2003

The design and evaluation of a PLC-based model predictive controller for application in industrial food processes

Bryan T. Griffen
University of Northern Iowa

THE DESIGN AND EVALUATION OF A PLC-BASED MODEL PREDICTIVE CONTROLLER FOR APPLICATION IN INDUSTRIAL FOOD PROCESSES

A Dissertation

Submitted

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Industrial Technology

Approved:

Dr. Ali E. Kashef, Chair

Dr. Recayi Pecen, Co-Chair

Dr. Teresa J. K. Hall, Committee Member

Eugene Wallingford

Dr. Eugene Wallingford, Committee Member

Julia Wallace

Dr. Julia Wallace, Committee Member

Bryan T. Griffen

University of Northern Iowa

December 2003

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ACKNOWLEDGEMENTS

In reflecting back over the several years of effort that went in to this degree, I want to take a moment to thank those who have given of their time and effort to help make this dissertation something of which I can be proud. I would like to express my deepest appreciation to the following people.

First and foremost I would like to recognize my wife, Hillary, for her unwavering support and encouragement that kept me moving forward through this long journey. I am grateful to her for the many personal sacrifices that she made so that I could attend classes and work on my research.

I would like to thank my father, Dr. Dana Griffen, for his example and guidance, without which I would never have embarked on this journey.

I owe a special thanks to my brother, Blaine Griffen. His guidance in statistics was key to analyzing the data and interpreting the experimental results.

To Karel Stryczek I am grateful for the tutelage and corroboration in developing the background and justification for this research.

In closing, I want to express thanks to my Graduate Advisory Committee for all of their support and patience as I completed this journey.

TABLE OF CONTENTS

PAGE

PAGE

PAGE

PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

LIST OF EQUATIONS

THE DESIGN AND EVALUATION OF A PLC-BASED MODEL PREDICTIVE CONTROLLER FOR APPLICATION IN INDUSTRIAL FOOD PROCESSES

An Abstract of a Dissertation

Submitted

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Industrial Technology

Approved:

____
Dr. Ali E. Kashef, Committee Chair

John W. Somervill
Dr. John W. Somervill
Dean of the Graduate College

Bryan T. Griffen

University of Northern Iowa

December 2003

ABSTRACT

Model Predictive Control (MPC) is a viable control strategy for industrial processes that display relatively large variations in the process variable, have complex process variable interactions, or display a large amount of process deadtime. The objective of using MPC in manufacturing is to reduce overall process variability, the result being an increase in process accuracy, precision and efficiency. This study focused on the implementation of model predictive control techniques on an industrial sugar cooking process. The goal was to implement a successful MPC solution directly on a programmable logic controller (PLC) rather than on a personal computer (PC). Although there are many commercially available MPC controllers for implementation on a stand-alone PC, to date there are no control packages for realizing model-based control techniques directly on the ubiquitous PLC.

This study implemented and evaluated three PC-based, commercial MPC technologies for the sugar cooking process, and a new model state feedback (MSF) MPC implementation directly on Rockwell Automation's Allen-Bradley ControlLogix® PLC. A standard proportional-integral-derivative (PID) control implementation was used as a baseline for comparing the MPC strategies. There were three main areas on which the overall comparative analysis focused. These comparison areas were the dynamic response of each strategy at startup, including both temperature rise time and overshoot, and the steady-state disturbance rejection capabilities of each strategy.

The test results showed that the MPC strategies controlled the sugar cooking process better than the traditional PID control method in regards to temperature rise time,

temperature overshoot, and disturbance rejection based on feed rate disturbances. It was seen that the differences between the various MPC strategies was not significant relative to temperature overshoot and disturbance rejection. The PLC-based MPC strategy was comparable, but not superior, to the PC-based commercial MPC applications. However, this strategy has several benefits such as requiring no external hardware, software, and communications protocols, which may result in a less expensive implementation than the commercial MPC strategies. The PLC-based strategy is also easier and cheaper to maintain because it is developed on the existing, well-known control platform with existing tools.

CHAPTER 1

INTRODUCTION

Model Predictive Control (MPC), a branch of Advance Process Control (APC), has been identified as a viable control strategy for production processes that display relatively large variations in system controlled output values in comparison to the system's control set point, for processes with appreciable process variable interactions, and for systems that display a large amount of process deadtime and/or system disturbances (Willis & Tham, 1994). The objective of using APC in manufacturing is to reduce overall system controlled output variability regardless of measured and unmeasured system disturbances such as product raw material changes, variation in production feed rates, wear on process equipment, or random environmental fluctuations. With sustained system output value repeatability, the control set point can be confidently moved closer to an actual process limit. The benefit being a reduction in quality variability and lost product, which can be caused by operating a process at a conservative control set point due to process variability. Additionally, as the set point is moved closer to the system specification control limit, the result is an increase in process accuracy, precision and efficiency.

Confectionery processing requires precise temperature and moisture control of a high-boil, sugar-based formulation product. The system must remain within process specifications even with the presence of various measured and unmeasured disturbances (weather changes, raw material changes, system steam pressure, and so forth). Capability to rapidly increase the sugar-based product's initial temperature to a final temperature with an overall gain of 55°F in 15 minutes or less, with minimal temperature overshoot, is also highly desirable. Standard programmable logic controller (PLC) based proportional-integral-derivative (PID) control was originally employed as part of the sugar cooking control strategy, but after observing the response of the cooking process due to measured disturbances during normal operation, it was evident that PID control could not meet the desired cooking specifications. It was decided to pursue advanced process control strategies as a means to meet the high-boil sugar cooking specifications.

The objectives of this advanced process control research study were to successfully implement viable commercial MPC technologies on the sugar cooking process and to develop a new MPC strategy implemented directly on the existing PLC. This study implemented and evaluated three PC-based, commercial MPC technologies for the sugar cooking process, and developed and implemented model predictive functionality using a combination of ladder logic code and function blocks directly on Rockwell Automation's Allen-Bradley ControlLogix® PLC[1]. The results of these solutions are compared to the traditional PLC-based PID control solution.

Ingredient cooking is one of the most common manufacturing processes within the food and beverage industry. By properly leveraging the knowledge gained from this research study, multiple sectors within the food and beverage industry may be able to reduce production costs while improving the quality of many manufactured products. The results of this study will be applied to the company's food and beverage manufacturing facilities throughout North America.

[1] This strategy was developed specific to this application, and is not a commercially available solution.

Statement of the Problem

The problem of this research study was to design and analyze the performance of a PLC-based model state feedback controller implementation for an industrial sugar cooker, and to determine its viability in comparison to commercially available PC-based model predictive controller implementations applied to the same sugar cooker.

Purpose of the Study

The purpose of this study was to validate the application of PLC-based model predictive control for industrial processes that exhibit deadtime behavior and are influenced by external disturbances. The objectives of this study that supported this purpose were:

1. To develop a PLC-based model predictive control strategy using model state feedback techniques.

2. To compare the application results for PLC-based model predictive control with commercially available PC-based model predictive controllers on an industrial sugar cooker.

3. To validate the use of MPC technologies by statistical comparison to the results of standard PID control on an industrial sugar cooker.

Need for the Study

The need for this study was based on the lack of an available model predictive control solution that can be applied directly using a typical PLC. Although there are many commercially available MPC controllers for implementation on a stand-alone PC (VanDoren, 2001, 2002) and large amounts of applications and research literature on PC-based MPC strategies (Blevins, McMillan, Wojsznis, & Brown, 2003), to date there are no commercial control packages for realizing model-based control using model state feedback techniques on the ubiquitous PLC.

Existing, commercially available model-based controllers require a personal computer (PC) for operation. Typically the existing process control system utilizes an industrially hardened programmable logic controller. As a result, the PC used for the MPC functions is an additional piece of hardware that must be purchased, enclosed, and maintained. The software run on this PC is also additional to the standard process control system. It must be purchased, configured and maintained separately from the base control system. Finally, communication drivers must be purchased and configured. These additional costs required throughout the life cycle of the system reduce the profitability of the process control system. A model-based controller implemented directly on the existing, standard process control hardware (the PLC), with the standard process control software, requires no additional hardware, software or communications drivers.

Research Hypotheses

The goal of this research study was to develop and evaluate a model predictive control strategy for implementation using a programmable logic controller. The controller selected for this study was Rockwell Automation's Allen-Bradley ControlLogix® PLC. The ControlLogix® PLC utilizes deterministic programming and is compliant with the IEC 61131-3 programming standards. The ControlLogix® PLC is an industry standard for implementing control system strategies within many manufacturing processes (Group Engineering, 2001).

The research hypotheses were:

1. It takes significantly less time for the PLC-based model state feedback implementation of the MPC controller to reach the final product temperature set point than it does for standard PLC-based PID control applied to the same industrial sugar cooker.

2. The PLC-based model state feedback implementation of the MPC controller experiences less temperature overshoot due to the initial product temperature rise than the standard PID control solution.

3. There is a smaller deviation in temperature around the set point, in the presence of system disturbances, during steady-state operation for the PLC-based model state feedback implementation of the MPC controller than there is for the standard PID control solution.

4. The temperature rise time is shorter for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions applied to the industrial sugar cooker.

5. The PLC-based model state feedback implementation of the MPC controller exhibits less temperature overshoot as a result of the initial product temperature rise than the PC-based commercial MPC solutions.

6. The deviation in temperature around the set point, in the presence of system disturbances, is smaller for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions.

Hypotheses Tests

The hypotheses were tested using the following test statistics:

1. A one-way analysis of variance (ANOVA) was used to test for a difference in the mean rise times for the different control solutions at the $\alpha = .05$ significance level. The test statistic was the amount of time it takes for the product to reach the final temperature, with the different test methods treated as levels of the variable. Specific differences among treatments were examined using Tukey's post hoc test for honestly significant difference comparisons. Summary statistics were then used to determine the direction of the observed differences.

2. A one-way ANOVA ($\alpha = .05$) was used to test for a difference in temperature overshoot as a result of the initial product temperature rise for the different control solutions. The test statistic was the amount of temperature overshoot, with the different test methods treated as levels of the variable. Specific differences among treatments were examined using a Tukey's post hoc test for honestly significant difference comparisons. Summary statistics were then used to determine the direction of the observed differences.

