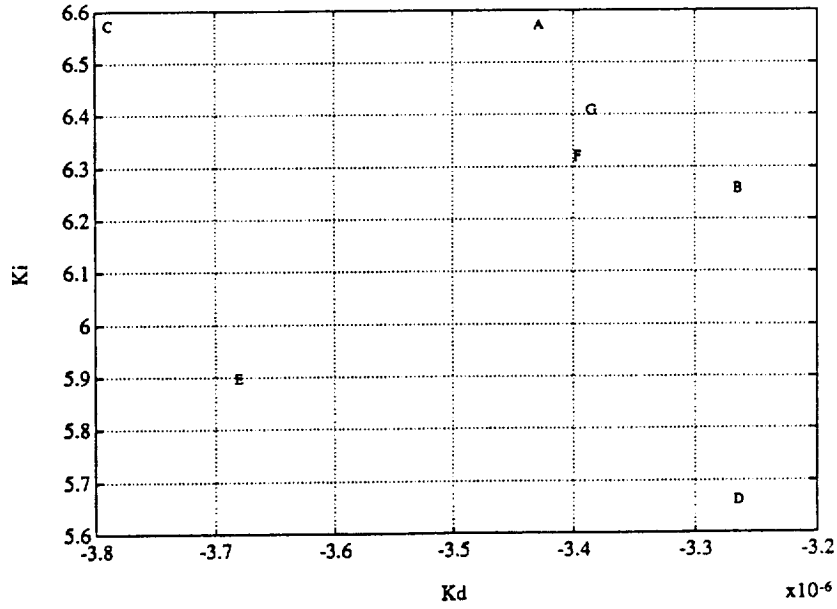

Figure 5.29: Efficient points: K_I versus K_D

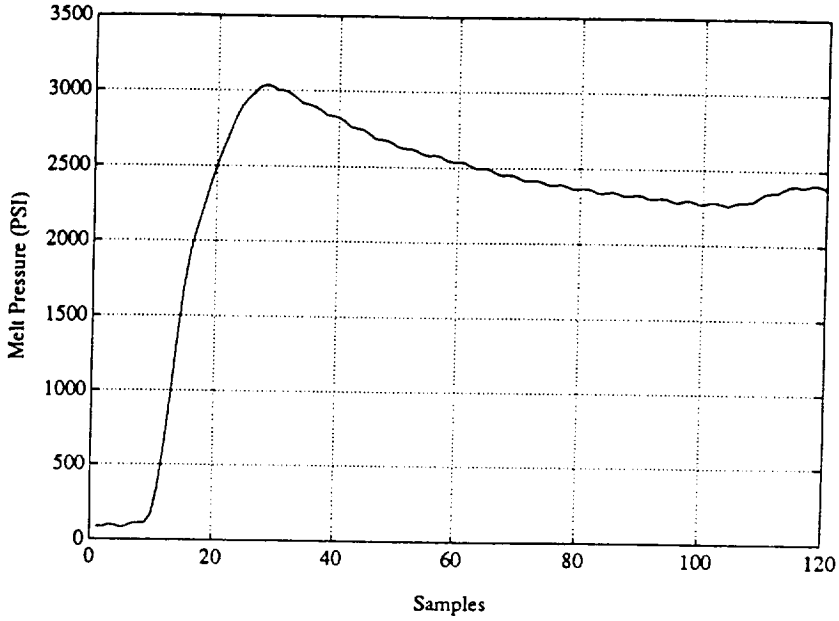

Figure 5.30: Melt pressure output at nondominated point A. The set-point is 2500 PSI.

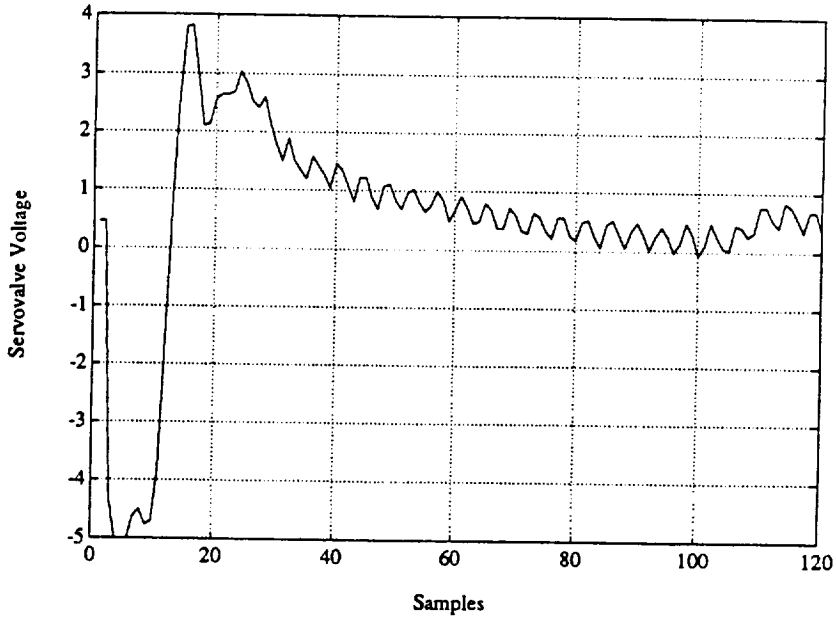

Figure 5.31: Servovalve input at nondominated point A.

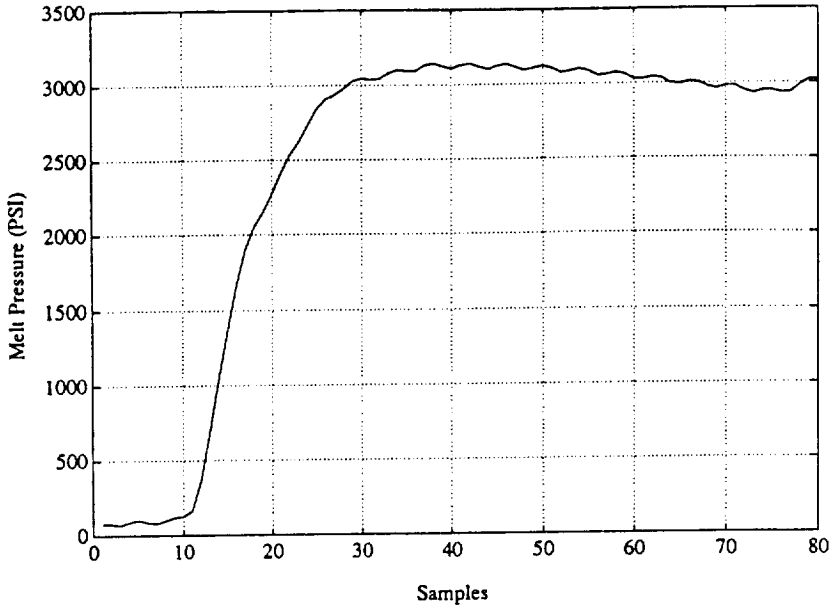

Figure 5.32: Melt pressure output at nondominated point G. The set-point is 2500 PSI.

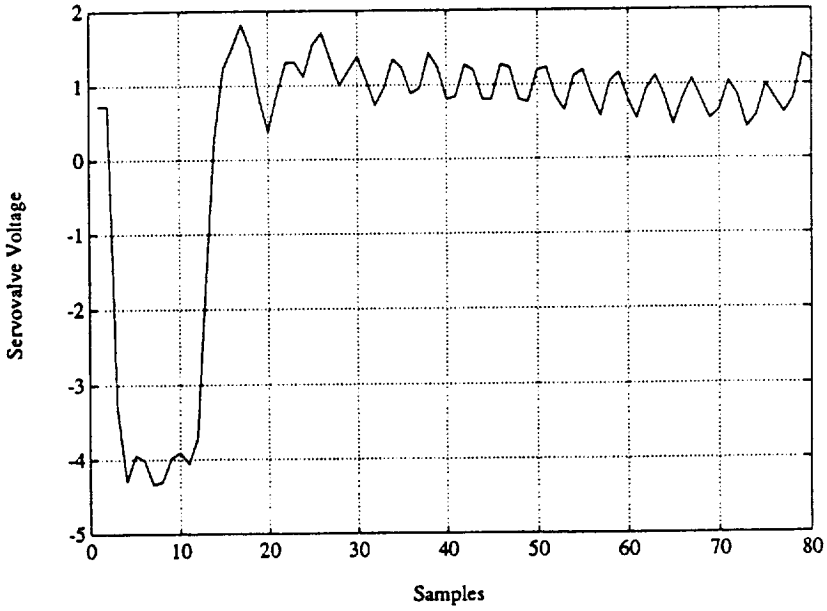

Figure 5.33: Servovalve input at nondominated point G.

5.4.4 Discussion

In general, it was found that this system was not nearly as well behaved as the plastication system. Some discussion of the behavior of the system will lend some insight into the problem.

The actuator for this system is a three port servovalve. This valve has two stages, a spool valve which regulates the hydraulic flow through the valve, and a pilot stage which drives the spool. The pilot stage can be thought of as the hydraulic equivalent to a "pre-amplifier". This pilot stage is actuated by an armature which is voltage driven. Ideally, if the valve is balanced, an input of zero volts results in no motion of the spool. This voltage is known as the *null voltage*. The valve used on the Arburg was not balanced, and due to the valve's integrating action, the spool would tend to drift to the completely closed or completely open position. Attempts were made to identify the null voltage, but this proved to be fruitless. One assumption was that the null voltage was the control output during plastication, when steady-state error was zero, but this did not prove to be correct. Apparently the null voltage is dependent upon the spool position, which is not unreasonable [4].

A second problem with the valve was stiction of the spool. In previous experiments, the valve was repeatedly given various open-loop inputs, to try and characterize valve response, but no clear relationships could be determined.

A third problem was slew limit and position limits of the spool. No matter how large an input was given to the valve, it could open only so fast and so far. This also limited the ability of the valve to react to large command signals. These effects were lessened by applying an empirically determined *preset signal* to the valve. This was an attempt to position the spool near it's steady-state operating position, hopefully making the valve behave "more linearly". This approach was successful for the plastication phase. These problems were probably not as significant during

[4] A complete discussion of servovalve dynamics can be found in [62].

plastication because the demands made on the servovalve were not as great.

The injection dynamics also contained a backlash nonlinearity with respect to the melt pressure sensor. After plastication, the polymer is "decompressed" by pulling the screw back slightly. This results in the polymer being pulled out of contact with the melt pressure transducer. During the initial stages of injection (typically the first eight samples), there is no pressure signal, causing the controller to maintain a "full on" signal. When the melt pressure does start registering, a very fast rise time is detected which the controller must counter-act. Because of the slow valve dynamics, due to the integrator, this reaction does not take place quickly enough, resulting in a large overshoot by the system. Before the controller can recover, injection is over. (In Figures [5.30] and [5.32], there is a pressure rise near the end of the cycle. This occurs when the mold is full. The injection dynamics change suddenly because there is no longer a polymer flow. Ideally, switchover from injection to holding would occur precisely at this point.)

It was also found that the injection screw itself was rate limited. This problem did not occur when using the spiral mold because the typical polymer flow path cross section was much smaller. This caused a higher "flow resistance", which translated into a higher melt pressure for a given velocity. In the four cavity mold, the flow resistance was roughly one fourth that of the spiral mold because of the many runners. At low injection *pressures*, the maximum screw velocity of 7 cm/sec was easily achieved. Essentially, the system could not track setpoints above 3000 PSI. The typical response, in this case, was a trajectory that ramped up to the setpoint but might not have reached it by the time the mold was filled.

To put these problems in perspective, consider that an injection molding operator typically does not have the process data available to detect these control problems. If process performance indicates that he should be increasing injection

pressure, he will do this without knowing that there is "poor tracking" of the command. The only indication he may have is that above a certain point, the process output is insensitive to any changes he makes. This information is built into the operator's intuitive process model, which is derived from experience.

Despite these problems, a *reasonable* set of controller gains were found. These were used for injection control during the quality experiments, that will be discussed in the next chapter.

5.5 Conclusions

In this chapter, results on using the multiobjective optimization algorithm for controller tuning were presented. The algorithm was used to tune a PID controller in three different situations; during a simulation of plastication, on-line for plastication, and on-line for injection.

This particular implementation of controller tuning has a very interesting analogy to root-locus design. Once a feasible direction is determined by the algorithm, the gains are parameterized along this direction and changed linearly; this is the algorithm line search phase. Performance of the controller is evaluated at each step of this line search with the hope that it is being continually improved. The Evans root locus provides location of the closed-loop poles of the system as this gain is varied linearly. The construction of an Evans root locus requires a very specific parameterization of the controller, which does not occur with the line search in the algorithm, but both show a continuous relationship between the closed-loop response and the free parameter.

In general, the construction of the tradeoff surface, by the algorithm, for each of the systems yielded information that was valuable in understanding the approach to control. In the case of the plastication phase, the nondominated set revealed that the PID controller provided quite good response. In fact, the efficient gains

from tradeoff surface point A in Figure [5.11] provide practically deadbeat control to the input (this response is shown in Figure [5.18]). Practically speaking, any of the nondominated gains for this system provide very good control.

Conversely, achievable control during injection is worse. Several conclusions can be drawn from this. One is that a different controller might possibly provide adequate control. It is also possible that the actuator is insufficient for the system. It should be noted that, even though control was lousy, if "setpoint–tracking" was the major performance objective, a set of controller gains was found that produced a machine response that had desirable response characteristics, namely being fairly flat after the initial transient. Since the multiobjective optimization approach uses both command inputs and control gains simultaneously, this could be taken advantage of.

In all of the cases discussed, the multiobjective algorithm provided a very methodical way of obtaining information about the relationship between system response and controller gains.

6. Tuning the Process to Achieve Quality Objectives

The multiobjective optimization algorithm was applied to the tuning of process inputs to achieve quality objectives. In this chapter, two different cases will be discussed; a spiral mold and an ASTM four cavity test specimen mold. Each of these molds present different control problems and have different quality objectives associated with them.

6.1 Spiral Mold Quality Tuning

One of two molds used in a quality control case study was a standard spiral mold. The mold consists of a constant cross–section channel arranged in an Archimedean spiral. The spiral is marked in one inch increments and has a total length of 65 inches. The actual length of a part during any given injection cycle is a function of the polymer viscosity, mold cavity temperature, the melt temperature, and the injection pressure and velocity [66]. Figure [6.1] shows an end-on view of the spiral part. From this vantage, the sprue is projecting out of the page.

6.1.1 Description of the Spiral Mold Quality Tuning Experiments

For this part, optimization was to take place with respect to the following quality objectives:

- prevent flashing (quality attribute);
- minimize cycle time (quality variable);
- minimize the variance of the spiral about a nominal length of 35 inches (quality variable).

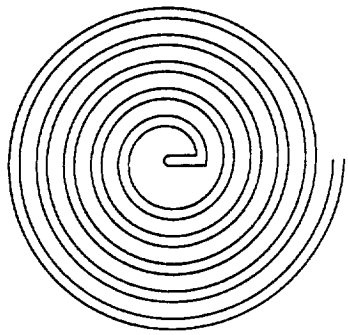

Figure 6.1: The spiral part.

Flashing and spiral length were measured by operator inspection. Each time a part was ejected, the presence or lack of flashing was noted and the length of the part was measured. This information was then input to the algorithm and the cycle was continued. Cycle time was measured as the elapsed time of plastication and injection and was recorded automatically. All other phases of the cycle were of fixed duration.

In running these experiments, it was decided that only plastication and injection setpoints would be varied. The philosophy behind this choice was that these are inputs that are typically controlled by an operator. It would be possible to vary controller gains as well, but it was felt that this would make the experimentation unnecessarily complex with no added benefit. (One alternative would be to run the algorithm sequentially, first finding the desirable operating points, then tuning the controllers to optimize performance around these points. This would, to some degree, require that the system response to setpoint changes be "orthogonal" to the response to gain changes.)

The algorithm was initialized with a plastication setpoint of 4000 PSI and an injection setpoint of 10000 PSI. All other process inputs were held constant,

at levels shown in Table 6.2. The algorithm was then used to determine those setpoints that would minimize all performance criteria simultaneously. Since there were only two inputs, a two level factorial design with center point was used for experimental perturbations to determine gradients. For experiments run using this design, each parameter can take on two distinct values, its nominal value *plus* the desired perturbation, and its nominal value *minus* the perturbation. An additional experiment was run with all parameters unperturbed: the centerpoint. This yielded a total of *five* input level combinations. Each input level combination was replicated four times so that deleterious effects due to process noise would be minimized. Therefore, twenty runs were required when a gradient had to be determined. To confound other unmeasured interactions, input levels for sequential experiments were chosen randomly [45]. The initial perturbation value was to vary each setpoint by 10%. Table 6.1 shows a possible randomized sequence of the 20 experiments. During experimentation, a new random order was chosen every time.

Randomization of experiments is important. In this case, the gradient of the performance objectives was being estimated. Essentially, this is the estimation of a linear, static model. It is important to make sure that *only* effects due to the desired inputs are measured in the output. A prime example of this type of problem is shown in Figures [2.33] and [2.34]. The controller response was highly correlated to hydraulic oil temperature which wasn't measured or controlled by the system. Randomization of input changes help prevent this type of deterministic disturbance from affecting the gradient estimation significantly.

Once a gradient was found, the line search was begun. During the line search, each run was replicated four times. Since it was not possible to randomize these experiments, data from these experiments were susceptible to the interaction that was avoided when estimating gradients. The line search was continued until a dominated point, in objective space, was generated. This indicated that either the

Table 6.1: A Sequence of Randomized Runs.

Run	Plastication Setpoint	Injection Setpoint
1	3600	11000
2	4400	9000
3	3600	9000
4	4400	11000
5	4400	11000
6	4000	10000
7	4400	9000
8	4000	10000
9	4000	10000
10	3600	9000
11	4000	10000
12	3600	11000
13	3600	9000
14	3600	11000
15	4400	9000
16	3600	9000
17	3600	11000
18	4400	11000
19	4400	9000
20	4400	11000

Table 6.2: Nominal Input Settings for the Spiral Mold

Polymer	Polypropylene
Screw Speed	380 RPM
Barrel Temperature	220 C, 220 C, 220 C, 205 C
Shotsize	2.5 cm
Pullback	0.25 cm
Cushion	1.0 cm
Holding Time	5.0 s
Cooling Time	10.0 s
Holding Setpoint	3000 PSI
Plastication K_P	$-4.13 \cdot 10^{-4}$
Plastication K_I	$-1.76 \cdot 10^{-5}$
Plastication K_D	29.3
Injection K_P	$-4.13 \cdot 10^{-4}$
Injection K_I	$-1.76 \cdot 10^{-5}$
Injection K_D	40.0
Holding K_P	$-4.13 \cdot 10^{-4}$
Holding K_I	$-1.76 \cdot 10^{-5}$
Holding K_D	40.0

tradeoff surface had been reached or the current direction no longer pointed to this surface.

6.1.2 Experimental Results

Process inputs generated by the algorithm are shown in Figures [6.2] and [6.3]. Those runs which were used for gradient estimation, and those used for the line search are easily distinguished. The resulting cycle time and spiral length are shown in Figures [6.4] and [6.5]. The initial inputs yield both a long cycle time and a spiral that is too short. The gradient identified by the algorithm directed the system to decrease the plastication setpoint and increase the injection setpoint. Both of these resulted in a faster cycle. At the same time, as the injection setpoint was increased a longer spiral was produced. At input levels for Runs 40 to 44, the spiral was too long. The algorithm was stopped because this operating point was dominated by one that had already been determined. The operator was then presented with the empirically determined nondominated set and efficient points depicted in Figures [6.6] and [6.7].

The operator selected a new region on the tradeoff surface to explore, by choosing a point from the experimentally generated non-dominated set. In this case, it was point **A** shown in Figure [6.6]. Automatically, the corresponding efficient operating points noted in Figure [6.7] were transmitted to the 68000 microprocessor. The objective function gradients were again determined through a series of planned perturbations from the new input settings. This series of experiments is depicted in Runs 45 through 60 in Figures [6.2] and [6.3]. In this instance, only one step along the feasible direction was made before the tradeoff surface was encountered. The resulting nondominated set and efficient points are shown in Figures [6.8] and [6.9].

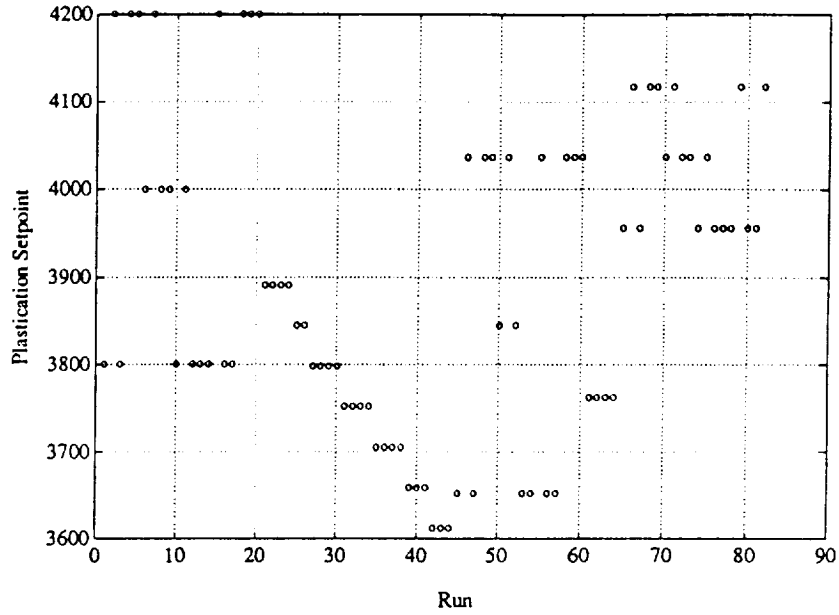

Figure 6.2: Plastication inputs for spiral mold optimization.

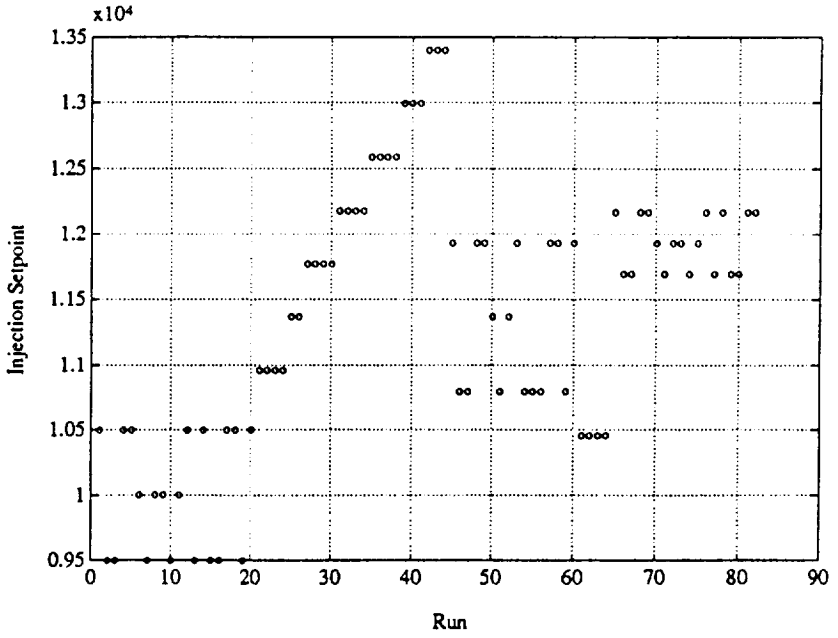

Figure 6.3: Injection inputs for spiral mold optimization.

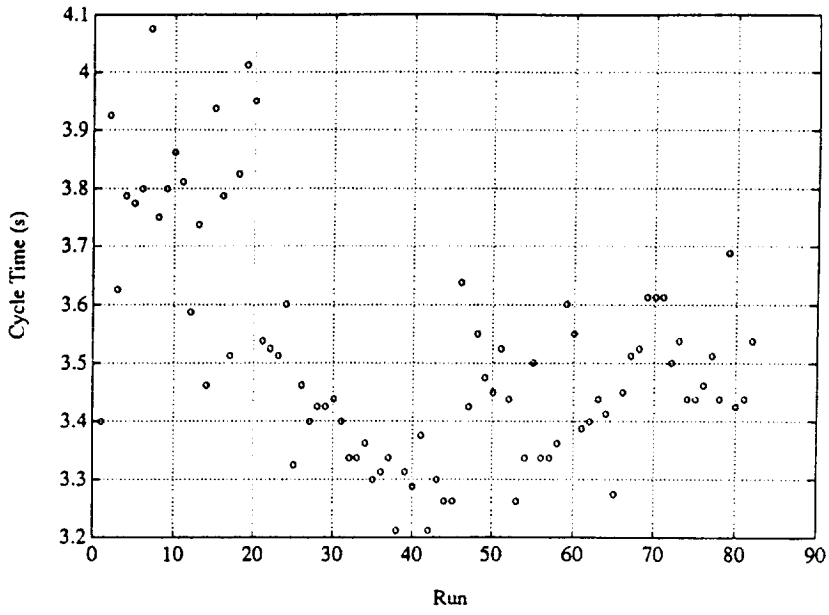

Figure 6.4: Cycle time as the algorithm progressed.

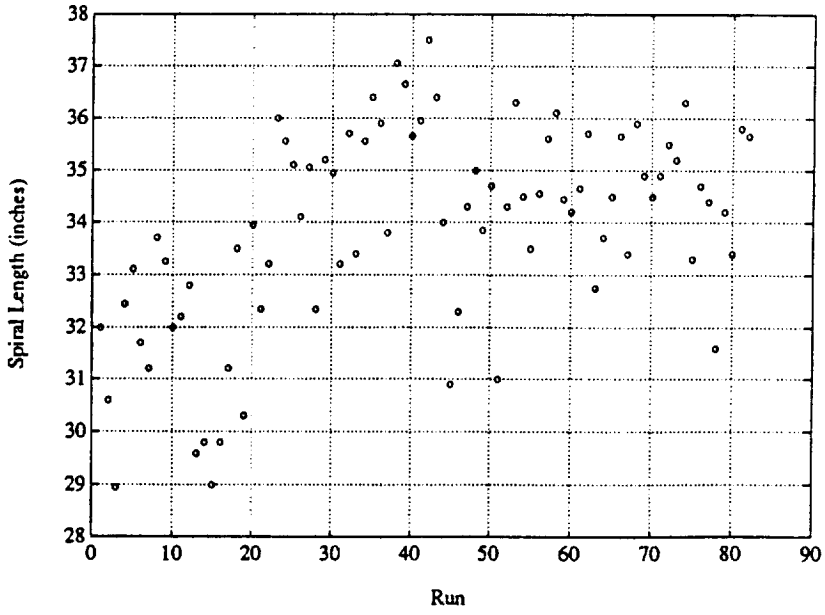

Figure 6.5: Spiral length as the algorithm progressed.

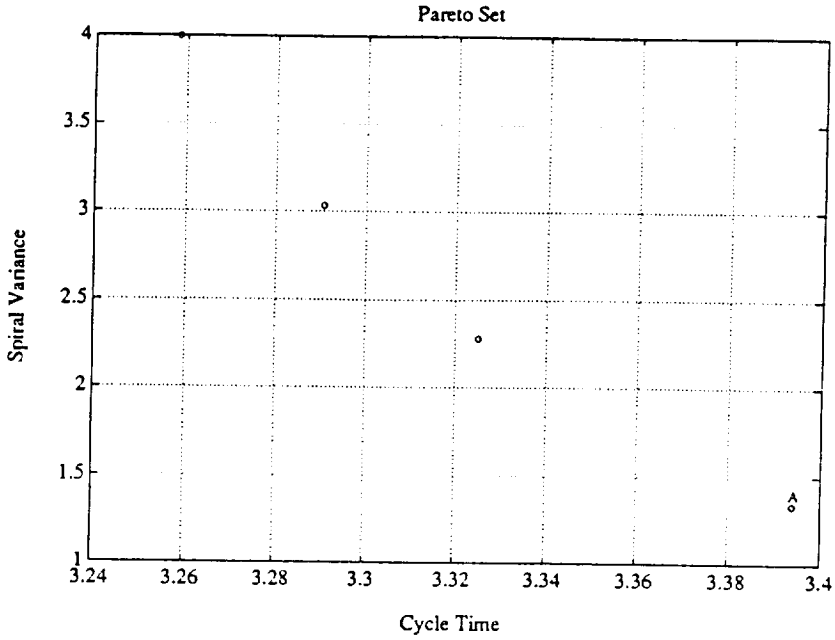

Figure 6.6: Nondominated set after first line search.

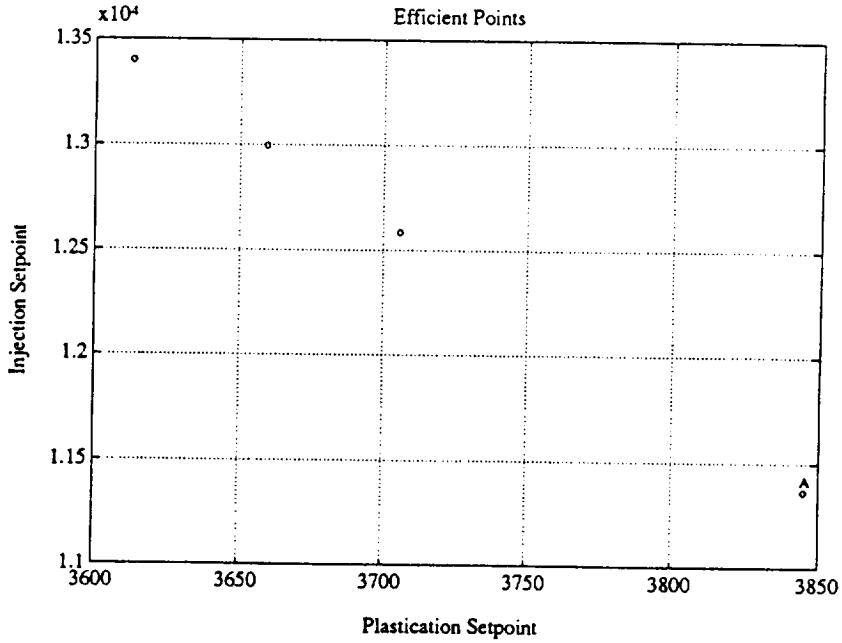

Figure 6.7: Efficient points after first line search

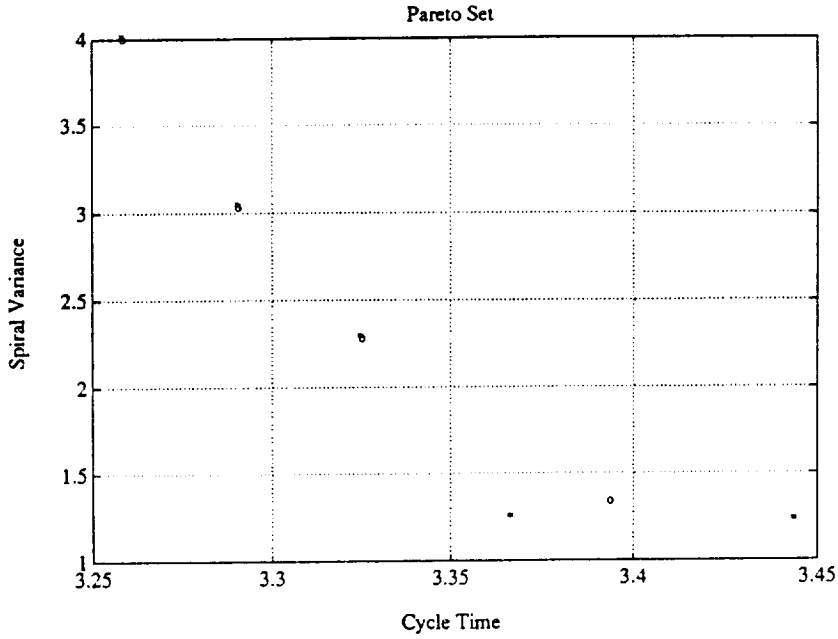

Figure 6.8: Comparison of first (o) and second (*) nondominated sets.

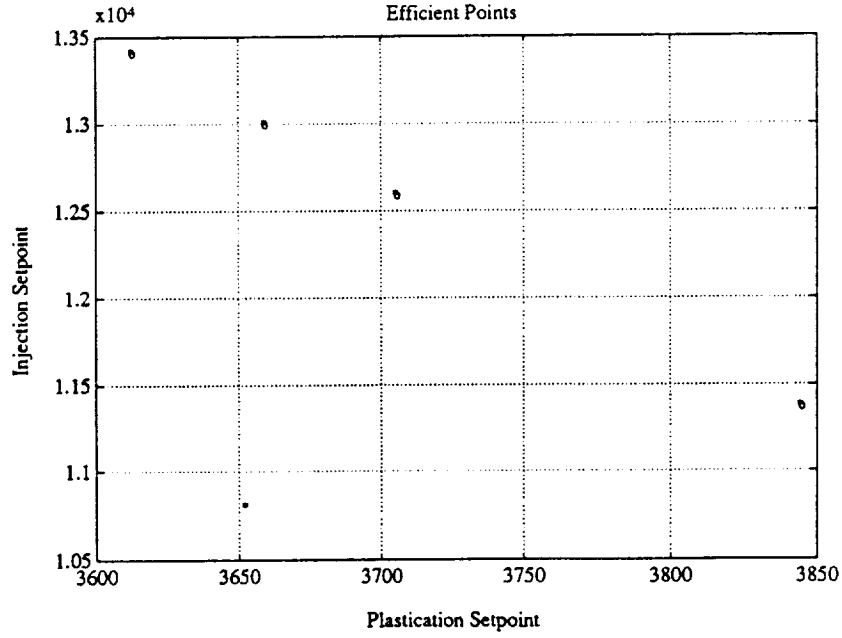

Figure 6.9: Efficient points for first (o) and second (*) nondominated sets.

6.1.3 Discussion

The overall behavior of the algorithm seems to improve the performance of the plastic injection molding process. Figures [6.4] and [6.5] show that there is a steady decrease in cycle time and the target spiral length of 35 inches is being approached. Also, the nondominated sets, shown in Figures [6.8] are beginning to take on the characteristic "hyperbolic" shape.

Close inspection of the figures shows that some levels do not appear as if they were replicated. In these instances, such as near Run 24, there was a control computer failure that resulted in unusable data being collected. After Run 87, the control computer failed completely.

Achieving the quality objectives of this mold was not pursued after these runs because it was desirable to see how the algorithm operated on a quality attribute, which this part did not contain. The problem is that, except for some uncontrolled runs, it was not possible to cause flashing of this part. This is essentially due to the mold design. In general, flashing occurs when the injection pressure exceeds some threshold, after the mold is filled. When this happens, the mold is forced open slightly, allowing the polymer to flow into this thin opening. In the case of the spiral mold, it is "open–ended"; with a total length of 65 inches. It is impossible to completely fill the mold. Any "over pressuring" that occurs merely results in a lengthening of the spiral, rather than flash.

During experimentation, a second quality problem was observed; the occurrence of shrinkage voids. This also could not be removed, because of the mold design. These voids occur when the plastic shrinks as it cools, allowing them to form inside the body of the part. Formation of voids is avoided by maintaining pressure on the polymer melt, during the holding phase, allowing more polymer to flow into the mold as shrinking occurs. Since this mold is open-ended, maintaining this pressure is nothing more than continuing injection except at a lower pressure.

6.2 ASTM Four Cavity Mold Quality Tuning

Quality experiments were run on the ASTM four cavity mold. The part from this mold is pictured in Figure [6.10]. Quality objectives for this part were:

- Minimize cycle time (a quality variable).
- Prevent flashing (a quality attribute).
- Prevent underfill (a quality attribute).

The first two objectives were also used for the spiral mold. The last objective, underfill, represents the condition where an insufficient amount of polymer has been injected into the mold such that the part does not fill completely. It is possible for both flashing and underfill to occur simultaneously, possibly when the molten polymer is too cold such that the mold freezes off before injection is complete and the remaining polymer flashes. Information about the quality attribute was provided via operator inspection. After each part was ejected, the operator was asked to decide whether the part had flashing or underfill. Two different approaches to inspection were tried. One approach was that a threshold was defined and the operator was allowed to answer either yes or no. For the second approach a *fuzzy* definition of these attributes was used, where the attribute was from zero to one, and this information was passed to the algorithm. Although this strictly violates the definition of a quality attribute, as described in Section 3.2.2.2, the use of a fuzzy measure improved the performance of the algorithm. The use of fuzzy measures for these constraints is analagous to the use of penalty or barrier functions in optimization problems with analytic constraints [55]. (It should be noted that, because of replications, even non-fuzzy measures of flashing and underfill can result in a quality attribute value that is between zero and one.)

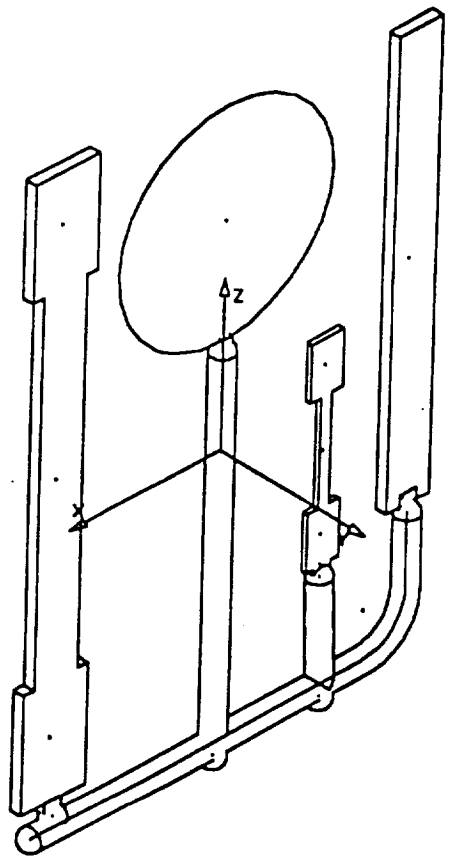

Figure 6.10: The ASTM test part.

6.2.1 Experimental Conditions

An additional quality variability was evaluated for feasibility, that of minimizing part weight variance. This would be another measure of uniformity of parts, and could possibly be related to consistency of part density. It turned out that for those parts which showed no flashing or underfill, the weight variability was on the order of a hundredth of a gram. This weight variation was considered too small to be meaningful and so this objective was dropped.

For these experiments, controller gains were held constant, using the values determined from the controller tuning experiments. As was done for the spiral mold, only the plastication and injection setpoints were changed. The initial setpoints were 4000 PSI for plastication and 2500 PSI for injection. Also, each experimental level was replicated four times, and randomization was used in determining the gradients, for the same reasons as discussed in Section 6.1. All of the machine inputs are given in Table 6.3.