3. To test for a difference in deviation in temperature around the set point in the presence of system disturbances during steady-state operation for the different control solutions, two independent variables were examined. These were the test method (PID control, MSF control, MANTRA®, and so forth) and the product feed rate. The product feed rate was the system disturbance. It was varied in a prescribed, step-wise fashion. A two-way ANOVA ($\alpha = .05$) was used. Both test method and feed rate were treated as

fixed factors. The pump speed was treated as a fixed factor due to the fact that speed adjustments were only made in fixed increments based on the baseline speed. Only main effects were examined. The interactions were not examined because the independent variables were not manipulated simultaneously. Specific differences among treatments were examined using a Tukey's test for honestly significant difference comparisons. Summary statistics were then used to determine the direction of the observed differences.

Assumptions of the Study

For this study certain assumptions were made that served as the basis for the ensuing analysis. These assumptions were:

1. The sugar cooking process was repeatable. Therefore, it would function the same, given the same operating parameters, throughout all of the trials performed.

2. The process instrumentation was properly calibrated and the final control elements operated as designed.

3. The raw materials used were consistent in quality and distribution throughout the trials.

4. The utilities supplied to the sugar cooker (for example the steam supply, electrical power and compressed air) were stable, consistent, and free of damaging components (such as electrical harmonics and moisture in the compressed air) throughout the trials.

5. The same operator was used to operate the sugar cooker equipment throughout the trials in order to minimize human error. He operated this equipment consistently

throughout the trials. No special training or skill set was required for the operator to run the equipment. It was assumed that the operator did not adversely affect the process.

6. Unmeasured process disturbances and uncontrollable external influences such as fluctuating relative humidity, raw material variances, and mechanical equipment wear had an insignificant effect on the process.

7. The commercial applications were correctly implemented, with assistance from each application vendor, and performed as advertised. Therefore, these applications did not need to be validated, but were assumed to be the standard by which the PLC-based MPC controller was validated.

8. The sugar cooker designer properly implemented the existing PID controller for the steam supply-modulating valve, and the PID controller was properly tuned to meet the control objectives. The properly tuned PID controller was used as the baseline to which the MPC techniques were compared.

9. Model predictive control algorithms assume a linear process model (Blevins et al., 2003). For this study it was assumed, that the sugar cooking process was linear.

Delimitations of the Study

The following delimitations were inherent in this research study:

1. The study was delimited to an existing sugar cooker, as configured, at an undisclosed food and beverage research facility in the Midwest.

2. The study was delimited to a single process, recipe and set of operational parameters for the sugar cooker.

3. The implementation of the PLC-based model state feedback control was delimited to Rockwell Automation's Allen-Bradley ControlLogix® PLC with version 10.0 software and firmware.

4. The commercial, PC-based MPC applications were delimited to MANTRA® from ControlSoft, Inc., BrainWave® from Universal Dynamics, and Process Perfecter® from Pavilion Technologies.

5. The PID control was delimited to a single loop, parallel algorithm, implemented on the ControlLogix® PLC.

Limitations of the Study

The following limitations were applied to this research study:

1. Due to operation costs, the individual trials for each implementation were limited by the research facility.

2. No additional trials were run, or data gathered, after the initial testing period for each component of the research project. The equipment was decommissioned after the trial-runs phase of this study.

3. Due to the sensitive nature of this research and its application to real industrial processes, specific recipe and equipment details are not publishable so as to maintain the integrity of industrial trade secrets.

4. The company owning the equipment is to remain undisclosed.

Methodology of the Study

The project involved developing an appropriate interface to the existing sugar cooking control system for each MPC solution. This included control panel operator controls, communication protocols, and bumpless control transfer. The operator controls were used to start and stop the process, to switch between manual, PID and MPC control of the process, and to set operational parameters such as the process temperature set point. Bumpless transfer ensured that the process remained stable during the transition from manual to automatic control and vice versa. Appropriate communication protocols were established to communicate between the MPC computer and the process controller as applicable. For PC-based solutions the PLC functionality remained unchanged. For the PLC-based solution a separate subroutine was developed so that the integrity of the existing PLC code would not be compromised. Minimal changes were allowed to the existing processing system. This ensured that the MPC solution was highly portable for implementation on other ingredient cookers.

The cooking process started from an established steady-state condition for each trial, and was ramped up to the cooking temperature as quickly as possible without experiencing excessive temperature overshoot. The system was not ramped up to temperature in less than five minutes. Doing so would have caused the product to burn and foul the walls of the heat exchanger, thus reducing heat transfer capacity. The controller was then required to maintain that cooking temperature while the process was operated through a preset sequence of tests that introduced disturbances into the cooking process.

Process Description

1. A high viscosity sugar syrup mixture was blended and heated to a temperature of 170°F in an 1800-pound batch mixer (see Figure 1).

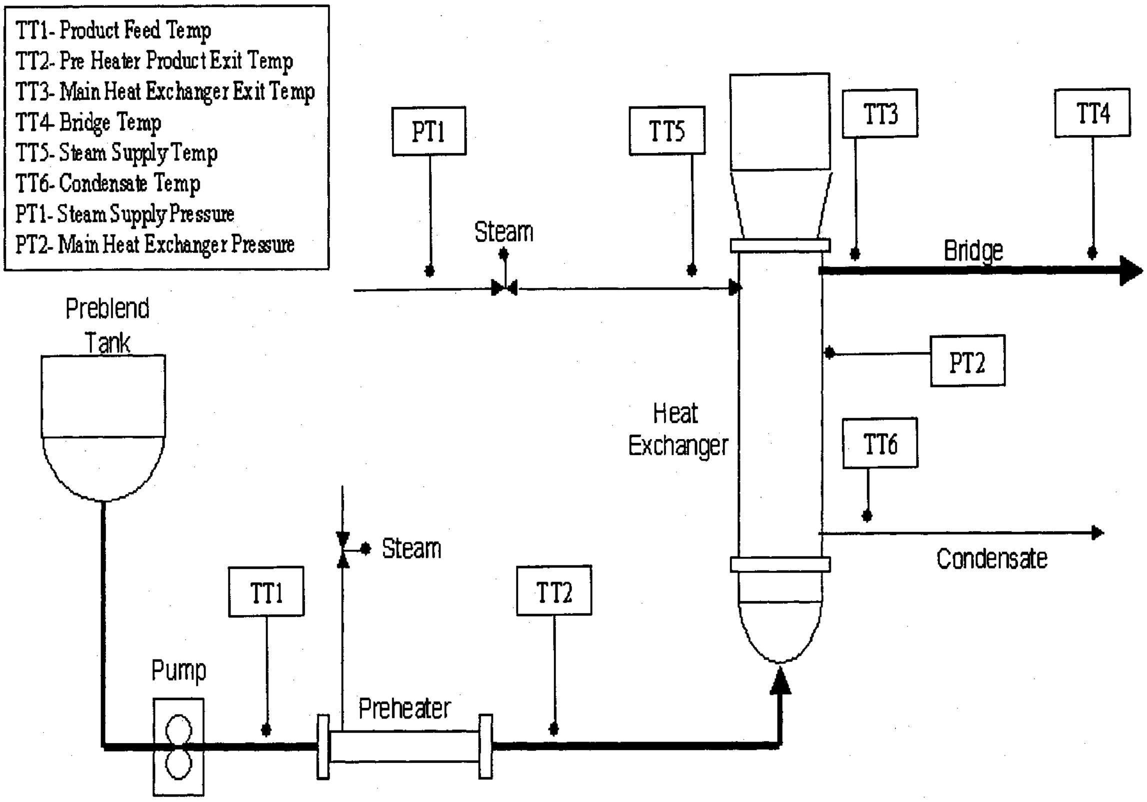


Figure 1. Sugar cooker process flow diagram.

2. Product was gravity fed to a positive displacement Waukesha pump (product feed pump). The product feed pump delivered the sugar syrup mixture at a rate of 8.5 pounds per minute through a small shell and tube heat exchanger (preheater). The product reached a temperature of 220°F at the discharge of the preheater.

3. Product entered the bottom of the sugar cooker shell and tube heat exchanger section at a constant flow rate of 8.5 pounds per minute, the product residence time in the heat exchanger was approximately 31.5 seconds. The product velocity in the tubes was approximately 0.11 feet per second.

4. Product exited the top of the heat exchanger and flowed down a transition duct (bridge) that was at atmospheric pressure. A resistive temperature device (RTD TT4) positioned in the bridge provided feedback regarding the product's discharge temperature from the cooker.

5. The control loop modulated a proportional steam supply valve to regulate the sugar cooker's internal vessel pressure, thus controlling the cooking temperature.

6. The desired outcome of the study was to maintain a discharge product temperature at the bridge (TT4) of 274°F ±1°F. If product temperature was too low the resulting sugar mixture would not caramelize, and would need to be reprocessed. If the product temperature got too high the cooker would foul and the product would be unusable. Therefore, the usable control limits were ±5°F over short (less than 30 second) periods.

7. The product feed rate varied by a maximum of approximately ±25%, as determined by downstream equipment. Typical feed rate variations were held between 5% and 10%. As the feed rate increased the product residence time in the heat exchanger decreased. The opposite was the case as the feed rate decreased. The control system had to maintain the discharge product temperature at the bridge regardless of feed rate (within the ±25% range).

Control Objectives and Analysis Criteria

To evaluate controller performance, the following objectives were established and measurable parameters were recorded:

1. The MPC controller was required to automatically ramp up the product temperature in 15 minutes or less.

2. The controller was to maintain a steady-state discharge product temperature at the bridge (TT4) of 274°F (±1°F on average) for a period of 10 minutes. A constant product feed rate was assumed.

3. The controller was to maintain a discharge product temperature at the bridge (TT4) of 274°F (±1°F on average, with a maximum deviation of ±5°F) while the product feed rate was varied by approximately ±25% in a prescribed test pattern.

4. The MPC solution was to show consistent start-up without fouling.

Standard PID control response was recorded and used as a baseline for comparison against all four model-based controllers.