6.2.2 Discussion of the Tuning Results

In Figures [6.11], [6.12], and [6.13], the non-dominated set found from the initial point is shown. Interestingly enough, underfill and flash are mutually exclusive for this part. Apparently there is no problem with the temperature of the polymer for these runs. The nondominated sets of Figures [6.11] and [6.13] have the characteristic hyperbolic shape one expects of a tradeoff surface. This is because of the "orthogonality" of flashing and underfill. It is especially easy to identify which parts violate the flashing objective in Figure [6.13] because they all appear on the zero-underfill axis, but with a faster cycle time than is possible for any of the parts with underfill. (This is another indication of the relationship between mold filling and injection pressure.)

At the end of thirty search iterations, the operator was presented with these

Table 6.3: Nominal Input Settings for the ASTM Mold

Polymer	Polypropylene
Screw Speed	380 RPM
Barrel Temperature	220 C, 220 C, 220 C, 205 C
Shotsize	3.25 cm
Pullback	0.5 cm
Cushion	0.75 cm
Holding Time	5.0 s
Cooling Time	10.0 s
Holding Setpoint	Inj. Setpoint - 2000 PSI
Plastication K_P	$-4.13 \cdot 10^{-4}$
Plastication K_I	$-1.76 \cdot 10^{-5}$
Plastication K_D	29.3
Injection K_P	$-4.13 \cdot 10^{-4}$
Injection K_I	$-1.76 \cdot 10^{-5}$
Injection K_D	40.0
Holding K_P	$-4.13 \cdot 10^{-4}$
Holding K_I	$-1.76 \cdot 10^{-5}$
Holding K_D	40.0

non-dominated sets. The point denoted by **A** in Figure [6.11] was used to initiate the algorithm for the second iteration. The corresponding efficient points are marked in Figure [6.14]. This nondominated point was chosen because it exhibited no underfill and the smallest amount of flashing. The approach was to try and satisfy the quality attribute constraints before minimizing the quality variable.

An additional thirty search iterations yielded the nondominated set and associated efficient points shown in Figures [6.15] and [6.16]. After this iteration, there was no underfill, whatsoever.

Evaluation of the data from these runs showed a curious phenomenon: as plastication setpoint decreased, injection stroke increased. This increased stroke was not correlated with an increased part weight or other physical characteristic. It can only be surmised that the check valve in the plasticating screw, which is supposed to prevent back flow of plastic, was not seating itself and therefore was allowing leakage. This may also be a function of the low plastication pressure and high injection pressure. In these experiments, a minimum setpoint constraint was set to be 1000 PSI. This was increased to 2000 PSI and an additional set of experiments were run.

An initial nondominated set was constructed from the data of the previous runs. The algorithm was initialized at a suitable point and was allowed to run through two iterations. The resulting nondominated sets and efficient points are shown in Figures [6.17] to [6.20]. The algorithm was allowed one more iteration, and the final nondominated set and efficient points are shown in Figures [6.21] to [6.22]. Again, operating conditions were found such that underfill was removed entirely.

Figures [6.23] and [6.24] show the input changes made during the last three iterations. From the initial point, the plastication setpoint is decreased and the

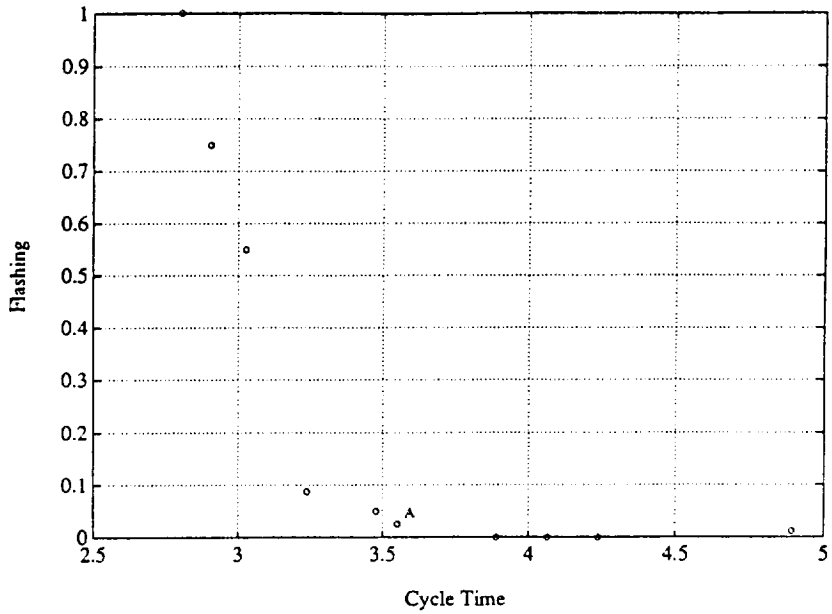

Figure 6.11: Nondominated set of cycle time versus flash, after the first algorithm iteration.

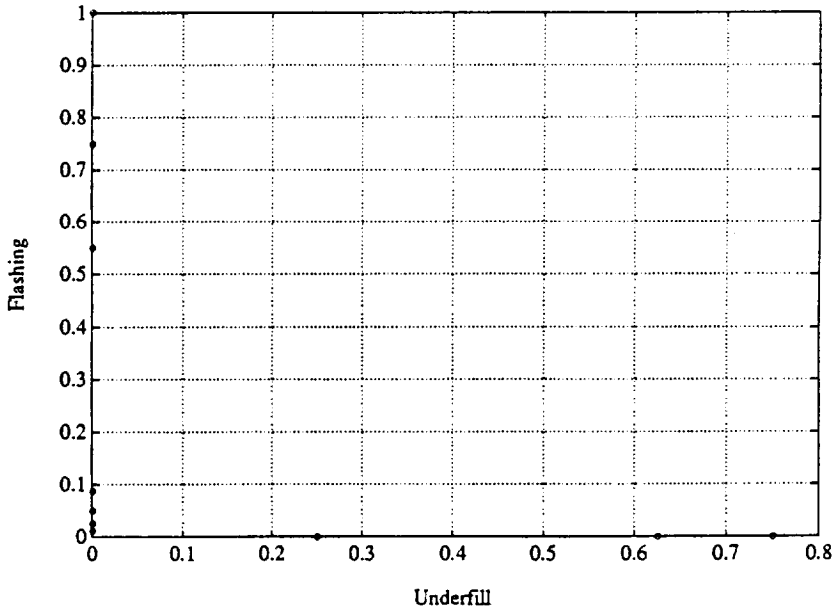

Figure 6.12: Nondominated set of underfill versus flash, after the first algorithm iteration.

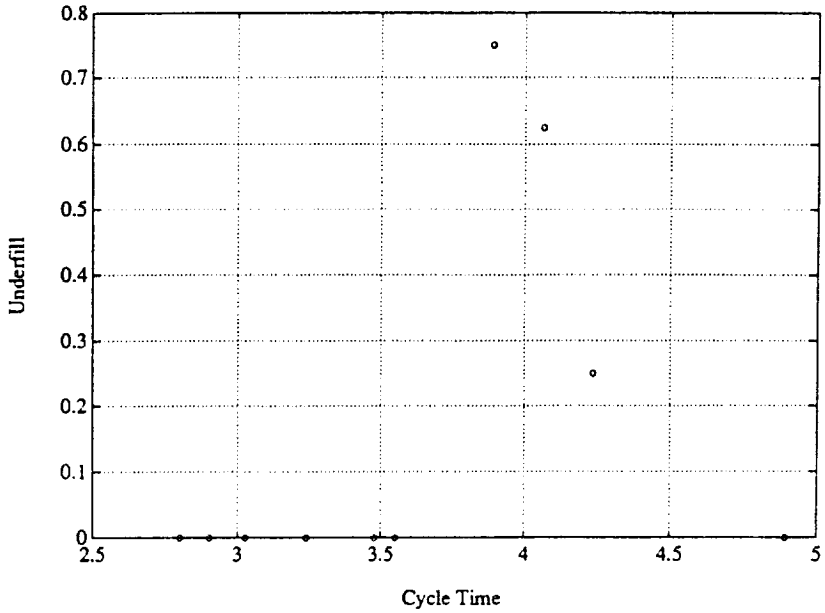

Figure 6.13: Nondominated set of cycle time versus underfill, after the first algorithm iteration.

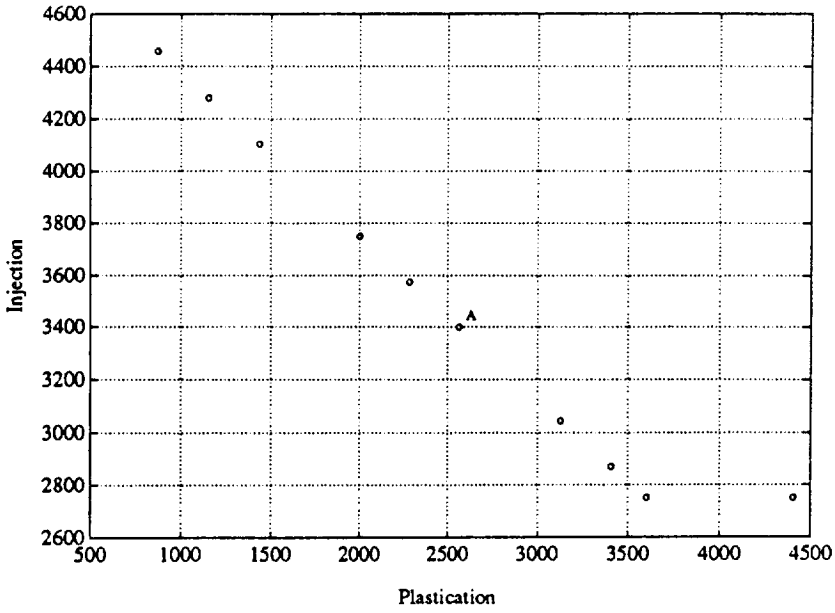

Figure 6.14: Efficient plastication and injection setpoints for the nondominated set.

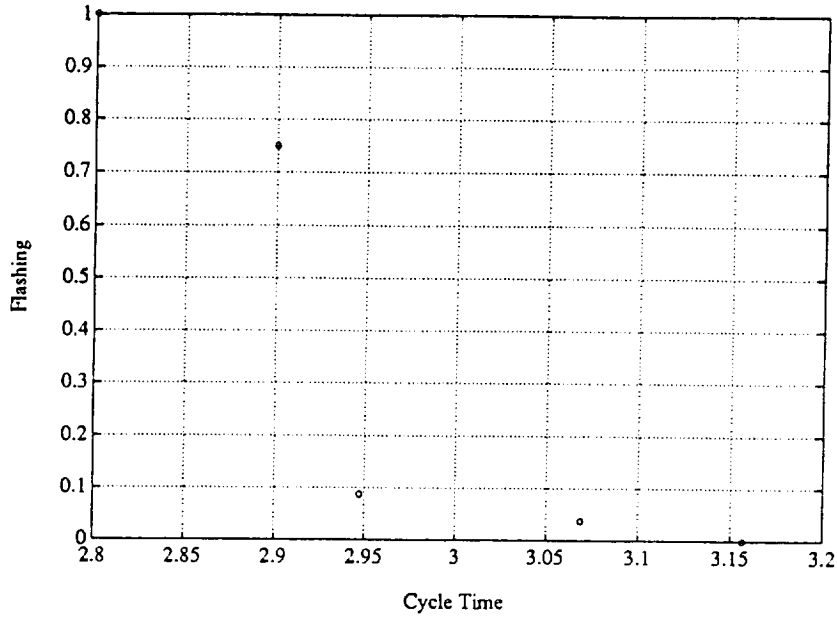

Figure 6.15: Nondominated set of cycle time versus flash, after the first algorithm iteration.

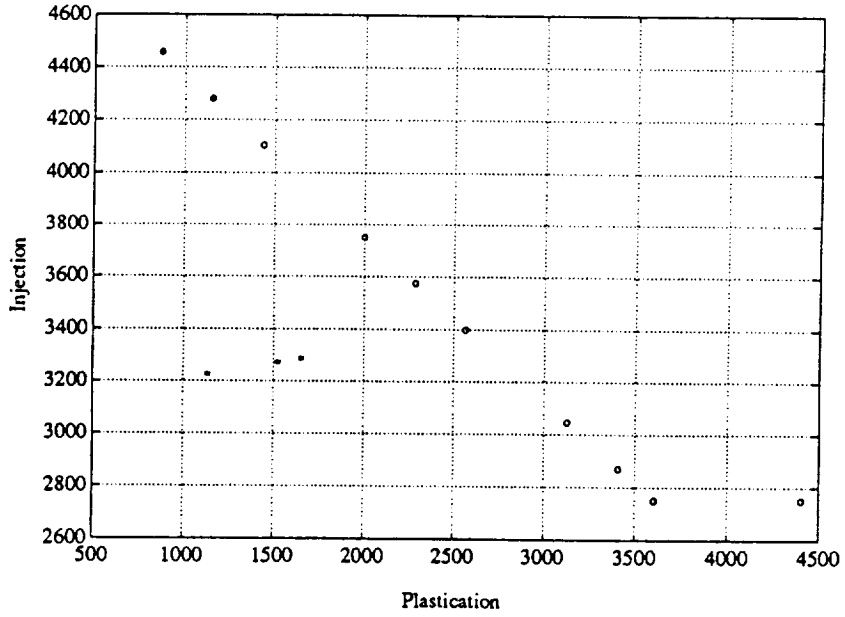

Figure 6.16: Efficient plastication and injection setpoints for the non-dominated set.

injection setpoint is increased to satisfy all objectives simultaneously. Once a dominated point was generated, a new efficient point was chosen, and iterations began again. In this instance, and the following, one, both setpoints were decreased, but with injection being decreased at a much slower rate. In the final iteration (beginning at Run 74), the injection setpoint is held constant and only the plastication setpoint is changed. This "steady-state" injection setpoint represents a value where neither underfill nor flashing will occur. The algorithm uses the remaining degree of freedom, the plastication setpoint, to decrease the cycle time.

An additional observation involves the treatment of *quality attributes.* For the ASTM mold trials, two different techniques were tried for rating these attributes:

- A *crisp* measure was defined for each of the attributes. In this case either the attribute constraint was violated (given a value of "1") or it was not violated ("0").

- A *fuzzy* measure was defined for each attribute. In this case, the operator acted as inspector and assigned a value to the severity of the attribute constraint violation.

The algorithm still managed to find the tradeoff surface in either case. The difference was in how quickly this was found. The fuzzy measures gave better information to the algorithm when it was necessary to determine a direction. Crisp measures only worked when the perturbations, used to determine the gradients (algorithm Step 2), "straddled" the boundary between violation and no violation. (In normal constrained optimization, this could be likened to an inequality constraint being suddenly activated.) In general, the trials where fuzzy measures were used involved fewer iterations to "get the process out of trouble". One caveat: the fuzzy measures, as implemented, took on a finite number of values between zero and one. If perturbations only caused the process output to fluctuate between two of these finite

values, the fuzzy measure performed no differently than the crisp measure.

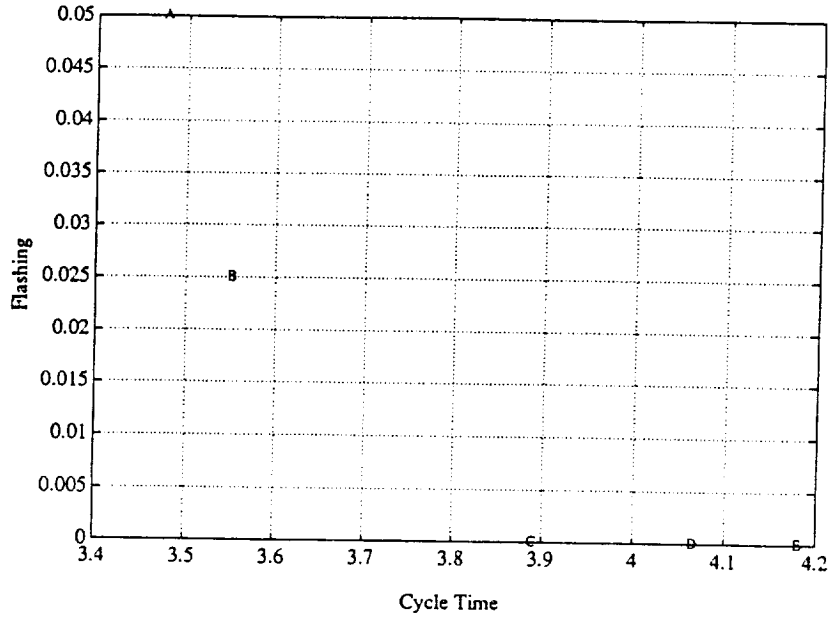

Figure 6.17: Nondominated set of cycle time versus flash, after the second algorithm iteration, under the new setpoint constraint.

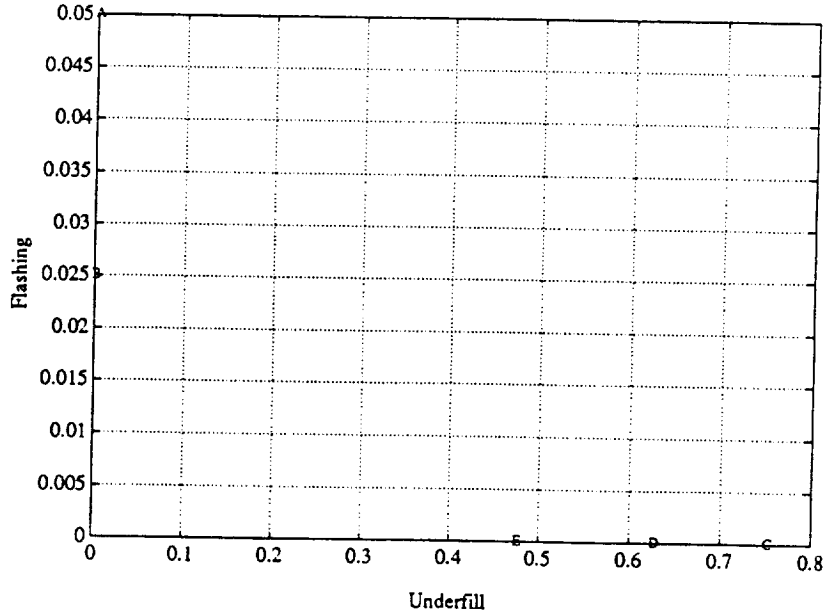

Figure 6.18: Nondominated set of underfill versus flash, after the second algorithm iteration, under the new setpoint constraint.

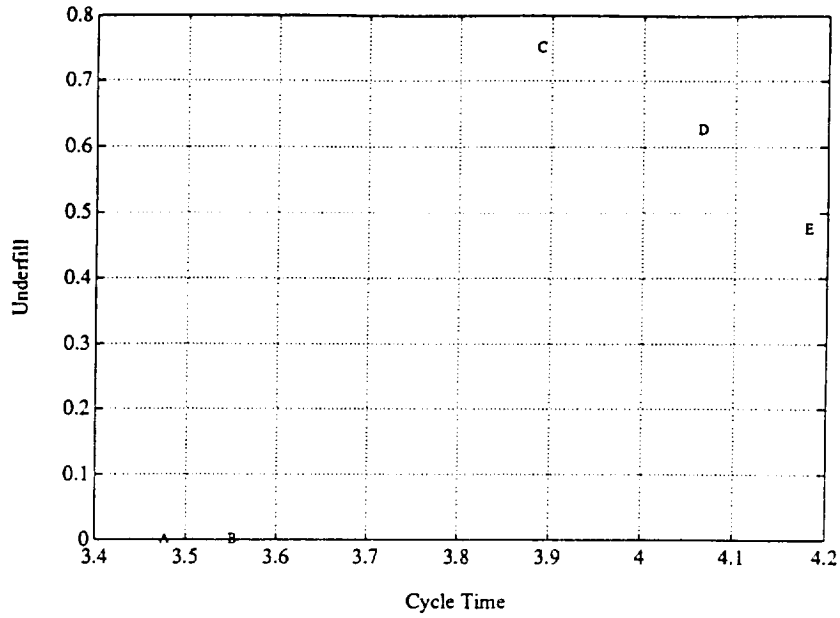

Figure 6.19: Nondominated set of cycle time versus underfill, after the second algorithm iteration, under the new setpoint constraint.