Definition of Terms

Following are certain terms used throughout this research study that, although not unique to this study, have been defined in order that readers have a common basis for understanding their use within the context of this research. The following terms were defined:

Accuracy – the difference between the measured and the true value of a quantity.

Advanced process control (APC) – the application of non-traditional control methodologies that "seek to discover, incorporate, and exploit knowledge about raw materials, process, product, equipment, instrumentation, and final elements" (Blevins et al., 2003, p. 1).

Bumpless transfer – the transition from one control mode to another wherein the process variable responds with little or no sudden deviations due to the transfer of control.

Communication drivers – software applets that enable data to pass between disparate systems.

Control loop – the method of adjusting the control variable in a process control system by analyzing process variable data and then comparing it to the set point to determine the amount of error in the system (Bryan & Bryan 1997, p. 1005).

Control horizon – the number of manipulated-variable moves ahead to be taken into account by the model predictive controller (Blevins et al., 2003, p. 363).

Control variable (CV) – the independent variable manipulated by the controller output that in turn adjusts a process actuator so as to reduce the error between the set point and the process variable.

Deadtime (θ) – delay from when the control variable is modified to when the process variable begins to react detectably to that modification.

Derivative action (D) – the controller output changes that are proportional to the rate of change of the error between the set point and the process variable.

Feed forward control – a control algorithm used to manipulate the control variable based on a measurement of a disturbance variable so as to proactively cancel the effects of the disturbance.

Feedback control – a control algorithm used to reduce the error between the set point and the process variable, e.g., a PID controller.

Gain (K) – ratio of the resultant change in the process variable to a step change in the control variable. Therefore, K = (%ΔPV / %ΔCV).

Heartbeat pulse – a time-based pulse train issuing from one controller and communicated to another to indicate that the originating controller is still operational. On loss of the heartbeat the control algorithm is switched to the receiving controller.

Human-machine interface (HMI) – is the means by which the operator controls the process equipment. Typically this is a graphical interface.

Integral action (I) – the controller output changes that are proportional to the integral of the error between the set point and the process variable.

Measured disturbance variable (DV) – a parameter that affects the control system response, is measured by the system controller, but not managed by the controller, and is included as feed forward variable to the system controller.

Multiple-input, multiple-output (MIMO) – A process having multiple input variables that control the process in such a manner as to affect multiple output variables. For example the moisture and density of a powder are controlled by the flow rate through a drying chamber with a given drying temperature, where flow rate and temperature are both inputs to the system, and moisture and density are the required outputs.

Model predictive control (MPC) – "model-based control of future trajectory of a process variable from the changes of control variables" (Blevins et al., 2003, p. 421).

Non-linear system – the deadtime, time constant, or gain are not constant, but are a function of time, direction, operating point, or load.

OLE – Object Linking and Editing.

OPC – OLE for Process Control. A standard communications protocol used to communicate between industrial processes.

Precision – the short term maximum variation in the output for the same input approached from the same direction and at the same conditions. See also Repeatability.

Prediction horizon – the range in scans of process output prediction (Blevins et al., 2003, p. 363).

Process control - refers to the equipment and methods used to automatically measure and manipulate the conditions of continuous and batch processes.

Process variable (PV) – the controller input to be maintained at a specified set point. Typically a process output such as pressure, temperature, flow or level.

Programmable logic controller (PLC) – an industrialized computer with specially designed architecture for interfacing to field devices that is used to scan inputs and manipulate outputs based on specific algorithms, sequences, and conditions.

Proportional action (P) – the controller output changes that are proportional to the change in error between the set point and the process variable.

Proportional-integral-derivative (PID) control – A continuous-mode controller that uses proportional, integral, and derivative actions to determine the control variable output based on the amount of error, its change over time, and its rate of change (Bryan & Bryan, 1997, p. 1018).

Repeatability – the short term maximum variation in the output for the same input approached from the same direction and at the same conditions. See also Precision.

Reproducibility – the long term maximum variation in the output for the same input approached from the same direction and at the same conditions.

Resolution – the minimum change in the input in the same direction that will cause a detectable change in the output.

Response time – the amount of time it takes a process to settle acceptably after an initial process disturbance.

RTD – Resistive Temperature Device.

Scan time – the time interval between successive controller updates of the system variables.

Sensitivity – the ratio of the steady-state change in output to the change in the input.

Set point (SP) – The target output value for the process variable.

Single-input, single-output (SISO) – A process having a single input variable that controls the process in such a manner as to affect a single output variable. For example an oven uses gas flow to the burner (single input) to control the temperature of the oven (single output).

Time constant (τ) – for first order systems forced by a step or an impulse it is the amount of time required for the process variable to reach 63.2% of its steady-state value, after deadtime, as a result of a change to the control variable. The time constant is the fraction of the total change in the output variable as a function of time and is expressed as an exponential response term, $1\text{-}\exp(-t/\tau)$. When $t = \tau$, the resulting response time is 63% of the total change.

Tuning – adjustment of the proportional, integral, and derivative mode settings for modifying the behavior of a PID controller. Adjusting of model parameters for modifying the behavior of a model-based controller.

Unmeasured disturbance variable (dV) – a parameter that affects the control system response, is typically not measured or managed by the system controller, and thus not accounted for in the control system.

Summary

Advanced automation solutions seek to extract optimum value from existing manufacturing assets. The goals of such an endeavor may include increased product consistency, decreased start-up time, increased production rate, decreased feedstock/utility consumption, and process mastery. Model-based control is a proven technique of advanced control that predicts future changes to a process and makes preemptive control adjustments to keep the process on track. MPC provides continuous model-based control of end-product quality, automated adjustment to changing environmental and feed conditions, and the ability to observe and respond in a coordinated and consistent fashion to a multivariate problem. MPC is appropriate for implementation on systems that exhibit complex interactions, complex dynamics, economic trade-offs made daily with limited tools, or tight quality control – where increased quality means increased profits.

For the sugar cooking process the ability to control the product temperature in an accurate and timely fashion is particularly important. If the product temperature rises too far above the set point, the possibility of solidifying the product inside the heat exchanger exists. If the product temperature falls too far below the set point, the final moisture specification will be too high and product quality compromises such as texture variations will occur. Additionally, the product feed rate changes relative to downstream equipment conditions. These unpredictable conditions can cause feed rate variations of a maximum of approximately ±25% throughout a production run. This variation is known to have a dramatic affect on temperature control.

Different commercially available advanced control techniques were implemented on a sugar cooker to determine a baseline by which to validate advanced regulatory control techniques applied directly on a PLC. Three options were investigated from the range of commercial MPC software packages. These were ControlSoft, Inc.'s MANTRA®, Universal Dynamics' BrainWave®, and Pavilion Technologies' Process Perfecter®. A model state feedback solution was developed and implemented directly on a ControlLogix® PLC using ladder logic and function block programming. The PLC-based solution was validated against the commercial MPC applications. These solutions were compared to the PID control algorithm currently used on the sugar cooker.

There are several benefits of direct PLC implementation as opposed to a PC-based implementation. These include no external hardware required, no additional software required, no communications (such as OPC) required, easier to maintain at the plant level, and reduced implementation costs. These benefits coupled with the additional benefits inherent to model predictive control strategies will be leveraged across numerous production facilities throughout the food and beverage industry.

In the remainder of this document, Chapter 2 lays the foundation for this research through a review of PID control, a brief discussion on the ubiquitous nature of the industrial PLC, an overview of advanced process control history and techniques, and an assessment of some of the commercially available model-based solutions. The chapter concludes with a look forward to the development of a model-based solution designed for application directly on a PLC. Chapter 3 discusses in detail the development and application of the model state feedback algorithm on an industrial PLC. The advantages and limitations to this approach are examined along with the implementation of several commercially available PC-based solutions. The results of each of these solutions are described in Chapter 4. These results are compared to each other and to the standard PID approach using analysis of variance, appropriate post hoc tests and summary statistics. Finally, conclusions are drawn and recommendations made in Chapter 5 based on the statistical analysis and the research experience.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter will review the major advances in regulatory control that lead up to the development of a PLC-based model predictive controller for application in industrial food processes. The chapter begins with an overview of proportional-integral-derivative control, the most fundamental control algorithm used today. A review of recent advances in programmable logic controllers follows. These advances make it possible to construct the advanced control algorithms directly in the PLC. A review of current advanced process control techniques then builds the foundation for application of the model state feedback algorithm. The chapter concludes with a review of commercially available advanced process control solutions for implementation on a personal computer, and looks forward to the development of an advanced control solution built directly on an industrial programmable logic controller instead of a personal computer.

"It is proven that intensive, well-organized use of advanced process control (APC) can substantially increase the profitability of a plant. One of the key effects is reducing the ever-present variations in the process more than standard controllers or the operators can do" (Eder, 2003b, p. 1). Reduced process variability has a direct, positive benefit on the process. When the variability is reduced the process may be operated closer to the process set point or limit. This shift in the mean operating point results in a more efficient and cost effective operation. Figure 2 illustrates how shifting the set point translates to benefits as seen on a trend recording of a typical process variable. The full benefits of this shift are only achievable using advanced control techniques.

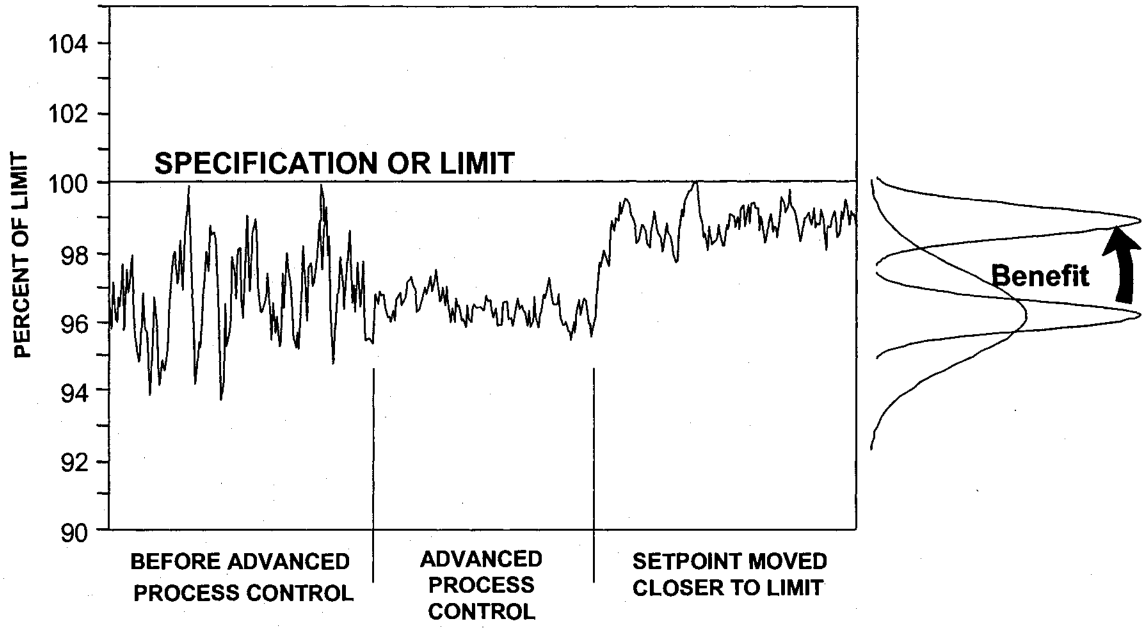


Figure 2. Benefits derived from using advanced process control to reduce variability.

In the food and beverage industry reduced process variability and operation closer to specifications directly affect the cost of goods sold. Government regulations, such as USDA requirements, dictate the limits for such parameters as moisture content and product net weight. By operating closer to the specified limits a company would for instance be able to sell more water on average and produce less overweight packages, thus reducing the amount of product effectively given away to the consumer.

In an effort to reduce process variations and increase process efficiencies, there are three main methods of control on which advanced process control techniques are applied. These are:

1. Regulatory control, which is provided when a process variable needs to be maintained at a given target value, and a standard PID regulatory controller provides insufficient control response.

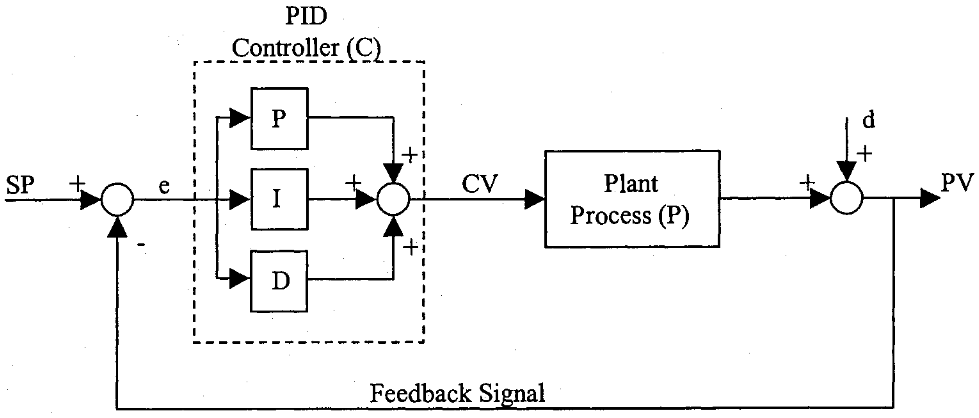
2. Constraint control, which is developed when there is no exact target set point given, but only an operational direction. The variable in question needs to be moved in this direction until a limitation is encountered.

3. Optimization techniques, which are applied when both positive and negative influences exist on an objective and the optimal operating point, that is, the point where these positive and negative influences just balance out must be found.

This study focuses on the first type of control, specifically regulatory control. The following sections outline the regulatory control methods investigated for the industrial sugar cooker. These included standard PID regulatory control and several model predictive regulatory controllers.

Proportional-Integral-Derivative Control

The proportional-integral-derivative (PID) control algorithm is ubiquitous in today's process industries. It is a simple and easy to implement regulatory control algorithm that uses feedback to generate a control output which causes a corrective effort to be applied to a process to reduce the error between the desired process value and the actual process value. The PID algorithm observes the value of the error between the set point and the actual process state, the integral of the error over a recent time interval, and the current derivative of the error to determine the magnitude and duration of the corrective action required to eliminate the error. The P, I and D values, or terms, depend on the characteristics of the actual process and must be properly tuned to yield a satisfactory control response. The proportional term causes a larger control action to be taken for a larger error. The integral terms adds to the control action if the error has persisted over the recent time interval. The derivative term modifies the control action if the error is changing rapidly with time. Figure 3 shows the control block diagram for a typical PID controlled process.



where,

C:	process controller
P:	physical plant process to be controlled
SP:	desired set point
PV:	actual process value
CV:	control value to the process actuator
e:	error between the SP and PV
d:	effects of an external system disturbance
P,I,D:	control terms applied to the error to modify the CV

Note that P indicates both the process and the proportional term in the PID controller depending on context.

Figure 3. PID control system block diagram for the parallel PID algorithm.

There are three basic configurations of the PID algorithm, as shown in Equations 1, 2 and 3. The ideal algorithm in Equation 1 is the most mathematically straightforward, but is difficult to implement in real-world controllers. The parallel algorithm shown in Equation 2 uses three independent calculations for the proportional, integral, and derivative constants. This configuration is designed so that the values of the various constants do not affect each other. However, this makes the algorithm difficult to tune. Lastly, the series algorithm, or interacting algorithm, given in Equation 3 is designed so that a portion of the output of one calculation is used in the input to the next calculation. This is the most common algorithm used in industrial control.

$$P(e(t) + I\int e(t)dt + D\frac{de}{dt}) \tag{1}$$

$$Pe(t) + I\int e(t)dt + D\frac{de}{dt} \tag{2}$$

$$P(e(t) + I\int e(t)dt)(1 + D\frac{de}{dt}) \tag{3}$$

where,

P: proportional term in percent (%) gain
I: integral term in minutes
D: derivative term in minutes
e: error between the set point and the process variable
t: time

The details and applications of these algorithms can be found in most control systems text books, as well as in Bryan and Bryan (1997), Gerry and Shinskey (2000), Harrold (1999), Lefkowitz and Beiler (1996), Johnson and Malki (2002) and VanDoren (1998).

PID Loop Tuning

The simplicity of the PID algorithm makes it easy to understand, implement, and diagnose when it doesn't perform optimally. Optimizing the performance of a PID loop involves tuning for both servo response (set point changes) and regulation (load rejection). Tuning a PID controller is the process of selecting the correct complement of proportional, integral and derivative actions to achieve a desired closed-loop performance. Tuning minimizes the detrimental effects of disturbances, interactions, control valve dead band and process deterioration over time. It is the "largest, quickest, and least expensive improvement one can make in the basic control system to decrease process variability" (Blevins et al., 2003, p. 183). However, according to Chia and Lefkowitz, "tuning control loops has been a problem since the earliest applications of PID control. It has been noted that a significant fraction of PID control loops in plant operations are not properly tuned" (VanDoren, 2003, p. 203). Buckbee (2002) shows how poorly tuned control loops cause increased operational costs.

There are many different tuning methods. Harrold (1999) gives a number of useful guidelines for manually tuning PID control loops. However, manually tuning control loops is a time consuming and often difficult endeavor. A more efficient way to collect and analyze process data for proper loop tuning is to use software developed specifically for loop tuning. One such software package is INTUNE® from ControlSoft, Inc. (Lefkowitz & Beiler, 1996). Loop tuning software packages provide a simple and quick method for detailed analysis of process loops. Features such as diagnostic reporting, tuning for a particular process and performance objective, estimates of performance and robustness, and predicted response plots are all part of a typical loop tuning software package. Chia and Lefkowitz (VanDoren, 2003) further indicate that the value of these packages is evidenced by their increased sales and integration into both PLC-based and stand-alone PID controllers. Additional PID tuning algorithms and methods are available in McMillan (2000) and Parker (2002).

Limitations of the PID Algorithm

Aggressive tuning techniques notwithstanding, there are certain conditions for which a PID controller is ill suited and can not be adequately tuned to meet the system control objectives. Tzovla and Mehta (2000) point out that it is "difficult to adequately control multiple-input multiple-output processes, processes with constraints and/or disturbances, and processes with associated complex dynamics using conventional PID-based approaches" (p. 1). The PID algorithm was designed for stable, robust, linear processes with a single input and single output parameter.

One of the most difficult problems to overcome with a feedback controller is process deadtime. Deadtime is the delay between the application of a control effort and its first effect on the process variable. It is typically a result of transport delays between the actuating device and the sensing device in a process. During the deadtime the process does not react to any controller efforts. Therefore, any attempt to manipulate the process variable before the deadtime has elapsed inevitably fails. The design of a feedback controller is such that it will always attempt to minimize the error in the process variable. Thus the controller overcompensates for the error by continuing to manipulate the control variable during the deadtime. This results in inaccurate error suppression and ultimately failed control efforts.

There are numerous well-developed techniques for improving the effectiveness of a PID controller, such as gain scheduling for set point dependent processes, cascaded loops for interacting variables, and the Smith predictor for deadtime-dominant processes (Boyce & Brosilow, 1996; Gerry, 1998; VanDoren, 2003). However, even with these enhancements and proper tuning, the PID controller is typically incapable of controlling processes with large variability, process changes after startup, highly interactive variables, or long deadtimes. The fundamental problem with the PID controller is that the control action is based on the instantaneous error between the set point and the process variable without taking into account the effects of previous control actions to which the process has not yet responded. Kay and Thompson (1998) and VanDoren (2002) suggest that the most effective method for overcoming these issues is using model-based control. This method will be discussed in detail later in this chapter.

Programmable Logic Controllers

As with the PID algorithm for analog control, the programmable logic controller (PLC) has become an industry standard for discrete control systems. The PLC is essentially a simplified, industrially hardened computer. It has input/output structures specifically designed to acquire real-world information such as temperatures, pressures, positions, and composition, and to control real-world outputs such as valves, relays, and indicators. Several programming languages are available for developing applications on a PLC. The most common are ladder logic and function block diagrams. Ladder logic is a direct implementation of the hardwired relay logic from which the PLC was developed. All PLCs are capable of being programmed using this language.