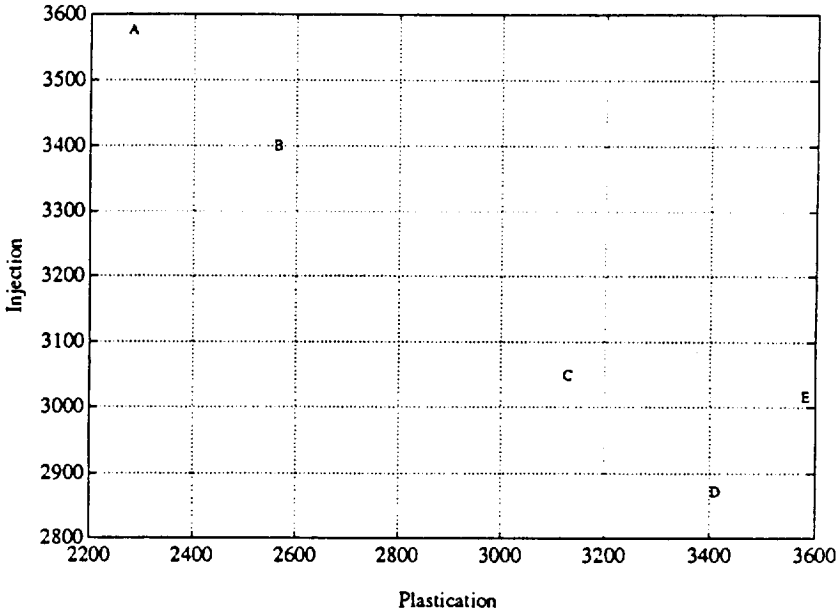

Figure 6.20: Efficient plastication and injection setpoints for the nondominated set, under the new setpoint constraint.

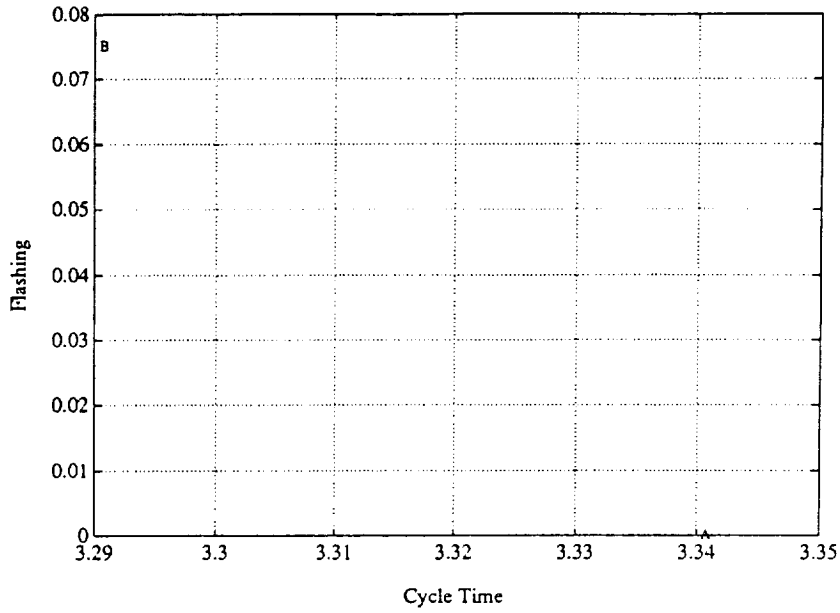

Figure 6.21: Final nondominated set of cycle time versus flash.

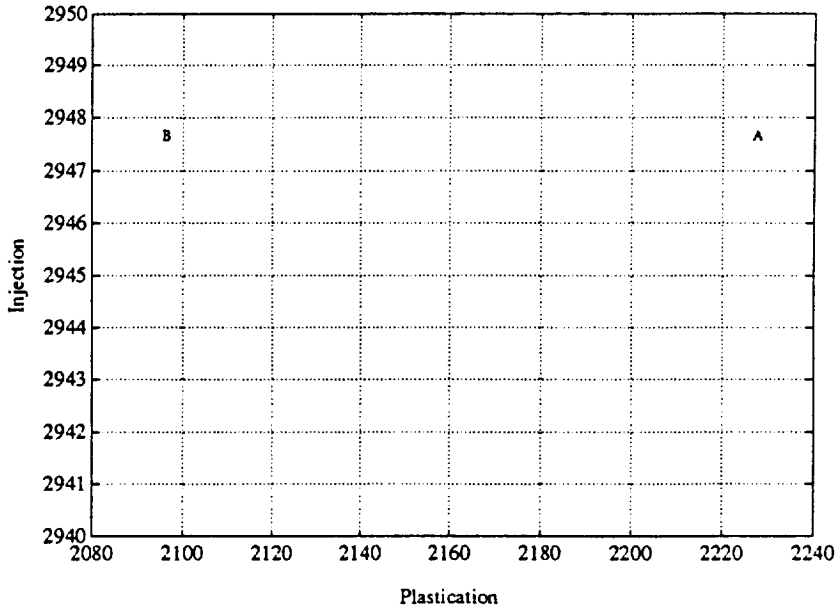

Figure 6.22: Final efficient plastication and injection setpoints.

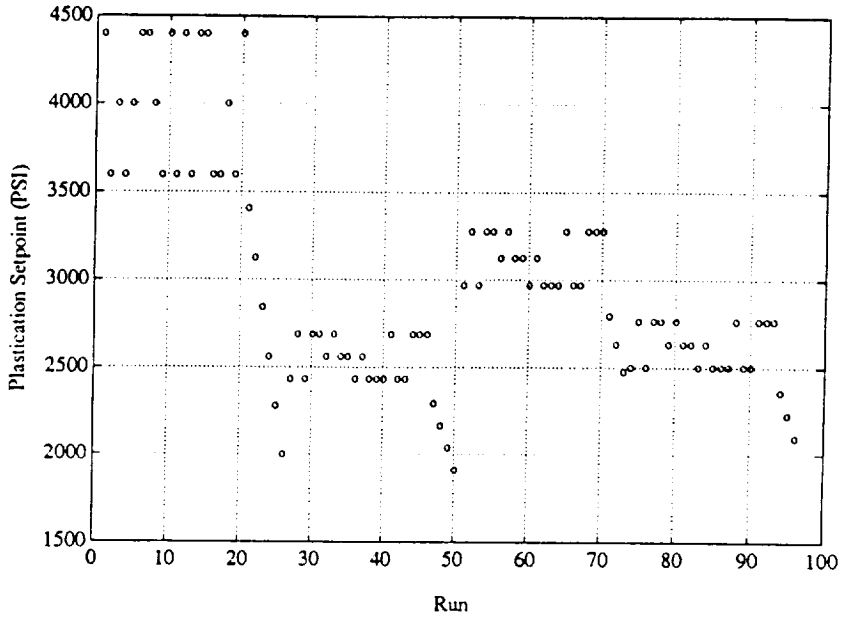

Figure 6.23: Plastication inputs for ASTM four cavity mold optimization.

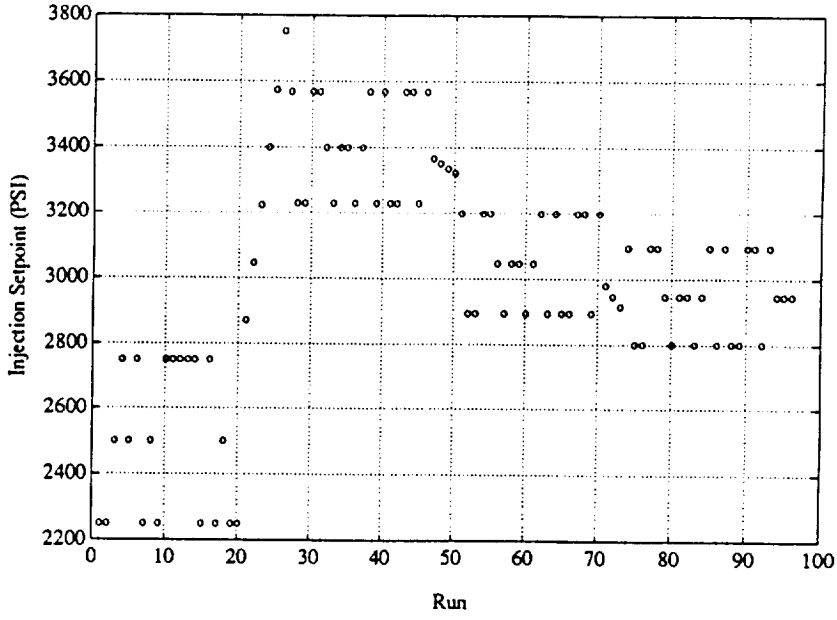

Figure 6.24: Injection inputs for ASTM four cavity mold optimization.

6.3 Conclusions

In this chapter, two case studies were presented where the multiobjective optimization algorithm was used for quality control. For each of the molds, quality objectives were defined and the algorithm was used to find inputs which were a compromise between all of these objectives.

The main difference between these cases and those discussed in the previous chapter was the presence of attribute constraints. They served to define the acceptable region of quality space, within which the quality variables would be optimized. These attributes presented the problem that they had to be integrated into an algorithm which depended upon estimating objective gradients. Using the experimental perturbations, it was possible to estimate a direction that was normal to the constraint boundary and use this with the quality objective gradients to define a direction that was used to improve the process output. A further refinement was implemented, where a fuzzy measure was used to evaluate these constraints and generate the direction normal to the boundary.

These constraints can be thought of as defining a boundary separating two regions. Far away from the boundary, the regions can be considered as taking a single value; figuratively speaking this can be thought of as two planes of height "1" outside the constraint boundary and of height "0" inside the boundary. In these plane regions, there is no information which can be used to find the direction to the other region. This information is present at the boundary. When crisp measures of the attributes are used, the boundary can be pictured as a "cliff". When fuzzy measures are used, this boundary becomes a slope, with the steepness dependent upon the membership function. The advantage found with the fuzzy measures is that they extend the range over which information about the boundary can be detected. As was found in the second case, this resulted in some savings on iterations towards the boundary. The fuzzy measure also provided a natural way to rate the

attributes. Experience showed that it tended to be harder to judge parts when the their attribute was near the "crisp threshold" whereas the fuzzy measures were more forgiving.

Behavior of the quality variables was very similar to the performance variables in the previous chapter. In fact, gradient estimation proved to be less sensitive to noisy data than was anticipated. This may also be a function of the fact that the gradients were used to determine a feasible direction, but were not used explicitly in the optimization. This additional step may provide some robustness to the algorithm. It must be remembered that this algorithm provides a "slow, plodding" approach towards optimization. There is no urgent need for speed, in fact, because the algorithm is to be run on-line, it is important that it make no moves that *surprise* the operator.

7. Conclusions

In this thesis the problem of quality control was considered from a control engineering perspective. The goal was to re-define the quality control problem such that the techniques and tools of control theory could be used. A fundamental basis of control theory is that there exists a system model which defines the functional relationship between system inputs and system outputs. Performance objectives are defined which reflect a desired relationship between these inputs and outputs. The goal is to formulate a control policy which augments the system such that the performance objectives are achieved.

In this context, the product quality was defined as the system output, and the optimization of quality was the performance criteria for manufacturing system control. The goal in manufacturing, for economic reasons, is to optimize product quality. One problem is that, as quality is a rather nebulous product characteristic, there is seldom an analytic function that can be used as a measure. Therefore, standard control approaches, such as optimal control, cannot readily be applied.

A second problem with optimizing product quality is that it is typically measured along many dimensions: there are many aspects of quality which must be optimized simultaneously. Very often these different aspects are incommensurate and competing. The concept of optimality must now include accepting tradeoffs among the different quality characteristics.

The solution to both of these problems was achieved by defining the quality control problem as a multiobjective optimization problem. This solution had the following advantages:

- Quality consists of many different factors. The multiobjective formulation allowed these factors to be incorporated simultaneously.

- A fundamental premise of multiobjective optimization is that there is a *decision maker* in the "loop". The decision maker was relied upon to evaluate those aspects of quality that could not be measured.

Automatic quality control was approached as the solution to a multiobjective optimization problem. The result was the development and implementation of automatic quality control using an algorithm which iteratively solved this multiobjective optimization problem on-line.

Performance of the multiobjective optimization algorithm was investigated using five case studies. Three case studies were centered around the tuning of PID controller gains to achieve a desired process response. For the first case study, a transfer function model of plastication was identified using input/output data. The controller was tuned in simulation, using this transfer function and performance criteria based on the process response and control action. In the second case study, plastication controller tuning was implemented to run on-line. This verified the simulation results and also demonstrated the ability of the algorithm to work in real-time. In the third case study, the algorithm was used to tune an injection PID controller. In the above cases, the algorithm was successful in generating a tradeoff surface showing the limits of performance of this controller. Also, the relationship between the nondominated points and efficient points showed how the tradeoff surface is parameterized by the controller gains.

In the final two case studies, the algorithm was used for closed-loop quality optimization for two different molded parts. For the spiral mold, quality criteria were defined based on the length of the spiral, the length of the injection cycle, and the presence of flashing. For the ASTM four cavity mold, quality criteria were defined based on the injection cycle time, the presence of flashing and the presence of underfill. In both cases, from an initial operating point, it was demonstrated that the algorithm steadily improved the performance of the process with respect to the

stated criteria, and also generated tradeoff information to be used by the operator in selecting the final operating point.

7.1 Contributions

The contributions of this work were as follows:

- Quality control was formulated as a multiobjective optimization problem. Quality is naturally measured as many possibly incommensurate and competing objectives. This feature was retained in the problem formulation.
- The multiobjective optimization formulation allowed the incorporation of both quality attributes and variables. Attributes were used to define the *feasible quality region* and variables were used to optimize within this region.
- A feedback control policy for control of quality in manufacturing processes was formulated as the solution to the multiobjective optimization problem.
- An algorithm was formulated which implemented automatic quality control. The algorithm has the following features:
  - An analytic formulation of the performance criteria (objective functions) is not necessary.
  - *A' priori* weighting of objectives is not necessary because of the multiobjective formulation.
  - Feedback is provided which informs the decision maker (DM) if the operating point is on the tradeoff surface or how far away it is.
  - If the operating point is not on the tradeoff surface, the algorithm generates input changes which will direct the system towards the tradeoff surface.

  - Determination of the tradeoff surface allows explicit tradeoff analysis by the DM.
  - The DM can interactively move along the tradeoff boundary, exploring alternate nondominated points, until the *best* operating point is found.
  - The evolutionary nature of the algorithm allows improvement to be made and information to be gained without upsetting the process excessively.

- The algorithm provides an interface between the process and the operator. By the use of this algorithm, the operator's role has been changed from one of *monitoring and operating the process* to one of *monitoring and improving quality.* This algorithm implements true quality feedback in that the operator's goal is to select the desired quality.

- The algorithm was implemented on the injection molding process and was used successfully for tuning of inputs to meet quality objectives and for tuning of controller gains to meet performance objectives *on-line.*

7.2 Future Work

This work can be extended in two general areas, improving the efficiency of the algorithm, and improving the decision maker's interface. Specific research issues are discussed in the following sections.

7.2.1 Improving Algorithm Efficiency

In the context of on-line quality optimization, requirements of the algorithm are different than those typically applied in an optimization exercise. Efficiency is important, because each iteration requires one injection cycle. More important than efficiency, though, are **safety** and **continuity**. By safety, we mean that under no circumstances should the algorithm make unpredictable operating point changes.

The algorithm relies only on data gathered from *small* perturbations about the nominal operating point. By continuity, we mean that new operating points are found by moving smoothly from old ones, and consequently process output is improved smoothly. This allows the operator to analyze trends in the process output to determine if the current operating conditions being explored are likely to yield improvements. Also, as operating conditions are changed automatically, any degradation of the output quality is minimized. With this in mind, the following improvements to the algorithm should be explored:

- The algorithm depends on estimating the objective function gradient when determining the feasible direction. This estimate is sensitive to the noisiness of the measurements, the presence of "second-order geographical features" in the response surface, and the magnitude of the input perturbation levels. It would be valuable to build in some mechanism that would provide feedback on how good the estimate is. One possible approach would be to use an F-ratio test as the measure of confidence for the gradient. This information could also be used to adjust the perturbation levels to improve gradient estimation accuracy.

- The same input perturbations are used to estimate the gradient for all of the objectives. Perturbations that are good for one objective may not provide good estimates for other objectives. Ways of dealing with the "insensitive" objectives must be determined. One concern that arises is that if a particular objective is insensitive, it may not be suitable for optimization. It may be more reasonable to use that particular objective as a constraint.

- The experimental design used to estimate the gradient was a full factorial design with center point. Given n inputs, $2^n + 1$ experiments must be run. As n gets large (more than four or five), this portion of the algorithm could

become time consuming and expensive. Other types of experimental designs, such as orthogonal arrays or fractional factorial designs, should be investigated to see if a savings can be made in the number of experiments run, without losing any of the accuracy in the gradient estimate.

- The speed with which the algorithm converges to the tradeoff surface is due, in large part, to how efficiently the line search is conducted. Different techniques of adaptive step–size control should be evaluated.

- As the algorithm iterates, the shape of the response surface with respect to each of the objectives is built up locally. Methods of incorporating this information into a global picture of the process response should be explored. One interesting question is, can this response surface be used as a "process signature", providing a useful diagnostic tool?

7.2.2 Improving the Decision Maker's Interface

The operator interface performs two functions: it allows the operator to provide information to the process and it provides the operator with information about the process.

Information from the process is presented in terms of the nondominated set and efficient points. Once there are more than three objectives, it becomes very difficult to present the tradeoff surface information in an intuitively meaningful manner. Methods of presenting this information, such as graphical or symbolic, should be investigated.

An important aspect of quality is its attribute nature. In the strictest sense, this attribute represents a binary decision; the part is good or bad. Often there is a gray area associated with this decision. A fuzzy interface was used for this work, but this still depended on the operator's interpretation. This suggests two areas of research: a determination of the effect of inconsistent decision making on the

algorithm and an investigation of ways in which the algorithm can automatically calibrate itself to a decision maker.

8. Literature Cited

[1] Freund, R. A. Definitions and Basic Quality Concepts. *Journal of Quality Technology*, January 1985, Vol. 17, No. 1, pp 50–56.

[2] Johannaber, F. *Injection Molding Machines: A Users Guide.* Macmillan, 1982.

[3] Ma, C. Y. W. A Design Approach to a Computer-Controlled Injection Molding Machine. *Polymer Science and Engineering*, November 1974, Vol. 14, No. 11, pp 768–772.

[4] Tadmor, Z. and C. Gogos. *Principles of Polymer Processing.* John Wiley and Sons, New York, 1979.

[5] Box, G. E. P. and N. R. Draper. *Empirical Model-Building and Response Surfaces.* John Wiley and Sons, Inc, New York, 1987.

[6] Sorenson, H. W. *Paremeter Estimation: Principles and Problems.* M. Dekker, New York, 1980.

[7] Spencer, R. S. and G. D. Gilmore. Equation of State for High Polymers. *Journal of Applied Physics*, June 1950, Vol. 21, pp 523–526.

[8] Gilmore, G. D. and R. S. Spencer. Role of Pressure, Temperature, and Time in the Injection Molding Process. *Modern Plastics*, April 1950.