The PLC is used throughout industry for controlling a wide variety of manufacturing processes such as packaging, batching, mixing and refining. It was originally developed to replace relay logic control systems. These systems were hardwired control systems that were not easily changed once installed. The PLC was developed to offer a method for developing these same control algorithms using software that is easily changed and manipulated as required by the process. For a complete overview of the PLC and its programming see Bryan and Bryan (1997), Godena and Colnaric (August, 2000), Johnson and Malki (2002), Martin (2002) and Rockwell International Corporation (2002).

Over the years PLCs have become standardized. Today all PLCs perform a standard set of functions and routines that are common throughout industry. Examples include timers, counters and mathematical functions. One of these common functions is the PID control loop algorithm. In a standard PLC the proportional, integral and derivative constants are programmed into a function block in the PLC that then executes the PID algorithm using the selected input and output (such as a temperature and steam control valve). The integration of analog process control functions, such as the PID control loop algorithm, allows the PLC to provide a complete manufacturing control system solution. For this reason it is on this platform that we wish to develop advanced control techniques.

Rockwell Automation's ControlLogix®

The ControlLogix® PLC from Rockwell Automation allows programming in both ladder logic code and function block diagrams. A ControlLogix® program may be written using integrated ladder logic subroutines and function block diagrams. The system maintains and updates all data values, including model predictions, in a tag data table that may be monitored in real-time or via an off-line log using a data collection system.

Rockwell International Corporation (2002) states that unlike many other PLC systems the ControlLogix® facilitates deterministic (i.e., fixed timing) execution and scheduling of the ladder logic subroutines and the function block diagrams. A ladder logic routine may execute sequential logic functions that are not time critical such as a communications watchdog, parameter setting and calculations, initialization, mode switching, and fault handling using non-deterministic execution. During this non-deterministic execution a function block diagram may be called by the ladder logic routine and deterministically executed as a part of the overall program. The entire program may represent a multivariable model predictive controller with appropriate timing, bumpless mode switching, control algorithm execution and other necessary features ensuring correct and robust operation in real-time. This ability to perform deterministic routines is a key factor in developing the advanced control solution presented in this research.

Advanced Process Control

Process control is all about understanding and controlling change. Changes or variations in raw materials, equipment performance, production rates, utilities, ambient conditions and process set points all necessitate the use of process controllers. Nothing remains constant during actual plant operations. Controlling the outcome of these changes provides opportunity for increased efficiency, decreased operating costs and improved product consistency and quality. While the PID algorithm is capable of controlling the vast majority of industrial processes, it is due to its lack of ability to control deadtime dominant systems and systems with multiple interacting input and output variables that advanced process control techniques have been a major focus of development over the past two decades. “Advanced process control is the intelligent, well-managed, intensive use of technology, systems, and tools based on sound process knowledge, with the objective being to deliver substantial benefits to plant operations in a most cost-effective and timely manner” (Eder, 2003a, p. 1).

Advanced process control (APC) has historically been achieved using the computational power and speed of PCs and UNIX machines. Traditional PLCs have not supported the required floating-point math or precise timing necessary to implement advanced algorithms. Because of this, with the exception of fuzzy logic that does not require precision timing, APC has not been realized on the PLC platform. APC has been implemented on some industrial distributed control systems (DCS) such as Emerson’s DeltaV® and Honeywell’s TDC3000®. However, DCSs are very expensive to own and maintain for single processes. They are typically used as a plant-wide control system in plants where the processes have a single purpose and are integrated from beginning to end. In the food and beverage sector this situation is typically not the case. Here processes tend to be stand-alone, flexible systems that require individual PLC-based control systems. As a result, APC applications in the food and beverage industry to date have been implemented using personal computers communicating to the PLC over a control network. Barnes (1996) illustrates this hybrid method of PC-based APC communicating with a standard PLC.

Types of Advanced Process Control

There are a number of APC algorithms. Each has seen some degree of industrial success. This section will briefly outline some of the many types and applications of APC technology in use today.

Perhaps the most common APC method is Fuzzy Logic. Fuzzy logic is a mathematical technique for set theory computation. It allows control system designers to develop control systems based on a set of grammatical rules that are then transferred to numerical sets for computation of a control effort. These systems have become a standard for regulatory control and simple optimization in the process industries. Fuzzy logic algorithms are now built in to many PLCs as a standard tool.

Artificial neural networks (ANN) are loosely modeled on biological reasoning. They are capable of learning through assigning levels of importance to connections between processing nodes. Given a proper configuration of nodes and connections, ANNs are capable of modeling any controllable system. ANNs are typically reserved for optimization techniques due to the intensive engineering required to develop a system, though they can be used for regulatory and constraint control.

Expert systems use sophisticated search methods to associate large quantities of human knowledge that have been codified in a database. They are designed to simulate human reasoning and are typically used as a decision maker or advisor for solving complex problems by drawing on these vast quantities of compiled knowledge. Expert systems are often found functioning as on-line plant maintenance advisors to technicians, forecasting potential problems based on current conditions and past experience.

Model predictive control (MPC) is another branch of APC that focuses mainly on regulatory and constraint control. Control algorithms are based on mathematical models of the process system. The controller then uses the model to predict future response requirements based on current conditions. This allows the controller to compensate for process lags and interactions that are beyond the capability of a PID controller. This research is focused on the application of model-based predictive control techniques.

APC Applications in Industry

Advanced process control techniques have been in use in industry for over 20 years. The installed base includes applications of artificial neural networks, fuzzy logic and other rule-based systems, model-based predictive control, and others. Following are a representative sampling of industrial applications using various advanced techniques. This is by no means an exhaustive overview of the successes realized with these technologies.

Studebaker (1995) has shown how internal model control (IMC) techniques have been utilized for controlling long deadtime (approximately 90 minutes) in an oil cracker while taking tarry residual oil and converting it to useful products. The improved control resulted in a $600 per day savings, generating an effective payback in less than 25 days. Chia and Brosilow (1991) showed similar results in applying a multivariable IMC technique to a heavy oil fractionator. It was shown that the advanced control techniques were able to handle output constraints and provide decoupled control much better than the traditional methods employed by Shell Oil. IMC techniques have also been used to reduce throttle pressure on a load increase in a utility boiler while maintaining maximum megawatt control, a task previously unachievable (Studebaker, 1995).

Molson Brewers in Canada uses model predictive control to establish boil consistency in the wort kettles during the beer brewing process. The process is considerably upset by the addition of hops, changes in atmospheric conditions and venting fan operations. These random changes coupled with the formula differences of the various products make it impossible for PID controllers to maintain control of the system. Implementing the model-based controller increased production capacity by 12% (Kay & Thompson, 1998). Similar results have been experienced in a variety of industrial processes including temperature control in beverage bottling, flow control in air separators and scrubbers, feed rate control during ore grinding operations, and numerous others (VanDoren, 2001).

Artificial neural networks have also enjoyed a great amount of success in controlling and optimizing industrial processes. One of the most frequent utilizations of this technology is in developing virtual sensors that can measure physical parameters that are traditionally difficult to measure. Helps and Griffen (1994) developed a virtual temperature sensor for accurately determining the internal mold cavity temperature of an injection molding operation. The mold cavity temperature is distorted by the cooling water jackets in the mold, making infrared and other traditional temperature sensing methods inaccurate. The virtual analyzer was shown to overcome these environmental difficulties in accurately measuring the internal mold cavity temperature profile. Guinan, Kelly, and Semrad (1998) have further shown that virtual analyzers can be coupled with

ANN control algorithms to effectively control nitrogen oxides emissions from a power station. An 18% to 21% reduction in emissions was realized by using the combined ANN technologies. In the food and beverage industry the author has used ANN technology in evaporation and drying operations. Evaporation and drying require tight control over temperatures, pressures and flow rates, all of which are interacting. Neural network control of these parameters has lead to increases of up to 0.4% moisture and 5% throughput while reducing energy consumption by up to 4.5%.

It should be noted that all of the applications discussed to this point have been developed and run on a personal computer rather than an industrial PLC. Historically these types of solutions have required deterministic timing and computing power that was not available in PLCs. As a result the only APC solution available for implementation directly on a PLC to date is fuzzy logic. The reason that fuzzy logic has been successful on a PLC platform is that a complex mathematical model of the process is not necessary. Instead, the control algorithms are rule-based. Anderson, Blankenship, and Lebow (1988) and Manesis, Sapidis, and King (1998) have each shown that fuzzy logic can be successfully integrated into an existing PLC to enhance PID control and wastewater treatment respectively. Blevins et al. (2003) also show that fuzzy logic control can be successfully applied on a secondary computer communicating with a PLC. Their system was successful for controlling the temperature of a chilled water refrigeration unit that was frequently exposed to large unmeasured load disturbances and large set point changes. They have also successfully controlled the moisture of a drying oven in a continuous dyeing operation for solid-color tufted carpeting while eliminating carpet overheating due to temperature overshoot during set point changes.

Hundreds of other detailed examples on applying advanced control techniques to industrial processes can be found throughout the literature. See Griffen (1995); Tay (1996); Demoro, Axelrud, Johnson, and Martin (1997); Manesis et al. (1998); Huzmezan, Gough, Kovac, Le, and Roberts (1999); Havener and Unrau (2001); Huzmezan, Dumont, Gough, and Kovac (2001); Martin (2002); Blevins et al. (2003); or VanDoren (2003) for a sampling of various industrial applications.

Model-Based Control

"The advanced control technology with the best track record to date for increasing plant efficiency and capacity is the model predictive controller" (Blevins et al., 2003, p. 93). The model predictive controller is a model-based control system. Model-based control systems employ a mathematical model of the process for developing and implementing the control system. Controller design is based on the adjustment of that model to predict future process response and calculate the control action required to reach and maintain a set point.

Model-based controllers are of particular interest for use on complex processes with significant deadtimes that typically cause PID-based controllers to overcompensate and lose control of the process. Improved control performance is achieved through basing the control action on the mathematical model of the process, including deadtime, so that the control action takes into consideration the effects of past control actions that have not yet appeared in the process variable (due to deadtime), as well as the long-term consequences of the currently calculated action. The mathematical model is adjusted to compensate for changes in the process characteristics so that the controller can maintain control under various operating conditions (Universal Dynamics Technologies Inc., 1998a).

A mathematical model is a relationship between the inputs and outputs of a system that is expressed in terms of mathematical equations. Figure 4 shows a block diagram for a generic process. The process has certain external input variables, given by u(t), that affect the internal state of the system. The process also has certain output variables, given by y(t), that cause some external action. Inputs can either be controlled variables that are deliberately manipulated, or disturbance variables that are not controllable by the process controller. The types and number of system variables present, as well as the output response to a change in the inputs, affects the type of mathematical model required to develop an effective model-based control system.

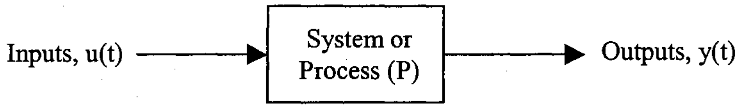


Figure 4. Model-based controller block diagram.

Several process models are of importance in controlling typical industrial processes. First principle models are based on fundamental relationships such as mass or energy balances, thermodynamic equations, etc. These are typically expressed as differential equations in the time domain, and are called state-space models. By applying the Laplace transform to these models and expressing them in the s-domain, the calculus-based differential equations may be converted to algebraic equations for easier computation and manipulation in digital processing systems such as PCs and PLCs. These models are known as transfer function models.

Often times the fundamental relationships in a process are not known and not easily derived. In these cases the process model is developed based on empirical methods that capture the input-output relationships in the process. Step tests are performed wherein the process actuator is instantaneously increased or decreased a prescribed amount and the process response to that change is recorded. The shape of the process response curve dictates the type of mathematical model, or transfer function, used to describe the system.

Brosilow and Joseph (2002) discuss several common single-input, single-output (SISO) transfer functions found in industrial processes. Their discussion includes:

1. The gain model that is used when the system dynamics are negligible. In this model the system input is multiplied by a preset gain to obtain the system output.

2. The integrator is used when there is an accumulation of mass or energy in the system. Level control is a common application of this model, for example a tank level control with independent inflow or outflow.

3. Time delay models are used to describe systems that incorporate transport lags caused by flow through pipes, transmission delays, etc. These are also known as deadtime dominant systems.

4. First-order lag models apply to systems where the accumulation of mass or energy is dominated by one term, such as with many temperature control systems. These systems include both a gain and a time constant.

5. A second-order system has a second-order polynomial in s in the denominator, where s is the Laplace operator. Oscillatory systems are characteristic of second-order systems, where the oscillations are controlled by a damping factor (ξ).

Table 1 summarizes these models and their transfer functions. There are many other process models that describe particular features seen in empirical modeling. Examples include lead-lag models that approximate differentiation, and inverse response models that describe processes that have an initial response opposite to the final process change. Specific mathematical model development depends on the actual process response curve.

Table 1

Mathematical Process Models and Transfer Functions

Model Type	Time Domain Representation	Laplace Domain Representation
Gain	$y(t) = Ku(t)$	$y(s) = Ku(s)$
Integrator	$y(t) = \int u(t)dt$	$y(s) = \frac{u(s)}{s}$
Time Delay	$y(t) = y(t - \theta)$	$y(s) = e^{-\theta s}u(s)$
First-Order Lag	$\tau\frac{dy}{dt} + y = Ku$	$y(s) = \frac{K}{\tau s + 1}u(s)$
Second-Order	$\tau^2\frac{d^2y}{dt} + 2\tau\xi\frac{dy}{dt} + y = Ku$	$y(s) = \frac{K}{\tau^2 s^2 + 2\tau\xi s + 1}u(s)$

The fundamental idea behind model predictive control is to use information generated by a model of a controllable process to control the process variables as close to the target or set point as possible, or to control the entire process to a specific objective.

Therefore, the process model is the most critical component of a model predictive control system. Dynamic models are frequently used to describe process behavior over a period of time and predict future values of the controlled signals based on current controller output and feed forward signals. The controller output is typically calculated such that the process variables follow a desired trajectory.

Several requirements must be met to successfully apply a model-based control strategy. Rivera (1999a) indicates that these include:

1. The system must be internally stable. Bounded inputs to the control system must result in bounded outputs elsewhere in the control system.

2. The system must be proper. The controller must not differentiate step changes (e.g., avoid subjecting step inputs to pure derivative action).

3. The system must be causal. The controller must not require prediction, i.e., it must rely on past and current measurements of the process.

One of the most common models used to describe real industrial processes is a combination of the first-order lag model and the time delay model called the first-order plus deadtime (FOPDT) model. It describes the open-loop response of many process systems. Equation 4 shows the mathematical model.

$$PV = \frac{Ke^{-\theta s}}{\tau s + 1} CV \tag{4}$$

where,

K:	process gain
θ:	process deadtime
τ:	process time constant
s:	Laplace operator

For a first-order plus deadtime process, the process model is based on the following three parameters:

1. Deadtime is the elapsed time from when the control variable is modified until the initial reaction to that modification is seen in the process variable.

2. Time constant is defined as the amount of time required for the process variable to reach 63.2% of its steady-state value as a result of a change to the control variable. It is a measurement of how fast the process variable will approach steady-state after the initial deadtime period.

3. Process gain is the ratio of the magnitude of the resultant steady-state change in the process variable to a step change in the control variable (Equation 5).

$$\text{Gain(K)} = \frac{\%\Delta \text{PV}}{\%\Delta \text{CV}} \tag{5}$$

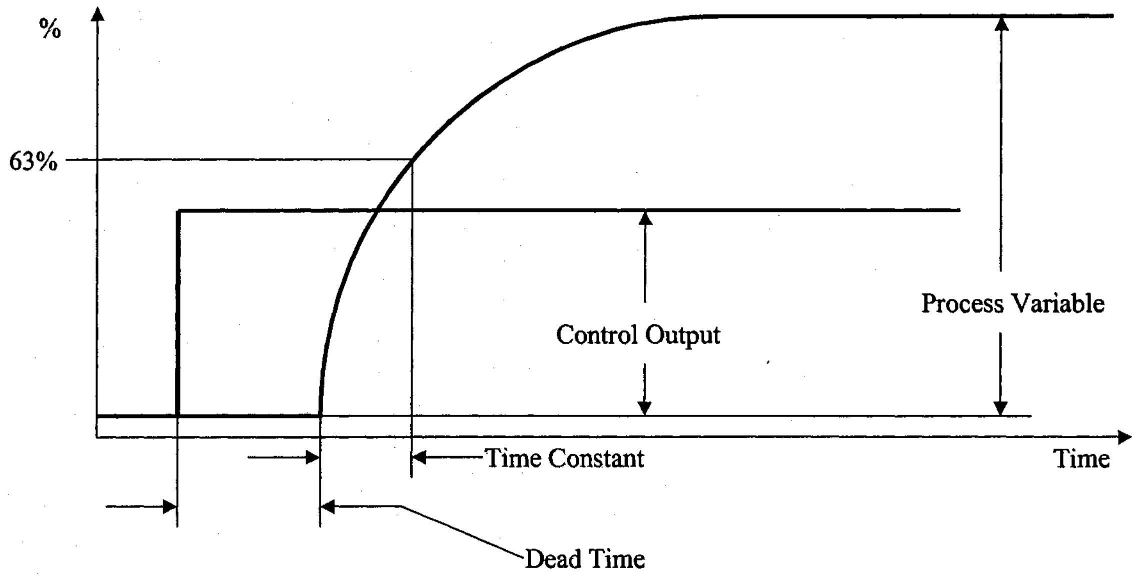


Figure 5. Parameters for a first-order plus deadtime process model.

A FOPDT mathematical model is perhaps the most common process model in the food and beverage industry. For most processes this model is simply developed using empirical data. The temperature control loop for the sugar cooking process that is the focus of this research is one such process. There are several model-based algorithms capable of controlling this system. An overview of four of these methods is discussed in the following section. Additional details on model-based algorithms can be found in Entech Control Engineering Inc (1993), Bequette (1998), and Isaksson (1999).

All of the model-based predictive control algorithms discussed in this research follow the general block diagram shown in Figure 6 and are developed based on the parameters shown in Figure 5. The particular control algorithms vary between the APC methods, but the control structure remains consistent.

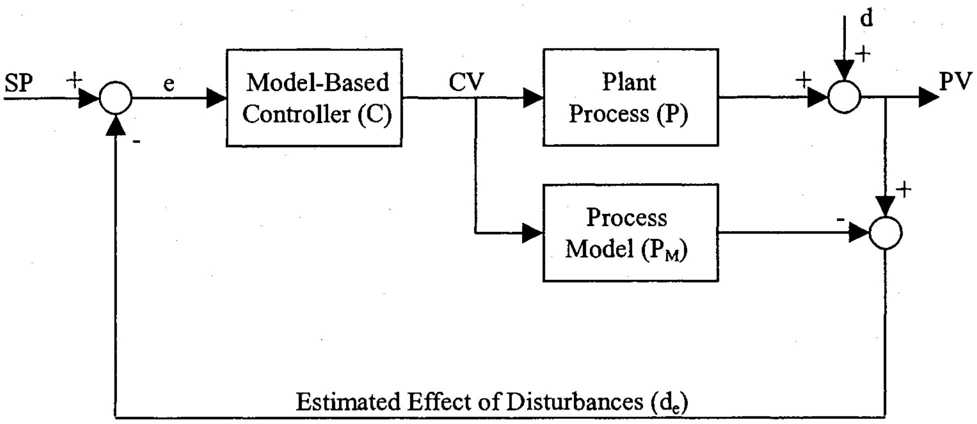

SP
+
-
e
Model-Based Controller (C)
CV
Plant Process (P)
Process Model (P_M)
d
+
+
PV
+
-
Estimated Effect of Disturbances (d_e)

Figure 6. Model-based controller block diagram.

Model State Feedback Control

Model state feedback (MSF) control provides a practical way of implementing model-based controls systems. The MSF algorithm has the advantages of relative ease of design for totally decoupled control, simplicity of its implementation, its flexibility with regard to specifying control system behavior when controls saturate or are lost by equipment malfunction or by being placed into manual operation, and its ability to be tuned either on-line or off-line. The algorithm was developed by to overcome the practical limitations of internal model control application. The limitations center on the inability of the internal model control algorithm to handle control effort constraints, such as a valve being limited to 80% of its full-open range. The MSF algorithm accounts for the actual control efforts applied, rather than determining the control based only on the disturbance estimate and the set point, as is the case with the internal model control algorithm.