[9] Donovan, R. C, D. E. Thomas, and L. D. Leversen. An Expermental Study of Plasticating in a Reciprocating Screw Injection Molding Machine. *Polymer Science and Engineering*, September 1971, Vol. 11, No. 5, pp 353–367.

[10] Donovan, R. C. A Theoretical Melting Model Model for a Reciprocating

Screw Injection Molding Machine. *Polymer Science and Engineering*, September 1971, Vol. 11, No. 5, pp 361–368.

[11] Donovan, R. C. The Plasticating Process in Injection Molding. *Polymer Science and Engineering*, February 1974, Vol. 14, No. 2, pp 101–111.

[12] Lipshitz, S. D., R. Lavie, and Z. Tadmor. A Melting Model for Reciprocating Screw Injection Molding Machines. *Polymer Science and Engineering*, August 1974, Vol. 14, No. 8, pp 553–559.

[13] Raimund, C. Control of Melt Temperature in Thermoplastic Injection Molding. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY, May 1988.

[14] Kamal, M. R., W. I. Patterson, and V. G. Gomes. An Injection Molding Study, Part I: Melt and Barrel Temperature Dynamics. *Polymer Engineering and Science*, Mid-July 1986, Vol. 26, No. 12, pp 854–866.

[15] Gomes, V. G., W. I. Patterson, and M. R. Kamal. An Injection Molding Study, Part II: Evaluation, of Alternative Control Strategies for Melt Temperature. *Polymer Science and Engineering*, Mid-July 1986, Vol. 26, No. 12, pp 854–866.

[16] Tadmor, Z., E. Broyer, and C. Gutfinger. Flow Analysis Network (FAN) - A Method for Solving Flow Problems in Polymer Processing. *Polymer Engineering and Science*, September 1974.

[17] Kreuger, W. L. and Z. Tadmor. Injection Molding into a Rectangular Cavity with Inserts. *Polymer Engineering and Science*, April 1980, Vol. 20, No. 6, pp 426–431.

[18] Mavridis, H., A. Hrymak, and J. Vlachopoulos. Finite Element Simulation of

Fountain Flow in Injection Molding. *Polymer Engineering and Science*, Mid-April 1986, Vol. 26, No. 7, pp 449–454.

[19] Kamal, M. R., E. Chu, P. G. Lafleur, and M. E. Ryan. Computer Simulation of Injection Mold FIlling for Viscoelastic Melts with Fountain Flow. *Polymer Engineering and Science*, Mid-February 1986, Vol. 26, No. 3, pp 190–196.

[20] Gogos, C. G., C. Huang, and L. R. Schmidt. The Process of Cavity Filling Including the Fountain Flow in Injection Molding. *Polymer Engineering and Science*, Mid-November Polymer Engineering and Science, Vol. 26, No. 20, pp 1457–1466.

[21] Shankar, A. *Dynamic Modeling and Control of Injection Molding Machines.* PhD thesis, Carnegie-Mellon University, Pittsburgh, PA, 1977.

[22] Shankar, A. and F. W. Paul. A Mathematical Model for the Evaluation of Injection Molding Machine Control. *Transactions of the ASME*, March 1982, Vol. 104, pp 86–92.

[23] Costin, M. H., D. A. Okonski, and J. C. Ulicny. Control of an Injection Molding Machine: Adaptive Regulation During Mold Filling. In *Proc. of the 1986 American Control Conference*, pages 711–716, 1986.

[24] Haber, A. and M. R. Kamal. The Dynamics of Peak Cavity Pressure in Injection Molding. *Polymer Science and Engineering*, Mid-October 1987, Vol. 27, No. 18, pp 1411–1418.

[25] Kamal, M. R., W. I. Patterson, N. Conly, D. Abu Fara, and G. Lohfink. Dynamics and Control of Pressure in the Injection Molding of Thermoplastics. *Polymer Science and Engineering*, Mid-October 1987, Vol. 27, No. 18, pp 1403–1410.

[26] Hoare, L. and D. Hull. The Effect of Orientation on the Mechanical Properties of Injection Molded Polystyrene. *Polymer Engineering and Science*, March 1977, Vol. 17, No. 3, pp 204–212.

[27] Bayer, R. K., H. G. Cachmann, J. J. Balta Calleja, and H. Umbach. Properties of Elongational Flow Injection- Molded Polyethylene. Part 1: Influence of Mold Geometry. *Polymer Engineering and Science*, Mid-February 1989, Vol. 29, No. 3, pp 186–192.

[28] Lopez Cabarcos, E., R. K. Bayer, H. G. Zachmann, J. J. Balta Callega, and W. Meins. Properties of Elongational Flow Injection- Molded Polyethylene. Part 2: Influence of Processing Parameters. *Polymer Engineering and Science*, Mid-February 1989, Vol. 29, No. 3, pp 193–201.

[29] Whelan, A. The Need for Improved Control Systems. *Plastics and Rubber: Processing*, September 1976, pages 135–140.

[30] Sanschagrin, B. Process Control of Injection Molding. *Polymer Science and Engineering*, Mid-June 1983, Vol. 23, No. 8, pp 431–438.

[31] Agrawal, A. R., I. O. Pandelidis, and M. Pecht. Injection Molding Process Control - A Review. *Polymer Science and Engineering*, Mid-October 1987, Vol. 27, No. 18, pp 1345–1357.

[32] Hsieh, D. S. An Exploration into Taguchi Off-Line Quality Control Methods with Applications to the Injection Molding Process and a Photocopy Machine Carriage Damper Design. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY, September 1988.

[33] Bernhardt, E. C., G. Bertacchi, and A. Moroni. Molding and Cost Optimization. In Bernhardt, E. C., editor, *CAE for Injection Molding*, pages 222–273. Macmillan, 1983.

[34] Jamieson, Archibald. *Introduction to Quality Control.* Reston Publishing Company, Reston, VA, 1982.

[35] Kamal, M. R. and G. L. Bata. Current Trends in the Analysis of Polymer Shaping Operations. In Astarita, G. and L. Nicolais, editors, *Polymer Processing and Properties*, pages 1–45. Plenum Press, New York, 1983.

[36] Redlitz, K. R. Design and Implementation of a Plastic Injection Molding Machine Controller. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY, September 1987.

[37] Redlitz, K. R. Technical Information on the Plastic Injection Machine Controller. Technical Report RAL 103, Robotics and Automation Laboratory, Rensselaer Polytechnic Institute, September 1987.

[38] Taguchi, G. *Introduction to Quality Engineering: Designing Quality into Products and Processes.* Asian Productiviy Organization, 1986.

[39] D'Azzo, and Houpis. *Linear Control System Analysis and Design.* McGraw-Hill Book Company, New York, 1981.

[40] Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley and Sons, New York, 1986.

[41] Derringer, G. and R. Suich. Simultaneous Optimization of Several Response Variables. *Journal of Quality Technology*, October 1980, Vol. 12, No. 4, pp 214–219.

[42] Cohon, J. L., editor. *Multiobjective Programming and Planning.* Academic Press, Inc., New York, 1978.

[43] Zeleny, M., editor. *Multiple Criteria Decision Making.* McGraw-Hill Book Company, New York, 1982.

[44] Sawaragi, Y., H. Nadayama, and T. Tanino, editors. *Theory of Multiobjective Optimization.* Academic Press, Inc., New York, 1985.

[45] Box, G. E. P., W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters.* John Wiley and Sons, Inc, New York, 1978.

[46] Benayoun, R., J. de Montegolfier, and J. Tergy. Linear Programming with Multiple Objective Functions: Step Method (STEM). *Mathematical Programming*, 1971, Vol. 1, No. 3, pp 366–375.

[47] Geoffrion, A. M., J. S. Dyer, and A. Feinberg. An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department. *Management Science*, December, Part I 1972, Vol. 19, No. 4, pp 357–368.

[48] Musselman, K. and J. Talavage. A Tradeoff Cut Approach to Multiple Objective Optimization. *Operations Research*, 1980, Vol. 28, No. 6, pp 1424–1435.

[49] Zionts, S and J. Wallenius. An Interactive Programming Method for Solving the Multiple Criteria Problem. *Management Science*, 1972, Vol. 28, No. 3, pp 785–793.

[50] Schy, A. P. and D. P. Giesy. Multicriteria Optimization Methods for Design of Aircraft Control Systems. In Stadler, W., editor, *Multicriteria Optimization in Engineering and in the Sciences*, pages 225–262. Plenum Press, New York, 1988.

[51] Kuhn, H. W. and A. W. Tucker. Nonlinear Programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, California, 1951.

[52] Mangasarian, O. *Nonlinear Programming.* McGraw-Hill Book Company, New York, 1969.

[53] Avriel, M. *Nonlinear Programming Analysis and Methods*, pages 463–494. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[54] Box, G. E. P. and N. R. Draper. *Evolutionary Operation.* John Wiley and Sons, Inc, New York, 1969.

[55] Luenberger, D. G. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley Publishing Co., Reading, MA, 1973.

[56] *UNIWARE Software Development System [Release 3.2].* Software Development Systems, Inc., Downers Grove, IL, 60515, 1986.

[57] Moler, C., J. Little, and S. Bangert. *PC-MATLAB User's Guide.* The Mathworks, Inc., Sherborn, MA 01770, 1087.

[58] Ziegler, J. G. and N. B. Nichols. Optimum Settings for Automatic Controllers. *Transactions of the ASME*, November 1942, pages 759–768.

[59] Åström, K. J. and T. Hägglund. Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins. *Automatica*, 1984, Vol. 20, No. 5, pp 645–651.

[60] Franklin, G. F. and J. D. Powell. *Digital Control.* Addison-Wesley, Reading, MA, 1981.

[61] Lockwood, R. Personal communication with Russ Lockwood of Hunkar, Inc.

[62] Merritt, H. E. *Hydraulic Control Systems.* John Wiley and Sons, 1967.

[63] Ljung, L. *System Identification Toolbox.* The Math Works, Cambridge, MA, 1988.

[64] Årzeén, K. E. *Realization of Expert System Based Feedback Control.* PhD thesis, Lund Institute of Technology, Lund, Sweden, 1987.

[65] Graham, D. and R. C. Lathrop. The Synthesis of "Optimum" Transient Response: Criteria and Standard Forms. *Trans. of the AIEE*, November 1953, pages 273–288.

[66] Karas, F. C. Spiral mold for thermosets. *Modern Plastics*, September 1963, Vol. 41, pp 140–142.

[67] Desrochers, A. A. Design and Analysis of Successive Approximation Algorithms via Bilinear Control Theory. *IEEE Trans on AC*, June 1981, Vol. AC-26, No. 3, pp 770–773.

[68] Seaman, C. M. A Multiple Objective Optimization Approach to Quality Control. Technical Report RAL 124, Robotics and Automation Laboratory, Rensselaer Polytechnic Institute, May 1990.

[69] Ingersoll, L. R., O. J. Zobel, and A. C. Ingersoll. *Heat Conduction with Engineering, Geological, and other Applications*, pages 190–198. University of Wisconsin Press, Madison, WI, 1954.

[70] Bird, R.B., W.E. Stewart, and E. N. Lightfoot. *Notes on Transport Phenomena.* John Wiley and Sons, Inc, New York, 1958.

[71] Donovan, R. C. A Theoretical Melting Model Model for Plasticating Extruders. *Polymer Engineering and Science*, May 1971, Vol. 3, No. 5, pp 247–257.

APPENDIX A
The Injection Molding Process

Plastic injection molding is a repetitive process used to make complexly shaped plastic parts, with one part being made each cycle. The goal of the injection molding machine operator is to consistently produce parts which meet some predefined quality specifications. Part quality can be measured in terms of suitable material properties such as strength, surface finish, and shape. The key word above is *consistently.* The time-varying nature of the process and the complex relationships between molding conditions and part properties make this a difficult task. In this appendix, the dynamics of each of the phases will be presented. From this, an understanding of the important process parameters and the interaction between the phases will be developed. This development is based on other studies of the dynamics of injection molding machines but will be specific to the AML Arburg machine [21, 22].

A.1 The Injection Molding Cycle

Each cycle of the process has four phases, which occur in the following order:

1. *Plastication* - the polymer is melted and deposited in front of an injecting mechanism. This continues until the required amount of polymer has been accumulated.

2. *Injection* - the molten polymer is forced into the mold under high pressure. This continues until the mold is completely full.

3. *Holding* - the molten polymer is held in the mold, under pressure, to insure complete mold filling. Cooling of the part begins at this point. Holding

continues until the gate has frozen, at which point no more polymer can enter the mold cavity.

4. *Cooling* - the polymer is allowed to cool in the mold. This continues until the temperature of the formed part is low enough so that it will retain its shape once ejected from the mold.

A cut-away view of a reciprocating screw injection molding machine is shown in Figure [1.1]. In this type of machine, the screw is used to both *melt* the polymer and *pump* it into the nozzle area. This melting and pumping is accomplished by rotation of the screw. As the melting and pumping occurs, the screw retracts, creating the volume occupied by the polymer melt. Injection and holding are accomplished by forcing the screw forward, like a piston. (A check valve at the screw tip prevents a flow of polymer back into the screw.) Once the polymer is in the mold, cooling begins, and at a predetermined time, the mold opens and the solidified part is ejected. There are other types of injection molding machines which accomplish these four phases via different mechanisms, but the phases and sequencing are essentially the same [2].

A sequential block diagram of the molding cycle is shown in Figure [1.2]. Interactions between these phases occur when one phase establishes the initial conditions for a subsequent phase [3]. It is the complexity of these interactions which motivates the need for control of the process. These interactions are shown in Figure [1.3] and will be discussed in detail at the end of this appendix.

In this appendix, the underlying physical processes in each phase will be described in detail. The relationship between the inputs and outputs will be discussed, and where possible, these relationships will be quantified with a differential equation model. All models will be specific to the Advanced Manufacturing Laboratory (AML) Arburg Allrounder injection molding machine, but the principles under which they are derived will be general to the process. (Specifications for this

injection molding machine can be found in Chapter 4.)

A.2 Plastication

Plastication is the portion of the injection molding cycle in which the solid polymer pellets are melted and deposited in the nozzle chamber (in front of the retracting screw) prior to injection. The pellets are fed from the hopper into the reciprocating screw. The screw rotates, simultaneously melting and pumping the pellets forward. As the pumping progresses, a mass of molten polymer (the shot) accumulates in the nozzle. Plastication continues until a predetermined shot size has been accumulated. Plastication can begin any time after the gate has frozen shut, which prevents the accumulating shot from "leaking" into the mold.

From the above scenario, it is obvious that there are two dynamic processes occurring concurrently. There is a thermodynamic process because of the heat input and associated melting. There is also a fluid dynamic process because of the pumping which is used to accumulate the shot. Both dynamic processes interact. In the following sections, the dynamics of each process and the mechanism of interaction will be discussed.

A.2.1 Polymer Melting (Thermodynamics)

Polymer melting takes place due to heat input. There are two sources of heat input:

- *Conduction* - energy from the barrel heater bands is conducted into the plastic;
- *Viscous Dissipation* - as the molten polymer is sheared, adjacent fluid laminae rub, transforming kinetic energy into frictional heat.

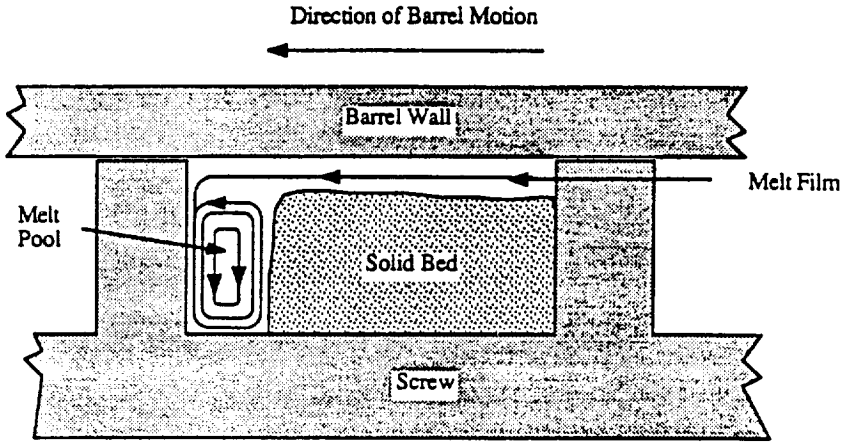

Figure A.1: Cross Section of One Screw Flight During Plastication

During the injection molding cycle, conduction heating is always taking place because the heater bands are always on. On the other hand, viscous dissipation contributes to melt temperature only during screw rotation. The goal of modeling the polymer melting is to determine the machine inputs which affect melt temperature and to develop models which will allow the control of melt temperature.

Figure [A.1] shows a cut-away view of one screw flight during plastication [11]. The heater bands contact the barrel at this point, and maintain the barrel temperature, T_b. This is the energy source for conduction heating. During plastication, there is both an axial screw translation corresponding to the shot accumulation rate and a rotation about the screw axis which corresponds to the screw rotational speed. This motion provides the shearing of the plastic for frictional heat input. Melting only takes place near the barrel. During plastication, this melt is swept into the melt pool and the solid bed continuously rearranges itself. This phenomenon is described more fully by Donovan, *et al.* , in [9].

The relative contribution of the conduction heat versus viscous dissipation must be understood so that a control strategy for melt temperature can be developed. In the following sections, the temperature dynamics due to conduction and viscous dissipation will be presented analytically. This will provide a functional relationship between machine inputs and temperature which will be useful for experimental design and empirical modeling of the system. It must be emphasized that this analytical modeling will only provide a general clue to the relationships, the actual situation is much more complex.

Polymer Melting Due to Conduction Heating

Donovan, in [10], developed a theoretical melting model for the polymer. In this paper, he was concerned with both the melt profile of the polymer in the barrel (ratio of melted to unmelted polymer as a function of screw distance) as well as the temperature of the molten polymer in the barrel. To develop an accurate approximation of the conduction melting model, the problem was posed as a one-dimensional heating problem with boundary conditions that depend on the melt film thickness. The solid bed is assumed to be a semi-infinite mass. This is known as *Neumann's Problem* [69]. Figure [A.1] shows the cross-section of one screw flight, which was used as a control volume to develop the model.

In the melt film, the one-dimensional transient heat conduction equation is

$$\frac{\partial T_f}{\partial t} = \alpha_m \frac{\partial^2 T_f}{\partial y^2}; \qquad 0 < y < \delta \tag{A.1}$$

T_f is the melt film temperature profile, the melt film thickness is δ and α_m is the thermal diffusivity of the melt. y is the coordinate direction starting at the barrel/melt film interface and increasing towards the solid bed.

In the solid bed, the conduction equation is

$$\frac{\partial T_s}{\partial t} = \alpha_s \frac{\partial^2 T_s}{\partial y^2}; \qquad \delta < y < \infty \tag{A.2}$$

T_s is the solid bed temperature profile and α_s is the thermal diffusivity of the solid bed. The initial conditions and boundary conditions are

$$\begin{aligned} T_f(y) &= T_b; & y &= 0 \\ T_f(y) &= T_s(y) = T_m; & y &= \delta \\ T_s(y) &\rightarrow T_r; & y &\rightarrow \infty \end{aligned} \tag{A.3}$$

T_b is the temperature of the barrel, T_m is the temperature at the melt film/solid bed interface, and T_r is a reference temperature, assuming the solid bed has infinite depth. In practice, this would be the steady-state temperature that is achieved at the center of the screw.