Mhatre and Brosilow (n.d.) have shown that in MSF, the control efforts are generated through a linear combination of both past and present model states. This can make the computing power for on-line control effort computation orders of magnitude less than that of other MPC algorithms that use on-line optimization or objective functions to compute the control efforts. Substantial operating efficiency gains are also achievable by allowing for a single algorithm to control all the units of integrated processes, even when units have widely varying time constants. Operational flexibility is maintained with the MSF algorithm because of its ability to allow both control efforts and outputs to be conveniently switched between automatic and manual operation.

Another advantage of the MSF algorithm is that even though it gives exactly the same control efforts as the lead-lag implementation of an internal model control system in the absence of control effort saturation, it does not require explicit inversion of the process model. The MSF structure automatically compensates for past control effort saturation because the model states from which the control effort is computed always reflect the applied control. While the MSF structure automatically compensates for past control effort saturation, it does not automatically compensate for future control effort saturation. A small filter time constant relative to the process time constant can cause the algorithm to fail to compensate for the possibility of future control effort saturation. This can result in highly oscillatory responses. However, with the MSF algorithm it is a relatively simple task to temporarily adjust the MSF filter time constant on-line to bring the controls within the constraint set. Future control efforts can be compensated for by either using an algorithm that projects controls into the future assuming that disturbances remain constant, such as in most MPC algorithms, or by selecting the filter time constant large enough so that future control effort saturation does not cause problems (Brosilow & Joseph, 2002). The derivations of the model state feedback algorithm and the filter time constant calculations are given in Appendix A.

Commercially Available Model Predictive Control Solutions

"It is fair to say that at the process unit level, model-based predictive control is the de facto standard technology for advanced control and optimization. It is effective in the majority of cases" (Dormer, 1998, p. 4). There are several proven MPC products on the market with hundreds of applications worldwide, as noted by Chia and Lefkowitz (1997), Blevins (2001) and Hartman (2002). This section will briefly identify and describe three such products. These particular products were used in this research for comparison with the PLC-based MSF implementation developed for the sugar cooker PLC.

ControlSoft MANTRA®. MANTRA® is designed as a process control software package without any hardware dependence. According to ControlSoft, Inc. (2000), it brings the power of a distributed control system (DCS) to the PC/PLC world and allows end users to easily configure complex control strategies. From constructing, editing, and configuring control loops to monitoring, tuning and optimizing, the software offers all the powerful features associated with a DCS. Among its unique features, the product offers self-tuning PID and model predictive controls. MANTRA® is a complete, modular control system consisting of powerful CPUs, a mix of intelligent analog and digital I/O modules, and bundled engineering and application software which includes control logic programming, HMI, and loop optimization tools. The controller supports both local and remote I/O links, and is peer-to-peer network-ready on Ethernet.

The model predictive features of MANTRA® are based on the internal model control (IMC) algorithm. The IMC programming block is used to control a single process variable by manipulating a single controller output (similar to a PID control block). The IMC controller has an advantage over PID controller when the process has a large deadtime or long time constant. The IMC provides predictive control on an error signal calculated from the process variable and set point values compared with the output of an internal model of the process. The Coordinated Control (CC) programming block uses multiple IMC models to control a single process variable by manipulating as many as three different controller outputs. The goal of the CC block is two-fold. First, the block needs to reject any disturbances to the process. Second, to optimize the three controller outputs for long-term steady-state control (ControlSoft, Inc. 2000).

In implementing IMC control, the control effort is chosen so that the model outputs are forced to follow desired trajectories. The algorithm assumes that there are no control effort constraints, and that the disturbances and set points will remain constant during the current sampling interval. The estimate is updated each sampling interval. While this may potentially cause control effort saturation, the advantage of IMC is the ease with which it can be designed to achieve desired behavior (e.g., non-interaction) and tuned to accommodate anticipated modeling errors.

The desirable features of IMC are that the control designer can tailor the response as needed, and know how the control system will respond in the absence of control effort constraints and modeling errors. When modeling errors do occur, they can be compensated by slowing the control effort response, without changing the process characteristics (e.g., overdamping the response). Further, if the designer has some knowledge of the process variability then tuning methods for the IMC controller can be applied to assure stability and performance. The major disadvantage of IMC is in dealing with control effort constraints. Hitting constraints can cause the standard lead-lag implementation to yield responses that are very sluggish, or have significant overshoot followed by a sluggish return to set point, or exhibit a pseudo inverse response, all depending on the current operating point and the tuning of the controller (Coulibaly, Maiti, & Brosilow 1992). Coulibaly et al. (1992) and Rivera (1999b) give a complete derivation of the IMC algorithm.

The MANTRA® product has been successfully implemented in a wide variety of industries and applications. Within the food and beverage sector the author has seen

MANTRA® applied to complex temperature profile control on extruders and final product moisture control in drying operations. Both systems suffer from deadtime issues as well as interacting variables. The controller successfully overcame these issues.

Universal Dynamics BrainWave®. The BrainWave® controller is an adaptive model-based controller that is designed to outperform PID controllers on deadtime dominant processes or processes with interactions and feed forward variables. The BrainWave® algorithm learns the process response to specified control actions. It then uses a model of this response to predict control actions that will drive the system to achieve and maintain process set points as quickly as possible.

The BrainWave® controller uses discrete time Laguerre functions to recursively determine at each time step a liner model of the process. The on-line Laguerre based identification algorithm has a number of parameters that are predetermined for the designer in order to minimize design efforts. A simple recursive least-squares estimation is used to choose the best fit of the Laguerre network that matches the process dynamic response, and to determine each individual Laguerre orthonormal parameter term. An essential issue is that the Laguerre state space that models the process can reflect only a self-regulating, or stable, dynamic system response (Huzmezan et al., 1999).

The main practical advantage of this methodology is the minimal amount of prior knowledge required to commission a process control loop. Essentially a rough estimate of the time delay, the dominant time constant and the process gain are all that are required to design the control system. This greatly reduces the time for setup and commissioning.

There are several theoretical advantages gained from the use of an orthonormal series representation of process dynamics. The Laguerre model is an output-error structure, is linear in its parameters, and preserves convexity for the identification problem. This allows the use of a simple recursive least-squares algorithm to generate the control output. The use of an orthonormal series representation also effectively eliminates, or greatly reduces parameter drift due to the influence of unmodeled dynamics. The model complexity can be easily changed on-line with minimal disruption to the process. This is a very difficult thing to do with a transfer function implementation. The Laguerre model is stable as long as the unmodeled dynamics are stable. There are also some downsides to using an orthonormal series in adaptive control. One problem is the loss of physical insight to the process. Poles and within some limits, zeros are usually easy to interpret, and are well known to control engineers. However, it is difficult to give a Laguerre spectrum such immediate physical interpretation. Another problem comes from the use of an unstructured model. This issue arises when the frequency content of the excitation signal used during identification phase is incompatible with the choice of the Laguerre pole and the process dynamics. This can result in unwanted artifacts in the identified response (Gough, 2003). A detailed derivation of the BrainWave® algorithm can be found in Huzmezan et al. (1999).

Typical BrainWave® applications in the food and beverage industry include color control of a product during the roasting process, final product moisture control of drying operations, and feed rate balancing and disturbance rejection on a production line. The

BrainWave® controller was able to successfully decouple the process interactions and control for deadtime in these applications.

Pavilion Process Perfecter®. Artificial neural networks (ANN) are noted as being the second best approach to control system design. A controller design based on fundamental physical principles mathematical model is the best method (Helps & Griffen, 1994). ANNs represent a set of powerful mathematical techniques for modeling, control and optimization, in which models learn process behavior directly from process data. The Process Perfecter® controller uses an ANN based model of the process to control the output actuators. Most MPC applications rely on linear dynamic models of the process, even though most processes are non-linear. This approach is acceptable when the process operates at a single set point and the controller is mainly used to reject system disturbances. When the process set point is not static, such as in many chemical and food processing applications, linear MPC systems have difficulty making the transition from one set point to another, resulting in poor control performance. The non-linear ANN model used by Process Perfecter® provides better control over these non-linear regions.

Unmeasured disturbances are common in industrial processes due to external influences that cannot be controlled such as ambient humidity, wear on bearings or product flow rates in adjacent equipment. These disturbances frequently have significant effects on process outputs and controllability. In such cases, process control efforts often cannot be accurately predicted from the independent process input variables alone. To enhance prediction accuracy, a common ANN modeling practice is to include dependent process output variables as inputs to the model. Including these variables almost always benefits prediction accuracy, and is generally acceptable if the model is used only for prediction. However, the process gains necessary for optimization, sensitivity analysis and other process characterizations are almost always incorrect in these models. Process Perfecter® overcomes this issue by using a newly developed ANN called the Focused Attention Neural Network (FANN) to allow steady-state models to obtain accurate predictions and gains in the presence of unmeasured disturbances. The FANN architecture uses dependent process variables as feed forward estimations of unmeasured disturbances. These estimates are used together with the independent variables as inputs to the ANN model. This allows the process gains to be correctly calculated as a function of both the feed forward variables (i.e., disturbances) and the independent variables. An expansive overview of the FANN architecture is found in Keeler, Hartman, and Piche (1996).

The quantity and quality of process data ultimately determines the structure of an ANN model. According to Piche, Sayyar-Rodsari, Johnson, and Gerules (2000), two types of data are readily available in the process industries:

1. Historical data. The values of the inputs and outputs of most processes are saved at regular intervals to a database. Most processing companies retain this historical data associated with their plant for several years.

2. Plant test data. Open-loop plant testing is a well-accepted practice for determining the process dynamics of an MPC application. Most open-loop tests are performed by manipulating a single input, then waiting until the process settles to a steady state. Multiple input moves may improve the quality of the dynamic models.