An energy balance was applied at the melting interface, and Donovan was able to derive the following solutions to the transient temperature problem in the melt film and the solid bed.

$$\frac{T_f(y)-T_m}{T_b-T_m} = 1 - \frac{\mathrm{erf}(y/2\sqrt{\alpha_m t})}{\mathrm{erf}(\delta/2\sqrt{\alpha_m t})}; \qquad 0 < y < \delta \tag{A.4}$$

and

$$\frac{T_s(y)-T_m}{T_r-T_m} = 1 - \frac{\mathrm{erfc}(y/2\sqrt{\alpha_s t})}{\mathrm{erfc}(\delta/2\sqrt{\alpha_s t})}; \qquad \delta < y < \infty \tag{A.5}$$

Temperature predictions based on this analysis were found to correlate well with the data near the start of melting but to underestimate it at the conduction melting near the end of the screw. The authors attribute this to the neglection of melting at the other three faces within the channel. (There may also be some melting due to heat generation during viscous dissipation. There is no accounting for an additional heat source in this model.) Equation (A.4) is the equation of interest since this predicts the transient temperature behavior in the melt.

Polymer Melting Due to Viscous Dissipation

As a first step towards understanding the melting due to viscous dissipation, a simple example from [70] will be presented. This example will then be compared with the situation actually occurring during plastication.

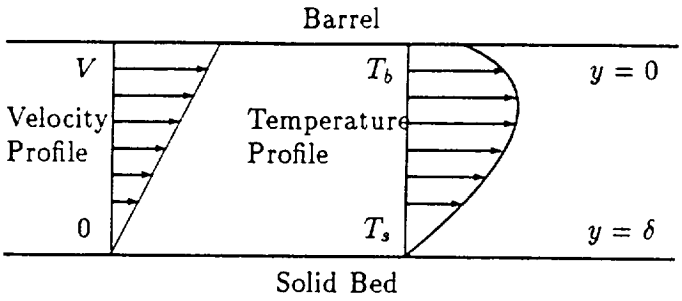

Figure A.2: Two Parallel Plates Separated by a Newtonian Fluid

Assume that the barrel and solid bed can be modeled by two parallel plates separated by the thickness of the melt film, which stays constant. Also assume that the molten polymer can be modeled as a Newtonian fluid with viscosity μ. The barrel is moving at a constant velocity V_s, relative to the solid bed. This situation is depicted in Figure [A.2]. This is analogous to the situation shown in Figure [A.1], if the thickness of the melt film is small compared to the radius of the barrel and solid bed.

This system has conduction heating as well as viscous dissipation, therefore the energy equation is

$$0 = \alpha_m \frac{d^2 T_f(y)}{dy^2} + \tau_{yx} \frac{dV_x(y)}{dy} \tag{A.6}$$

The first right hand side term in this equation is energy from heat conduction and the second term is energy from viscous dissipation. $V_x(y)$ is velocity in the x-direction as a function of y and τ_{yx} is shearing stress in the x-direction due to a force in the y-direction. Since the fluid is Newtonian, the following relationship holds:

$$\tau_{yx} = \mu \frac{dV_x(y)}{dy} \tag{A.7}$$

The steady state velocity profile is

$$V_x(y) = \frac{y}{\delta} V_s \tag{A.8}$$

The boundary conditions are

$$\begin{aligned} T_f(y) &= T_b; \qquad y = 0 \\ T_f(y) &= T_s(y); \qquad y = \delta \end{aligned} \tag{A.9}$$

Equations (A.7) and (A.8) can be substituted into (A.6), and the solution for the resulting differential equation is

$$\frac{T_f(y) - T_s}{T_b - T_s} = \frac{y}{\delta} + \frac{1}{2}\mathrm{Br}\frac{y}{\delta}(1 - \frac{y}{\delta}) \tag{A.10}$$

where

$$\mathrm{Br} = \frac{\mu V_s^2}{\alpha_m(T_b - T_s)} \tag{A.11}$$

Br is the Brinkman number. It is an important dimensionless quantity which is the ratio of the heat generated via viscous dissipation (the numerator) to the ability of the fluid to conduct this heat away (the denominator).

The steady-state temperature profile has two components. There is a conduction component which makes the temperature a linear function of the distance from the top plate, as expected. The second term is the viscous dissipation component. The temperature profile for this component is parabolic. If the Brinkman number is large enough, there will be a maximum temperature, somewhere between the two plates, which is greater than either T_b or T_s, as shown in Figure [A.2].

An exact solution to the steady-state temperature profile of a screw flight for a plasticating extruder has been carried out by Donovan in [71]. It will be reviewed here briefly, as a comparison to the simplicity of Equation (A.10). Again, the analysis is begun with the energy equation.

$$0 = \alpha_m \frac{d^2T_f(y)}{dy^2} + \tau_{yx}\frac{dV_x(y)}{dy} \tag{A.12}$$

Now, the molten polymer is assumed to be non-Newtonian. The constitutive equation, assuming a power-law model, is

$$\tau_{yx} = m_0 e^{a(T_f(y) - T_m)} \left(\frac{dV_x}{dy}\right)^n \tag{A.13}$$

This is an empirical equation where m_0, a, and n must be determined through experimentation.

If (A.13) is substituted into (A.12) and the resulting differential equation solved, the following temperature distribution is found.

$$T_f(y) = T_m + \frac{n}{a}\log\left[\left\{\cosh^2\left(C_1 - B_1 C_2^{\frac{n+1}{2}} y\right) C_0^{-2}\right\}\right] \tag{A.14}$$

The constants C_0, C_1, and C_2 must be found using the following initial conditions.

$$\begin{aligned} T_f(y) &= T_b; \quad & y &= 0 \\ T_f(y) &= T_s; \quad & y &= \overline{\delta} \\ V_x(y) &= V_s; \quad & y &= 0 \\ V_x(y) &= 0; \quad & y &= \overline{\delta} \end{aligned} \tag{A.15}$$

$\overline{\delta}$ is the average thickness of the melt film. The constant B_1 is defined as

$$B_1 = \sqrt{a m_0 (V_x^{n+1})(\overline{\delta}^{1-n})/2n\alpha_m} \tag{A.16}$$

Notice that this constant is a ratio similar to the Brinkman number defined by Equation (A.11), but due to the complexity of the mathematics, a simple form does not result. The point to be emphasized is that there is no longer an intuitive relationship between the effect of conduction heating versus viscous heating. This must be developed experimentally.

Overall Melting Dynamics

Two different mechanisms for heat generation have been investigated. Equations (A.4), (A.5), and (A.14) are useful for understanding the physical relationships in the process, but they are too complex to be used for empirical modeling.

Fundamentally, the different mechanisms work on two different time scales. The conduction transients are quite slow, because of the "thermal mass" associated with the polymer and barrel. This dynamic response was investigated both theoretically and experimentally by Kamal, *et al.* , in [14, 15]. Their results for the

melt temperature response to the barrel heaters, using a lumped parameter model, was a second-order, overdamped system with a time delay. This seems reasonable since a thermal system can typically be modeled as a first-order system and that the barrel and polymer melt would represent two cascaded first-order systems (*i.e.* , the heater heats the barrel and the barrel heats the polymer). A transfer function for this system would be

$$T_{mc}(s) = \frac{K_c e^{-sd}}{(\tau_b s + 1)(\tau_m s + 1)} T_b(s) \tag{A.17}$$

In the above equation, T_{mc} is the component of the melt temperature due to conduction heating. The constants K_c, d, τ_b, and τ_m must be found via experimentation. It should be noted that screw speed was not manipulated in any of the experiments discussed by Kamal, *et al.* , therefore any effect due to changes in viscous dissipation would not appear in their results.

Heat input from viscous dissipation only occurs during screw rotation. This can be thought of as a periodic step disturbance to the melt temperature. A model of this phenomenon was developed by Lipshitz, *et al.* , in [12]. This work was mainly concerned with how this disturbance affected the solid bed profile, but analogous conclusions can be drawn for the melt temperature profile.

During plastication, the melt film is swept into the melt pool, therefore the melt pool will "asymptotically" approach the volume average temperature of the melt film. The temperature of the melt pool is therefore governed by the speed of screw rotation as well as the length of time of screw rotation.

Equation (A.10) is an approximation of the steady-state melt temperature profile during plastication. This equation indicates that the melt temperature will be a function of the shear velocity. Now, as the screw rotates, the melt film depth stays approximately constant and the solid bed is continuously rearranged [71]. Therefore, the melt pool volume in the screw flight is increased by a constant stream of polymer from the melt film, at a volume average temperature determined by the

viscous dissipation. This is analogous to the situation of a constant stream influx to a well stirred tank. The lumped parameter temperature dynamics of this situation is represented quite well by a first-order system. Since conduction dynamics have been accounted for, the input to the mixing dynamic system is shear velocity. A simple transfer function representing this system would be

$$T_{mv}(s) = \frac{K_v}{\tau_{vd}s + 1} V_y(s) \tag{A.18}$$

T_{mv} is the component of temperature due to viscous dissipation. The shear velocity, V_y, is both a function of screw translational velocity, V_s, known as *recovery rate*, and screw rotational velocity, ω, known as *screw RPM*. The following approximation to the numerator of (A.18) can be made.

$$K_v V_y = K_{vs} V_s + K_{v\omega}\omega \tag{A.19}$$

K_{vs} and $K_{v\omega}$ are determined experimentally. This is essentially a linear approximation of the function between these velocities. Then, using this equation, (A.18) becomes

$$T_{mv}(s) = \frac{K_{vs}V_s(s) + K_{v\omega}\omega(s)}{\tau_{vd}s + 1} \tag{A.20}$$

Equation (A.20) can be used to determine the effect of the three primary plastication variables, screw RPM, recovery rate, and recovery time (t_r), on the melt temperature. Recovery time and recovery rate are tightly coupled. Shot size, X_s, is typically measured in terms of the distance the screw must retract to develop the required volume of melt. Therefore,

$$X_s = t_r \overline{V_s} \tag{A.21}$$

In the above equation, $\overline{V_s}$ is the average or steady-state recovery rate. The relationship between t_r and τ_v will give some indication of the degree of control that can be exercised over the melt temperature in the time available.

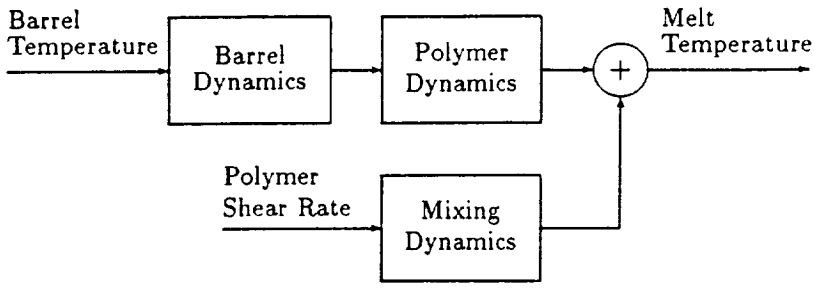

Figure A.3: The Melt Temperature Dynamic System

Recovery rate and screw RPM are loosely coupled through the screw pumping dynamics, which will be discussed in the next sections. At this time it will just be noted that it is important to determine the magnitudes of K_{vs} and $K_{v\omega}$. This will indicate the degree of temperature control which can be attained using these velocities. At the same time, this control is constrained by the dynamic coupling between these velocities, as well as the range over which they can be varied.

In summary, the system, depicted in Figure [A.3], can be used to empirically model the system temperature dynamics. In this model, the barrel and polymer conduction dynamics are much slower than the viscous dissipation dynamics. With respect to the conduction dynamics, the viscous dissipation dynamics would appear as an exponential disturbance, with dynamics dependent upon the mixing, where the duration of the disturbance depends upon the rotation time. With respect to the viscous dissipation dynamics, the conduction dynamics establish the initial temperature of the melt pool. The effect of the viscous dissipation is dependent on the screw *rotation time*. This is directly related to the desired *shot size* and the *accumulation rate*. In the following sections, the machine dynamics involved with accumulating the shot will be derived.

A.2.2 Polymer Pumping (Fluid Dynamics)

In this section, the dynamics of pumping the polymer into the nozzle region will be presented. Raimund, in [13], presented an analysis of the melt conveyance assuming the system was isothermal, non-Newtonian, and steady-state. A solution for the volume flow rate was obtained. Since it was steady-state, it wasn't particularly applicable for control system design. In the following section, a lumped-parameter analysis of the system will be done, using simple mass and force balances. The analysis will not be as rigorous as [13], but the basic dynamics and interactions of the system will be captured. It should be noted at the outset that the coefficients of the model will be dependent on the particular machine being studied, the polymer being used, and the operating conditions. Ideally, these would be found through designed experiments or on-line identification.

As the screw rotates, molten polymer is forced into the nozzle, generating a pressure in the melt. The accumulation of the polymer is modeled by a simple mass balance. The melt pressure in the nozzle forces the screw backwards against the hydraulic oil in the piston chamber. These screw dynamics can be modeled by a force balance between the "polymer-side" and the "hydraulic-side". Finally, the hydraulic oil flow out of the piston is regulated by a servovalve. This can be modeled by another mass balance plus the dynamics of the valve. In the following sections, the analysis outlined above will be presented, yielding a simple melt-filling transfer function. Finally, an overall dynamic model of the plastication stage, combining melting and pumping, will be presented.

Polymer Flow Mass Balance

Polymer flow into the nozzle chamber is governed by the following mass balance equation [62].

$$Q_{mi} = A_m V_s + \frac{V_m}{\beta_m} \dot{P}_m + Q_{mo} \tag{A.22}$$

In the above equation, Q_{mi} is the flow of molten polymer into the nozzle of the screw and Q_{mo} is flow out of the nozzle, which is zero during plastication. The right hand side of (A.22) consists of two terms: the first is the change in volume in the nozzle, which is proportional to screw recovery rate, V_s, and the second is the change in volume due to compression of the polymer, which is proportional to the derivative of melt pressure, $\dot{P}_m$. In this equation, A_m is the effective cross-section of the melt, V_m is the melt volume, and β_m is polymer melt bulk modulus.

Flow into the nozzle, Q_{mi}, has two components: a drag flow component, Q_{md}, imparted to the polymer by the rotating action of the screw, and a pressure flow component, Q_{mp}, due to the pressure differential between the nozzle and the hopper.

$$Q_{mi} = Q_{md} + Q_{mp} \tag{A.23}$$

The drag flow component is the flow imparted to the molten polymer by the rotating action of the screw, driving it down the screw channel. The pressure flow component is the flow of the polymer, back towards the hopper, imparted by the pressure developed in the polymer in the nozzle. The exact solution to (A.23) involves non-Newtonian fluid flow analysis. The modeling goal, here, is to determine simple relations, to be used for feedback control. Equation (A.23) can be approximated by [13]

$$Q_{mi} = K_\omega \omega - K_m P_m \tag{A.24}$$

In this equation, K_ω and K_m are empirical gains related to screw speed and melt pressure.

Combining this with (A.22) yields

$$\frac{V_m}{\beta_m}\dot{P}_m = -A_m V_s - K_m P_m + K_\omega \omega \tag{A.25}$$

Screw Force Balance

During plastication, the screw retracts, creating the volume in the nozzle in which the molten polymer is deposited. The screw dynamics can be derived using a force balance, shown below.

$$A_H P_H - A_m P_m = -m\dot{V}_s - bV_s \tag{A.26}$$

The inputs to the system are the hydraulic pressure, P_H, and melt pressure, P_m, which apply a force through the system by acting through the hydraulic piston surface, A_H, and the nozzle piston surface, A_m. m is the approximate mass of the polymer and screw and b is the approximate viscous friction on the screw. The resultant force is countered by an inertial term, and a viscous friction term. This system can be linearized for a given operating point, and the resulting equation is

$$m\dot{V}_s = -bV_s + A_m P_m - A_H P_H \tag{A.27}$$

Hydraulic Fluid Mass Balance

The hydraulic pressure is supplied to the piston through the servovalve. These dynamics can be found using a mass balance of the hydraulic fluid around the piston chamber, similar to that done for polymer flow.

$$Q_{Hi} = \frac{V_H}{\beta_H}\dot{P}_H - A_H V_s + Q_{Ho} \tag{A.28}$$

In this case, flow into the hydraulic piston chamber, Q_{Hi} is zero. Q_{Ho} is the flow of hydraulic fluid out of the piston chamber. A_H is the effective area of the hydraulic piston, V_H is the hydraulic oil volume, β_H is the hydraulic oil bulk modulus, and P_H is the hydraulic pressure.

Fluid flow into or out of the piston chamber is a function of the servovalve spool position. A reasonable approximation [62] can be obtained by using an equation for

turbulent flow through an orifice:

$$Q_{Ho} = C_d X_v \sqrt{\frac{2}{\rho}(P_H - P_R)} \tag{A.29}$$

Flow through the orifice is a function of the orifice opening, X_v, the pressure differential across the orifice, $(P_H - P_R)$, the fluid mass density, ρ, and the valve discharge coefficient, C_d. The pressure differential will always be positive because hydraulic fluid will only be flowing out of the piston chamber during plastication. Equation (A.29) can be linearized around the valve operating point and the following simplified relation results:

$$Q_{Ho} = K_{Xp} X_v + K_{Pp} P_H \tag{A.30}$$

In this equation, K_{Xp} is the plastication valve position flow gain, and K_{Pp} is the plastication pressure flow gain. Strictly speaking, the pressure term in the above equation should be a function of the pressure differential, not just P_H, but since the return pressure, P_R is typically zero, the approximation is valid.

Equation (A.30) can be substituted into (A.28) to yield the linear differential equation:

$$\frac{V_H}{\beta_H}\dot{P}_H = A_H V_S - K_{Pp} P_H - K_{Xp} X_v \tag{A.31}$$

Modeling the Servovalve

The dynamics of the servovalve are dependent on the internal construction of the valve. On the Arburg machine, the valve was a three-port valve. A reasonable model of the valve dynamics is [61]

$$X_v[s] = \frac{K_v}{s(\tau_v s + 1)} V[s] \tag{A.32}$$

The input to the servovalve is a voltage, V, which drives the pilot stage. The valve transfer function has two poles. The integral term is a result of the fact that there is no internal feedback within the valve because of the solid spool [62]. The first order

term, with time constant τ_v, is an approximation of the rest of the valve dynamics, including the windings driving the pilot stage.

A.2.3 Overall Polymer Pumping Dynamics

The analysis above resulted in three first-order linear differential equations which describe the machine dynamics during plastication. Equations (A.25), (A.27), and (A.31) are essentially state equations. The system has three states: melt pressure, P_m, screw linear velocity, V_s, and hydraulic pressure, P_H. A block diagram of this system is shown in Figure [A.4]. In the block diagram, the input to the system is the servovalve opening, X_s. The screw rotational velocity, ω, appears as a disturbance input. The physical variables can be defined as state variables and the following matrix equation results:

$$\dot{Z}_p = \begin{bmatrix} -\frac{K_m\beta_m}{V_m} & -\frac{A_m\beta_m}{V_m} & 0 \\ \frac{A_m}{m} & -\frac{b}{m} & -\frac{A_H}{m} \\ 0 & \frac{A_H\beta_H}{V_H} & -\frac{K_{Pp}\beta_H}{V_H} \end{bmatrix} Z_p + \begin{bmatrix} \frac{K_\omega\beta_m}{V_m} & 0 \\ 0 & 0 \\ 0 & -\frac{K_{Xp}\beta_H}{V_H} \end{bmatrix} U_p \qquad \text{(A.33)}$$

In the above equation, the states are defined as $Z_{p1} = P_m$, $Z_{p2} = V_s$, and $Z_{p3} = P_H$. The control inputs are defined as $U_{p1} = \omega$ and $U_{p2} = X_v$. An output equation must be defined for this state-space model. If recovery rate is the output, then the output equation is:

$$Y_p = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} Z_p \qquad \text{(A.34)}$$

A transfer function relating recovery rate to servovalve spool position and screw rotational speed, ω, can be derived using (A.33) and (A.34), and is

$$V_s[s] = \frac{n_1\omega[s] - n_2X_v[s]}{d_3s^3 + d_2s^2 + d_1s + d_0} \qquad \text{(A.35)}$$

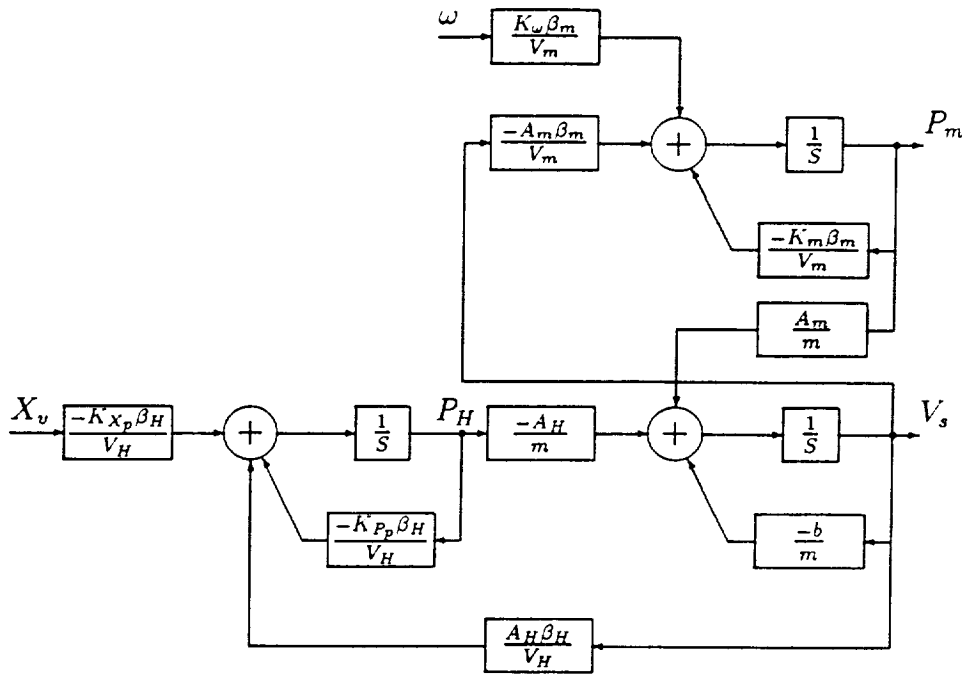

Figure A.4: Pumping Dynamics Block Diagram

The coefficients in the above equation are:

$$n_1 = A_m K_\omega \left(s \frac{\beta_H}{V_H} + K_{P_p} \right)$$

$$n_2 = A_H K_{X_p} \left(s \frac{\beta_m}{V_m} + K_m \right)$$

$$d_3 = m \frac{\beta_H}{V_H} \frac{\beta_m}{V_m}$$

$$d_2 = m \frac{\beta_H}{V_H} K_m + m K_{P_p} \frac{\beta_m}{V_m} + b \frac{\beta_H}{V_H} \frac{\beta_m}{V_m} \tag{A.36}$$

$$d_1 = m K_{P_p} K_m + b \frac{\beta_H}{V_H} K_m + b K_{P_p} \frac{\beta_m}{V_m} + A_m^2 \frac{\beta_H}{V_H} + A_H^2 \frac{\beta_m}{V_m}$$

$$d_0 = b K_{P_p} K_m + A_m^2 K_{P_p} + A_H^2 K_m$$

The servovalve dynamics can be included by substituting Equation (A.32) into Equation (A.35). The inputs to the system are now screw speed and servovalve voltage.

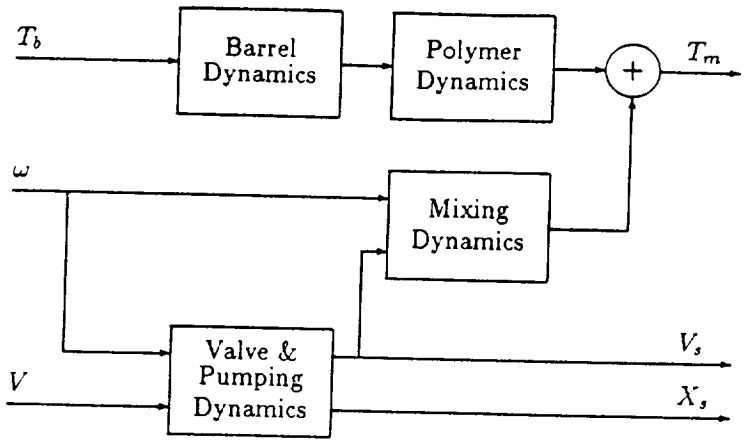

Figure A.5: Plastication Dynamics

A.2.4 Overall Plastication Dynamics

Earlier, it was stated that there are two interacting dynamic systems which describe the plastication dynamics. After the individual dynamics have been described, it is now possible to understand these interactions. Figure [A.5] is a simplified block diagram showing the interaction between the thermodynamic system and the fluid dynamic system. The goal of plastication is to generate the *required shot size*, X_s, at the *correct temperature*, T_m, in the *fastest possible time*, t_r (to maximize productivity while maintaining part quality). As can be seen from Figure [A.5], the interaction takes place through the recovery rate, V_s. It will be up to the quality control system to determine the correct screw RPM, recovery rate profile, and barrel temperature to meet the objectives.

A.3 Injection

Injection is the phase during which the molten polymer is injected into the mold cavity. Initially, the molten polymer is in the nozzle, and the screw acts like a piston, forcing the polymer through the nozzle into the mold. Injection continues until the mold is completely filled, and is terminated based either on screw position

or time.

The microscopic modeling of injection has centered around the modeling of the unsteady flow of a hot, non-Newtonian polymer melt into a cold cavity. These dynamics are modeled as unsteady-state free surface flow coupled with transient cooling and are described by the basic equations of change. Tadmor, Boyer, and Gutfinger, in [16], modeled the flow of polymer into the melt cavity using the finite element method. Experimental work by Kreuger and Tadmor [17] validated this modeling method by studying the injection of polymer into a rectangular cavity with various obstructions. One result was that it could be assumed that molten polymer behaved as a Newtonian fluid, which simplified modeling and simulation. Later work by Mavridis, Hyrmak, and Vlachopoulos [18], Kamal, *et al.* [19], and Gogos, Huang, and Schmidt [20] included fountain flow in the modeling. Understanding of the injection phase is important because shear and elongational stresses cause high orientation of the polymer. Polymer which contacts the mold surface has frozen-in orientation while the remainder of the polymer has time to relax while cooling. These microstructure differences, within a part, due to the injection process can have a great impact on quality properties such as strength and warpage. Empirical modeling of the injection phase has focused on control of injection velocity or pressure. In a sense, the filling of the mold can be considered "open-loop" once the plastic leaves the nozzle. Shankar, and Shankar and Paul in [21, 22] developed a lumped-parameter model for the injection process and evaluated a state-space approach to injection control. Costin, Okonski, and Ulicny in [23] examined adaptive control of the injection process and compared it to the performance of a PI controller for hydraulic pressure control. Haber and Kamal [24] and Kamal *et al.* [25] also studied the control of pressure during injection.

For implementation of screw velocity melt pressure control, injection dynamics should include the following:

1. a mass balance model of molten polymer flow, relating melt pressure and polymer flow into the mold;

2. a force balance model of the screw, relating polymer deposition rate to the melt pressure and hydraulic pressure;

3. a mass balance model of hydraulic fluid flow, relating screw speed and hydraulic pressure to servovalve spool position;

The only difference between injection and plastication dynamics is that now polymer flow out of the nozzle through an orifice must be modeled instead of polymer flow into the nozzle.

A.3.1 Injection Polymer Flow Dynamics

Polymer flow during injection can be described as a fluid flow mass balance (similar to plastication) [62].

$$Q_{mi} = A_m V_s + \frac{V_m}{\beta_m}\dot{P}_m + Q_{mo} \tag{A.37}$$

During injection, flow into the nozzle, Q_{mi}, is zero. Flow out of the nozzle, Q_{mo}, is described by a laminar flow equation [21].

$$Q_{mo} = RA_n P_m \tag{A.38}$$

In the above equation, A_n is the nozzle cross-sectional area, and R is the effective resistance to flow of the polymer through the nozzle. This equation is also a simplification of the flow behavior of the polymer. It should be noted that R will change with time. There are two reasons for this: as the polymer enters different parts of the mold, the area available for flow will change, and as the polymer freezes to the sides of the mold, the area available for flow will decrease. Because of this resistance, injection dynamics are also a function of the mold design.

Equations (A.37) and (A.38) can be combined to form the following state equation:

$$\frac{V_m}{\beta_m}\dot{P}_m = -RA_nP_m - A_mV_s \tag{A.39}$$

Screw Force Balance

During injection, the screw dynamics are the same as those during plastication. The equation will be re-stated for completeness.

$$m\dot{V}_s = -bV_s + A_mP_m - A_HP_H \tag{A.40}$$

Hydraulic Fluid Mass Balance

The hydraulic fluid dynamics are the same for injection as for plastication, except that fluid is now flowing into the piston chamber, and the system is being driven by the supply pressure. The mass balance equation is

$$Q_{Hi} = \frac{V_H}{\beta_H}\dot{P}_H - A_HV_s + Q_{Ho} \tag{A.41}$$

and the orifice flow equation is

$$Q_{Hi} = C_dX_v\sqrt{\frac{2}{\rho}(P_S - P_H)} \tag{A.42}$$

Notice that the only difference between Equations (A.29) and (A.42) is how the pressure differential term is generated. During injection, the flow is driven by the supply pressure, as opposed to plastication where pressure is developed due to screw rotation. Again, the flow equation can be linearized to yield one in which hydraulic flow is proportional to spool position and hydraulic pressure,

$$Q_{Hi} = K_{Xi}X_v + K_{Pi}P_H \tag{A.43}$$

K_{Xi} and K_{Pi} are spool position and hydraulic pressure empirical gains.

Combining (A.41) and (A.43) yields the following state equation:

$$\frac{V_H}{\beta_H}\dot{P}_H = A_HV_s + K_{Xi}X_v + K_{Pi}P_H \tag{A.44}$$

A.3.2 Overall Injection Dynamic Model

As for plastication, a state space model for injection can be formulated using Equations (A.39), (A.40), and (A.44). The matrix equation is:

$$\dot{Z}_i = \begin{bmatrix} -\frac{RA_n\beta_m}{V_m} & -\frac{A_m\beta_m}{V_m} & 0 \\ \frac{A_m}{m} & -\frac{b}{m} & -\frac{A_H}{m} \\ 0 & \frac{A_H\beta_H}{V_H} & \frac{K_{P_i}\beta_H}{V_H} \end{bmatrix} Z_i + \begin{bmatrix} 0 \\ 0 \\ \frac{K_{X_i}\beta_H}{V_H} \end{bmatrix} U_i \qquad \text{(A.45)}$$

In the above equation, the injection state variable, Z_i, is defined using the same physical variables as Z_p. Notice that this system only has one input, namely servovalve spool position. Again, if the desired output is recovery rate, then

$$Y_i = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} Z_i \qquad \text{(A.46)}$$

A transfer function can be derived from the above state and output equations. The system input is servovalve spool position and the output is recovery rate:

$$V_s[s] = \frac{A_H K_{Xi}\left(s\frac{\beta_m}{V_m} + RA_n\right) X_v[s]}{d_3 s^3 + d_2 s^2 + d_1 s + d_0} \qquad \text{(A.47)}$$

The denominator coefficients in the above equation are:

$$\begin{aligned} d_3 &= m\frac{\beta_H}{V_H}\frac{\beta_m}{V_m} \\ d_2 &= -m\frac{\beta_m}{V_m}K_{P_i} + mRA_n\frac{\beta_H}{V_H} + b\frac{\beta_m}{V_m}\frac{\beta_H}{V_H} \\ d_1 &= -mRA_nK_{P_i} - b\frac{\beta_m}{V_m}K_{P_i} + bRA_n\frac{\beta_H}{V_H} + A_m^2\frac{\beta_H}{V_H} + A_H^2\frac{\beta_m}{V_m} \\ d_0 &= -bRA_nK_{P_i} + A_m^2K_{P_i} + A_H^2RA_n \end{aligned} \qquad \text{(A.48)}$$

The servovalve dynamics can be included by substituting Equation (A.32) into Equation (A.47). The input to the system is servovalve voltage. A block-diagram for this system is shown in Figure [A.6].

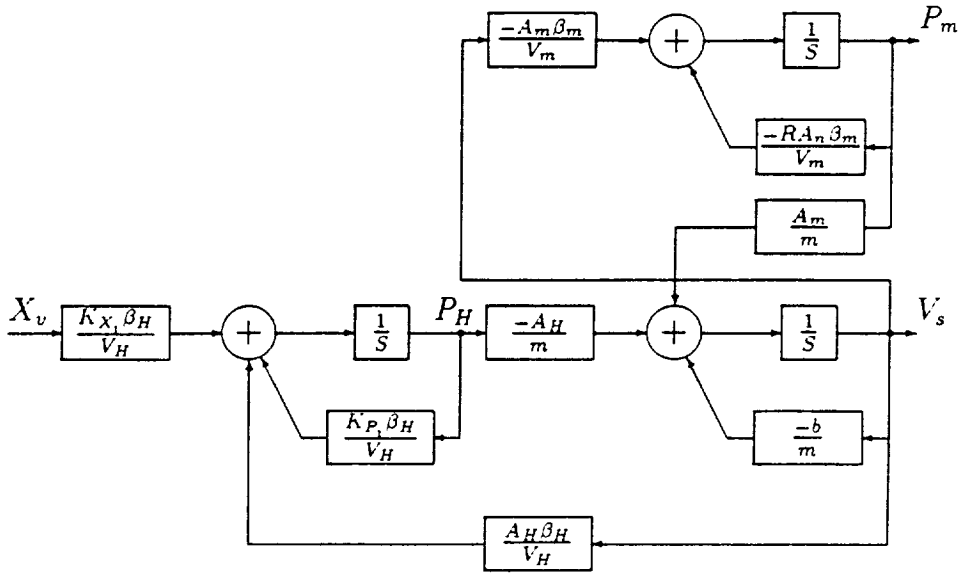

Figure A.6: Injection Dynamics Block Diagram

A.4 Holding

Holding occurs after injection. At this point, the mold is full and cooling begins. The gate is not frozen off completely, so polymer can still flow in or out. The goal of holding is to ensure that no voids or porosity form due to shrinkage resulting from the initial cooling and that part density is consistent. Holding continues until the gate is frozen shut. Holding dynamics are identical to injection dynamics, except for the polymer mass balance. This will be discussed in the following section.

The main phenomenon governing the holding phase is an equation of state [7, 8]:

$$(P_m + \alpha)(\nu + \beta) = \rho T_m \tag{A.49}$$

This is a modified van der Waal's equation of state which relates melt pressure and specific volume to melt temperature. In Equation (A.49), P_m and T_m are the melt pressure and temperature respectively, ν is the specific volume (the inverse of density), and α, β, and ρ are empirically determined constants.

Agrawal, *et al.* [31] discuss the use of this equation for control. One example is the control of part density. If it is desired to maintain constant density, (A.49) becomes

$$P_m = xT_m - y \tag{A.50}$$

This relationship can be used to determine the input trajectory to a holding phase pressure controller.

Since the mold is full at this point, the start of holding is effectively the start of cooling. Physically, any pressure control on the polymer in the mold terminates when the gate freezes shut, since there is no longer any way to maintain pressure on the molded part. Therefore, there are three ways to terminate the holding phase: pressure in the mold is monitored or melt temperature is monitored and some reference value is used as a termination threshold, or assumptions are made about the heat dissipation of the mold and holding is terminated by time. In this work, holding will be terminated by time. The length of this time will be dependent upon the melt temperature at the beginning of holding, but a constant value was used throughout.

A.4.1 Holding Polymer Mass Balance

Polymer flow during holding can be described by the following fluid flow mass balance:

$$Q_{mi} = \frac{V_m}{\beta_m}\dot{P}_m + Q_{mo} \tag{A.51}$$

For holding, the control volume for the mass balance is now the mold. There is no volume change term in this equation, as there was in plastication and injection. In this case, there is no flow out of the mold, so Q_{mo} is zero. There is a flow into the mold, corresponding to the shrinkage of the cooling part. This flow is equal to the polymer which flows out of the nozzle into the mold. This is proportional to the

velocity of the screw.

$$Q_{mi} = -A_m V_s \tag{A.52}$$

The negative sign in the above equation is a consequence of the fact that screw motion towards the nozzle is considered to be in the negative direction so as to conform with the previously derived dynamic equations. Now, substituting (A.52) into (A.51) yields

$$\frac{V_m}{\beta_m}\dot{P}_m = -A_m V_s \tag{A.53}$$

A.4.2 Overall Holding Dynamic Model

The form of the state equations for hydraulic pressure and screw velocity are identical to those of plastication. The only difference will be in the coefficients of the linearization. For completeness, these state equations are:

$$m\dot{V}_s = -bV_s + A_m P_m - A_H P_H \tag{A.54}$$

and

$$\frac{V_H}{\beta_H}\dot{P}_H = A_H V_s + K_{Xh} X_v + K_{Ph} P_H \tag{A.55}$$

K_{Xh} and K_{Ph} are spool position and hydraulic pressure empirical gains.

Again, defining the holding state vector, Z_h, analogous to the plastication and injection state vector, the following matrix equation can be formed:

$$\dot{Z}_h = \begin{bmatrix} 0 & -\frac{A_m \beta_m}{V_m} & 0 \\ \frac{A_m}{m} & -\frac{b}{m} & -\frac{A_H}{m} \\ 0 & \frac{A_H \beta_H}{V_H} & \frac{K_{Ph}\beta_H}{V_H} \end{bmatrix} Z_h + \begin{bmatrix} 0 \\ 0 \\ \frac{K_{Xh}\beta_H}{V_H} \end{bmatrix} U_h \tag{A.56}$$

If melt pressure is the desired output, then

$$Y_h = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} Z_h \tag{A.57}$$

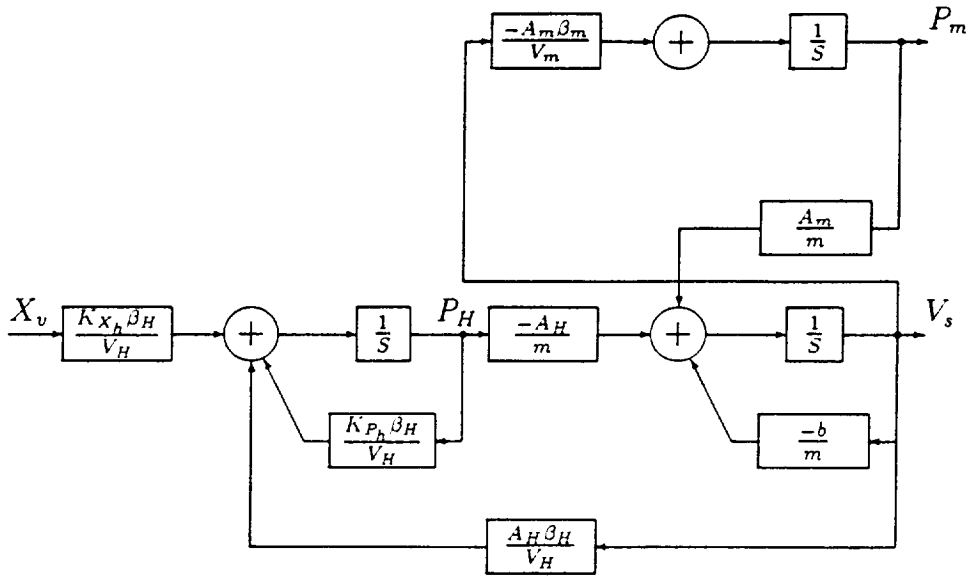

Figure A.7: Holding Dynamics

The melt pressure transfer function, for holding, is

$$P_m[s] = \frac{K_{Xh}X_v}{d_3s^3 + d_2s^2 + d_1s + d_0} \tag{A.58}$$

The denominator coefficients in the above equation are:

$$\begin{aligned} d_3 &= m\frac{\beta_m}{V_m}\frac{\beta_H}{V_H} \\ d_2 &= b\frac{\beta_m}{V_m}\frac{\beta_H}{V_H} - mK_{Ph}\frac{\beta_m}{V_m} \\ d_1 &= A_H^2\frac{\beta_m}{V_m} + A_m^2\frac{\beta_H}{V_H} - K_{Ph}b\frac{\beta_m}{V_m} \\ d_0 &= -A_m^2K_{Ph} \end{aligned} \tag{A.59}$$

The servovalve dynamics can be included by using Equation (A.32), as in plastication and injection. A block-diagram for holding is shown in Figure [A.7].

A.5 Cooling

During cooling, the part rests in the mold until it is cool enough to be ejected and still retain its shape. The cooling dynamics are determined by the ability of

the mold to dissipate the heat of the injected plastic. In general, this is a complex transient heat conduction problem, akin to that described for the polymer during plastication. In this case, the mold will typically have cooling water running through it, removing heat. For control purposes the mold dynamics can be assumed to behave as a first order system with a time delay, as in plastication [21]. The following empirical model for cooling can be used

$$T_{m_M}[s] = \frac{K_M}{\tau_M s + 1} T_W[s] \tag{A.60}$$

Average melt temperature in the mold, T_{m_M}, is a function of the water temperature, T_W, and the heat dissipation time constant, τ_M. K_M is an empirical gain.

In the mold used for this study, there was a constant flow of cooling water through the mold, and there was no instrumentation for collecting mold temperature. For all experiments, cooling was terminated by a time, which was determined empirically. Again, the length of this time is dependent on the temperature of the polymer in the mold but a constant value was used. This is typical of the methods used in industry.

A.6 Process Control and Phase Interaction

In the previous section, the dynamics of each phase have been examined independently. The result was an approximate transfer function model for each phase. In each case, the model parameters are dependent upon the specific machine, the operating conditions, and the particular polymer used, so some experimentation and identification would have to be done to get an accurate model. Once obtained, this model would be suitable for use with traditional control system design techniques.

Control design for a specific phase would consist of determining desirable controller performance, making assumptions about the inputs and disturbances that may be expected, *etc.* The result will be a controller which meets the design goals within that phase, independent of the other phases. This is *intra-phase control.*

All phases of the injection molding cycle affect part quality. This is due to the interaction among each of the phases of the injection molding cycle. This interaction is a product of the fact that the output of a phase establishes the operating conditions for subsequent phases and machine cycles.

The phase inputs and outputs, and interactions are shown in Figure [1.3]. Two different types of inputs are defined, external and internal. External inputs are those which the operator has direct control over. Typically, these are machine settings which are used to control the process directly, or are inputs to controllers on the process. Examples of external inputs are screw speed and the barrel temperature setpoints. In Figure [1.3], these inputs are shown entering from the left side. Internal inputs are those inputs which cannot be controlled directly. These can be inputs which are outputs from a previous phase or are disturbance inputs. Examples of internal inputs are polymer melt temperature and raw material variations. In Figure [1.3], these inputs are shown entering from the top.

To better understand this interaction, the effect of melt temperature will be traced through the cycle. During plastication, the melt is generated at a temperature which is a function of the barrel temperature, screw RPM, and the recovery rate. The result is a melt with an axial temperature profile which varies with the distance down the barrel from the nozzle. (This is discussed in more detail when the experimental results of Raimund are considered, in Chapter 2.) Following plastication comes injection. The major physical effect of melt temperature is on the viscosity of the polymer melt. Both the viscous friction term, b (Equation (A.40)), and the effective mold resistance, R (Equation (A.39)), are functions of viscosity. When holding commences, pressure is applied to the polymer in the mold until the gate freezes off. During this time, the polymer cools and shrinks. The amount of shrinkage that occurs, and therefore the amount of polymer which must be introduced to ensure complete mold filling (Equation (A.49)) is proportional to the

temperature drop which occurs through the holding cycle. Finally, the part cannot be ejected from the mold until it has cooled below the point at which it would warp. This time would depend upon the average melt temperature in the mold at the start of cooling and the heat dissipation time constant (Equation (A.60)).

So far, the effect on subsequent cycles has not been discussed. Consider that the total volume in the screw flights can be several times the volume of one shot. Therefore, the melt temperature profile at the n^{th} cycle may be dependent upon the temperature history of the k previous cycles. Sanschagrin analyzed this cyclical dependence with several different inputs and outputs in [30]. This interaction signifies the need for *inter-phase control* and *inter-cycle control* [31].

Inter-phase and inter-cycle control is what an experienced operator does. He starts with an initial set of control inputs;the operating policy. He will adjust these inputs, as the process evolves, based on his monitoring of process outputs. The size and type of adjustments to make are determined based on experience with the process. The operator has developed an intuition about the interactions and makes adjustments based on this knowledge. This control might be viewed as a combination of *automatic control of the process* and *operator control of quality.*

A.7 Summary of Previous Research

In this appendix, the dynamics were described in detail. The most comprehensive work on *plastication* was done by Donovan, *et al.* in [9, 10, 11]. The emphasis of this work was on melting due to conduction heating. A theoretical model of conduction transients was developed, and then verified using cooling experiments. Donovan applied earlier work in plastication extruders to understand melting dynamics due to viscous dissipation [71]. This work was extended by Lipshitz to the injection molding process [12]. Raimund used these results as the foundation for his experimental work [13]. Kamal, Patterson, and Gomes described the dynamics

and control of melt temperature in [14, 15]. This work was different from the previous work in that their concern was on the implementation and effect of feedback controllers rather than a study of the process.

The remaining phases of the injection molding process were studied as a lumped-parameter system. This analysis followed the development of Shankar in [21, 22]. This was used to formulate the dynamic models of injection and holding, as well as the melt accumulation dynamics during plastication.

Ma presented one of the first analyses of the injection molding process as a system [3]. The fundamental result is that no portion of the cycle exists in isolation. This was further emphasized by the experimental work of Sanschagrin in [30], and the discussion of the importance of cycle-to-cycle control in the survey of injection molding machine control done by Agrawal *et al.* in [31].

A.8 Summary

In this appendix, a detailed analysis of the plastic injection molding process was carried out. This process consisted of several different dynamic processes that occur sequentially. These are the phases of the injection molding process. Because each of these phases establishes the operating conditions for subsequent phases, this interaction must be understood before designing a controller for one phase and for the whole process. This control is essential for maintaining product consistency.

APPENDIX B

Symbol Table and Nomenclature for Machine Dynamics

Below is a list of symbols used in formulas throughout the dynamics appendix. It is arranged according to the sections in which the particular symbols appear. A tradeoff has been made for clarity over duplicity. Also, due to the number of equations, some symbols have been used to denote several different quantities. In general, the meaning of the symbol should be evident from the context.

B.1 Definition of the Quality Control Problem

$u(\cdot)$	command input
$k(\cdot)$	controller parameter
$\mathcal{U}$	set of feasible command signals
$\mathcal{K}$	set of feasible controller parameters
$t_c(l)$	length of the l^{th} cycle
q_V	quality variable
$\mathcal{Q}_V$	quality variable space
$\mathcal{Q}_{V_T}$	quality variable target set
q_A	quality attribute
$\mathcal{Q}_A$	quality attribute space
$\mathcal{Q}_{A_T}$	quality attribute target set
$y(\cdot)$	process output
$\mathcal{Y}$	set of measurable outputs
$H(\cdot)$	process model
$G_V(\cdot)$	quality variable model
$G_A(\cdot)$	quality attribute model

B.2 A Review of the Multiobjective Optimization Problem

$f(\cdot)$	vector objective function
z	objective function value
Z	objective space
x	optimization variable
X	optimization feasible set
$g(\cdot)$	constriant function
$\mathcal{N}$	nondominated set
U	utility function
u	Langrange constraint multiplier
w	Lagrange objective function multiplier
d	feasible direction

B.3 A Linear Programming Solution to Multiobjective Quality Control

P_o	feasible direction projection value
α	objective gradient normalizing constant
$f(\cdot)$	objective function
d	feasible direction
β	constraint gradient normalizing constant
j	active constraint index
$\tilde{d}$	approximate direction
$\mathcal{N}$	nondominated set
$\hat{x}$	feasible operating point
δ	feasible direction magnitude threshold
ϵ	optimization stopping threshold
η	line search step size
$\tilde{x}$	operating condition generated by the line search
$\tilde{q}$	process output during line search

B.4 Polymer Melting Due to Conduction Heating

erf()	error function
erfc()	error function complement
T_b	barrel temperature
T_f	melt film temperature profile
T_m	melt film/solid bed interface temperature
T_r	reference temperature
T_s	solid bed temperature profile
t	time
y	melt film coordinate direction
α_m	melt film thermal diffusivity
α_s	solid bed thermal diffusivity
δ	melt film thickness

B.5 Polymer Melting Due to Viscous Dissipation

Br	Brinkman number
T_b	barrel temperature
T_f	melt film temperature profile
T_s	solid bed temperature (constant)
V_s	barrel velocity
$V_x(y)$	velocity profile due to force in the x-direction
y	melt film coordinate direction
α_m	melt film thermal diffusivity
δ	melt film thickness
$\overline{\delta}$	average melt film thickness
μ	Newtonian viscosity
τ_{yx}	shear stress in the x-dir due to force in y-dir

B.6 Overall Melting Dynamics

d	conduction heating time delay
K_c	conduction heating gain
K_v	viscous dissipation heating gain due to V_s
K_{vs}	viscous dissipation heating gain due to ω
$K_{v\omega}$	viscous dissipation heating gain
T_{mc}	melt temperature due to conduction
T_{mv}	melt temperature due to viscous dissipation
T_b	barrel temperature
t_r	recovery time
V_y	shear velocity
V_s	screw recovery rate
X_s	shot size (screw position)
$\overline{V_s}$	average screw recovery rate
τ_b	barrel heating time constant
τ_m	melt film heating time constant
τ_{vd}	viscous dissipation heating time constant
ω	screw rotational velocity

B.7 Polymer Fluid Dynamics

A_H	hydraulic pressure effective area
A_m	melt pressure effective area
b	approximate viscous friction on screw
C_d	orifice flow discharge coefficient
K_m	melt pressure empirical gain
K_{Xp}	servovalve spool position flow gain
K_{Pp}	servovalve hydraulic pressure flow gain
K_v	servovalve gain (voltage to spool position)
K_ω	screw speed empirical gain
m	approximate mass of screw and polymer system
P_H	hydraulic pressure
P_m	polymer melt pressure
P_R	hydraulic return pressure
P_S	hydraulic supply pressure
Q_{Hi}	hydraulic flow into piston chamber
Q_{Ho}	hydraulic flow out of piston chamber
Q_{mi}	polymer melt flow into nozzle
Q_{mo}	polymer melt flow out of nozzle
Q_{md}	polymer melt drag flow component
Q_{mp}	polymer melt pressure flow component
U_p	plastication input
V_H	hydraulic oil volume
V_m	melt volume
V_s	recovery rate (screw linear velocity)
V	servovalve input voltage
X_v	servovalve spool position
Y_p	plastication output
Z_p	plastication state vector
β_m	melt bulk modulus
ω	screw RPM (screw rotational velocity)
β_H	hydraulic oil bulk modulus
ρ	fluid mass density
τ_v	servovalve time constant

B.8 Injection Dynamics

A_H	hydraulic pressure effective area
A_m	melt pressure effective area
A_n	nozzle cross-sectional area
b	approximate viscous friction on screw
C_d	orifice flow discharge coefficient
K_{Xi}	servovalve spool position flow gain
K_{Pi}	servovalve hydraulic pressure flow gain
K_v	servovalve gain (voltage to spool position)
m	approximate mass of screw and polymer system
P_H	hydraulic pressure
P_m	polymer melt pressure
P_R	hydraulic return pressure
P_S	hydraulic supply pressure
Q_{Hi}	hydraulic flow into piston chamber
Q_{Ho}	hydraulic flow out of piston chamber
Q_{mi}	polymer melt flow into nozzle
Q_{mo}	polymer melt flow out of nozzle
R	effective resistance to flow in the mold
U_i	injection input
V_H	hydraulic oil volume
V_m	melt volume
V_s	recovery rate (screw linear velocity)
V	servovalve input voltage
X_v	servovalve spool position
Y_i	injection output
Z_i	injection state vector
β_m	melt bulk modulus
β_H	hydraulic oil bulk modulus
ρ	fluid mass density
τ_v	servovalve time constant

B.9 Holding Dynamics

A_H	hydraulic pressure effective area
A_m	melt pressure effective area
A_n	nozzle cross-sectional area
b	approximate viscous friction on screw
C_d	orifice flow discharge coefficient
K_{Xh}	servovalve spool position flow gain
K_{Ph}	servovalve hydraulic pressure flow gain
m	approximate mass of screw and polymer system
P_H	hydraulic pressure
P_m	polymer melt pressure
Q_{Hi}	hydraulic flow into piston chamber
Q_{Ho}	hydraulic flow out of piston chamber
U_h	holding input
V_H	hydraulic oil volume
V_m	melt volume
V_s	recovery rate (screw linear velocity)
V	servovalve input voltage
X_v	servovalve spool position
Y_h	holding output
Z_h	holding state vector
β_m	melt bulk modulus
β_H	hydraulic oil bulk modulus
ρ	fluid mass density
τ_v	servovalve time constant

B.10 Cooling Dynamics

K_M	heat dissipation empirical gain
T_{m_M}	average melt temperature in the mold
T_W	cooling water temperature
τ_M	heat dissipation time constant

APPENDIX C
An Example of Approximation Programming

In this thesis, a nonlinear program (NLP) was defined, the solution of which yielded a feasible direction to the tradeoff surface. The solution to this NLP was obtained by approximating the nonlinear constraints with linear constraints, forming a linear program (LP) and iteratively solving this LP. This technique is known as *approximation programming* [53]. In this appendix, this process will be illustrated with a simple example.

C.1 Example

The following nonlinear program is used to determine the descent direction:

$$\max_{d} P_o \tag{C.1}$$

subject to

$$P_o + \alpha_i(\tilde{x})\nabla f_i(\tilde{x}) \cdot d \leq 0 \qquad i = 1, \cdots, p \tag{C.2}$$

$$-\beta_j(\tilde{x})\nabla g_j(\tilde{x}) \cdot d \leq 0 \qquad j \in [1, m] \ni g_j(\tilde{x}) = 0 \tag{C.3}$$

$$P_o \geq 0 \tag{C.4}$$

$$\sum_{k=1}^{n} d_k^2 = 1 \tag{C.5}$$

where α_i and β_i are chosen to satisfy the following:

$$\alpha_i(\tilde{x})^2 \nabla f_i(\tilde{x}) \cdot \nabla f_i(\tilde{x}) = 1 \qquad i = 1, \cdots, p \tag{C.6}$$

and

$$\beta_j(\tilde{x})^2 \nabla g_j(\tilde{x}) \cdot \nabla g_j(\tilde{x}) = 1 \qquad j \in [1, m] \ni g_j(\tilde{x}) = 0 \tag{C.7}$$

The nonlinear constraint of Equation (C.5) is approximated by the following three linear constraints:

$$-1 \leq d_k \leq 1 \qquad k = 1, \cdots, n \tag{C.8}$$

$$\sum_{k=1}^{n} |d_k| \leq \sqrt{n} \tag{C.9}$$

$$(1 + \sum_{k=1}^{n} \tilde{d}_k^2) = \sum_{k=1}^{n} 2\tilde{d}_k d_k \tag{C.10}$$

These linear constraints represent a faceted approximation of the hypersphere constraint of Equation (C.5). One point that should be emphasized is that the only constraint which is *necessary* is Equation (C.8), which ensures that the solution is *bounded* and is a *feasible direction.*

Assume that the gradient vectors have been obtained for some operating point, $\tilde{x}$. After being normalized, they are:

$$\nabla f_1(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = \begin{bmatrix} -0.8 & 0.56 & -0.23 \end{bmatrix} \tag{C.11}$$

and

$$\nabla f_2(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = \begin{bmatrix} 0.95 & 0.2 & 0.22 \end{bmatrix} \tag{C.12}$$

A physical interpretation of these gradients would be that to decrease f_1 one must increase input $\tilde{x}_1$, decrease input $\tilde{x}_2$, and increase input $\tilde{x}_3$. To decrease f_2 one must decrease all three inputs. If the elements of f_1 and f_2 are interpreted as the sensitivity of the objective to each of the inputs, one would conclude that any changes in $\tilde{x}_3$ would produce completely conflicting changes in the both objectives, because the relative weights are equal; there is some conflict in the objectives with respect to $\tilde{x}_1$; and there is concurrence with respect to $\tilde{x}_2$. The solution to the original NLP is the following direction

$$d^* = \begin{bmatrix} -0.21 & -0.98 & 0.02 \end{bmatrix} \tag{C.13}$$

This result suggests that $\tilde{x}_1$ should be changed "a little bit", $\tilde{x}_2$ should be changed "a lot", and $\tilde{x}_3$ should be "left alone". This supports the intuition discussed above.

If the approximating LP is solved, as is specified by the multiobjective algorithm, the problem is first solved without using Equation (C.10). This yielded the following direction:

$$d^1 = \begin{bmatrix} -0.31 & -1.0 & 0.42 \end{bmatrix} \qquad \|d^1\| = 1.27 \tag{C.14}$$

This direction is feasible, but there is a problem. The solution to a linear program is always a *vertex* of the *simplex* formed by the constraints. In this case, because of the poor approximation of the nonlinear constraint by the linear equations, there is no vertex "along" the correct direction. For the second iteration, $\tilde{d}$ was set to d^1, in Equation (C.10), and the new LP was solved. This yielded the following direction:

$$d^2 = \begin{bmatrix} -0.24 & -1.0 & 0.14 \end{bmatrix} \qquad \|d^2\| = 1.08 \tag{C.15}$$

The solution is beginning to converge to d^*. After the third iteration,

$$d^3 = \begin{bmatrix} -0.19 & -1.0 & 0.05 \end{bmatrix} \qquad \|d^3\| = 1.04 \tag{C.16}$$

This solution is in good agreement with d^*.

If one more iteration of the algorithm is run, the resulting direction is

$$d^4 = \begin{bmatrix} -0.31 & -0.98 & 0.44 \end{bmatrix} \qquad \|d^4\| = 1.25 \tag{C.17}$$

Notice that this answer is almost exactly the same as d^1. This is another consequence of the fact that the solution to an LP is a vertex of the simplex. As the LP result begins to approach d^*, the approximating constraint of Equation (3.39) becomes tangent to the hypersphere described by Equation (3.34) near d^*. Vertices created by this constraint are positioned *away* from d^*. This problem of convergence is inherent to approximation programming, as is noted by Avriel in [53]. Since any direction generated by the multiobjective algorithm is feasible, if divergence of the approximation program is detected, the last "best" feasible direction is used.