The Process Perfecter® controller uses an empirical, linear dynamic model developed using the step test data. Historical data is used to build a non-linear steady-state model. An advanced form of gain scheduling is then used to combine the linear dynamic and non-linear steady-state models to compute the MPC control output. Typically the historical data was not used in model development because it was hard to extract and was collected under closed-loop conditions, thus removing the dynamic nature of the process. But relying only on step test data means that only linear dynamic models can be developed. Historical data is typically not useful for developing dynamic models. However, the addition of the historical data allows for non-linear steady-state model development showing the relationship between the inputs and the outputs for varying operating conditions.

Developing process models using ANNs is a very time intensive task. This has the effect of significantly increasing the cost of an installed solution. Because of this high cost, Process Perfecter® is typically not used for regulatory control, but rather is used for process optimization. Additionally, Process Perfecter® has found the most success in the petrochemical industries where there is a very high return on investment due to high margin products. However, Process Perfecter® has been successfully implemented in a variety of other industries as well. In the food and beverage industry the author has used this technology to control and optimize final product moisture in spray dryers and ammonia compressor deployment in large refrigeration plants.

Looking Forward to A PLC-Based Model Predictive Controller

All of the commercially available model-based controllers are PC-based. Because these do not run on the standard industrial control platform (i.e., the PLC) the control system requires additional hardware, software and communications to perform the advanced control functions. Because the PC is used as the platform for the APC algorithms, there are several benefits seen with the direct PLC implementation that are not achievable with these commercial applications. The benefits of the PLC-based solution include:

1. No external hardware (such as a PC) required
2. No additional software required
3. No communications (such as OPC) required
4. Easier to maintain at the plant level
5. Reduced implementation costs

The realization of these benefits will be discussed in detail in Chapter 5.

As previously stated, the PLC is the accepted industrial platform for robust control algorithm execution. Currently the standard control algorithm used in the PLC is the PID controller. When this control algorithm is insufficient for accurately controlling a process, an advanced process control system may be required. The ability to directly replace the PID controller in the PLC with an APC control scheme can provide the benefits listed above. The study outlined in the next chapters implements the model predictive control functionality directly on a PLC with the model state feedback algorithm using a combination of ladder logic and function block programming.

Model-Based Controller Tuning

As with PID control, a model-based controller must be properly tuned to achieve optimal operation. Several techniques have been developed for tuning MPC controllers, based on specific objectives. These can be found in Stryczek (1996); Shridhar and Cooper (1998); and Wojszni, Blevins, Gudaz, and Mehta (2001). In each case, the tuning is achieved by adjusting the gain, deadtime and time constant values appropriately.

Summary

This chapter has briefly discussed some of the key aspects of process control in the food and beverage industry. The ubiquitous nature of the PLC-based implementation of the PID controller was established, along with some of the limitations of the PID algorithm to real industrial processes. Various advanced process control techniques have been developed over the years to address these limitations. One such APC technique is model-based or model predictive control. The MPC solution is based on a mathematical model of the system under test. The model is typically derived from empirical data generated during step tests. One such model-based algorithm is the model state feedback.

Several commercial MPC software packages are available for developing model-based control solutions. These commercial packages have the distinct disadvantage of being PC-based rather than being implemented on a PLC, the industry standard control platform. The application of a model-based controller directly on a PLC would have the advantages of requiring less additional hardware, software and communications protocols. This research looks at the development of the model state feedback algorithm implemented directly on a PLC.

CHAPTER 3

RESEARCH METHODS

Model predictive controllers use a mathematical model of the process to predict the effects of the current control actions. The ability to model where the process is going to be and how the process variables will react to the control actions allows the controller to aggressively move the process to achieve the desired process parameters. This chapter overviews the sugar cooking process used in this study, and the various control algorithms applied. The development of the model used for the predictive controllers will be discussed, as well as the key issues in designing and implementing each control strategy. The methods for collecting the process data will also be outlined.

The design of this study was experimental. The goal of the study was to implement and analyze three commercial and one experimental model predictive control algorithms for a sugar cooking process. To accomplish this, a Hohberger sugar cooker was used as the test system on which the control algorithms were applied. The sugar cooker was controlled by a standard industrial PLC that also served to collect the process data generated during the experiment trials.

The experimental trials followed a procedure whereby the sugar cooker was individually controlled by each of the MPC control paradigms, as well as by a standard PID control algorithm. During each trial, a predefined sequence of process tests was performed. The process temperature set point, the modulating steam valve position, the actual process temperature, and the product feed rate were collected, recorded, and stored for each trial at a one second acquisition rate.

Process Description

The high-boil sugar cooking process starts by mixing an 1,800 pound batch of product consisting of corn syrup, water and selected dry ingredients in a steam-jacketed mixing tank. The product inside the preblend mixing tank is heated to a target temperature of 170°F and contains approximately 12% moisture (see Figure 7).

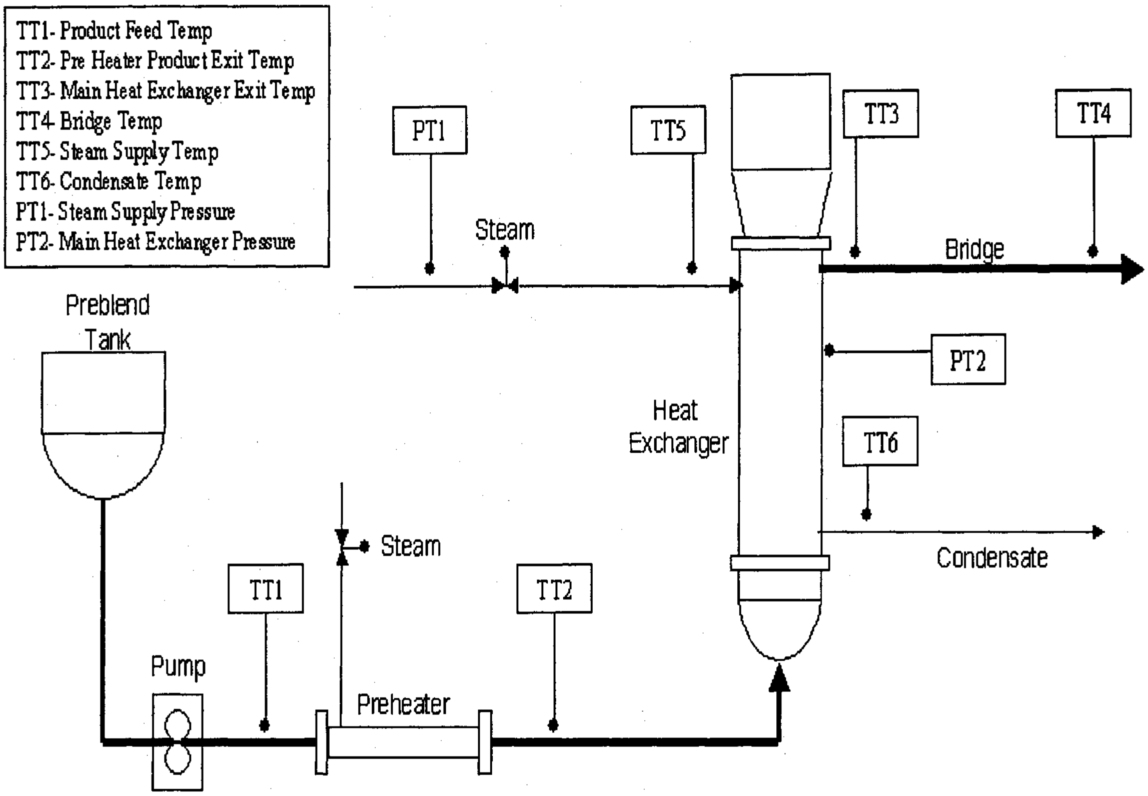


Figure 7. Sugar cooker process flow diagram (reprint of Figure 1).

After the syrup mixture has reached the 170°F target, it is pumped from the steam-jacketed mixing tank through a preheater that raises the product temperature to approximately 220°F (±4°F). The preheater reduces the heat transfer duty of the cooker heat exchanger. The product is then pumped to the Hohberger cooker heat exchanger section. A Waukesha positive displacement pump running at 83 RPM is used to pump the syrup mixture. The product velocity in the tubes is 0.111 ft/sec. At this pump speed 8.5 pounds per minute of syrup mixture is delivered to the Hohberger cooker. The Hohberger cooker consists of two main sections: (a) a heat exchanger is used to elevate the product temperature to the desired set point, and (b) a vacuum chamber is used to assist in removing product moisture.

The heat exchanger is a shell and tube style heat exchanger with a rating of 150 psi at 400°F for both the shell and tube. The heat exchanger is engineered to take product at 220°F and produce 20 pounds per minute of product at a final cook temperature in the 274°F range. For the research trials the 8.5 pounds per minute delivery rate to the heat exchanger section resulted in approximately a 31.5 second resident time as the product traveled through the heat exchanger. A steam line is plumbed to the heat exchanger and serves as the heat media. It is metered to the heat exchanger through a Worcester steam-modulating flow valve. The steam-modulating flow valve is equipped with a positioner that provides valve position feedback with a resolution of 0.1%. By controlling the position of the steam-modulating flow valve, the Hohberger heat exchanger shell pressure can be directly controlled. High-boil product temperature is a function of the heat exchanger shell pressure. As the steam valve opens the shell side of the cooker heat exchanger is pressurized with saturated steam. For a feed pump speed of 83 RPM, experience has shown that a cooker pressure of 78.5 psi (which corresponds to a saturated steam temperature of 323°F) provides enough temperature differential to obtain the necessary heat transfer to reach a cooked sugar temperature of 274°F. When the process is run in the manual mode the operator uses a saturated steam table to determine the steam valve position required to achieve the necessary cook temperature. For a clean heat exchanger and any given feed pump speed there will be a unique cooker pressure (and a corresponding unique saturated steam temperature) to reach a cooked sugar temperature of 274°F.

A rapid start can cause the steam valve to open momentarily to 100%. This leads to a layer of overcooked or brown sugar getting burned onto the exchanger tube walls, which causes fouling. To visualize this, consider the illustration in Figure 8. Three tubes are shown in cross-section. Laminar flow is illustrated in the top tube, with a velocity profile that is zero at the tube wall and maximum in mid tube.

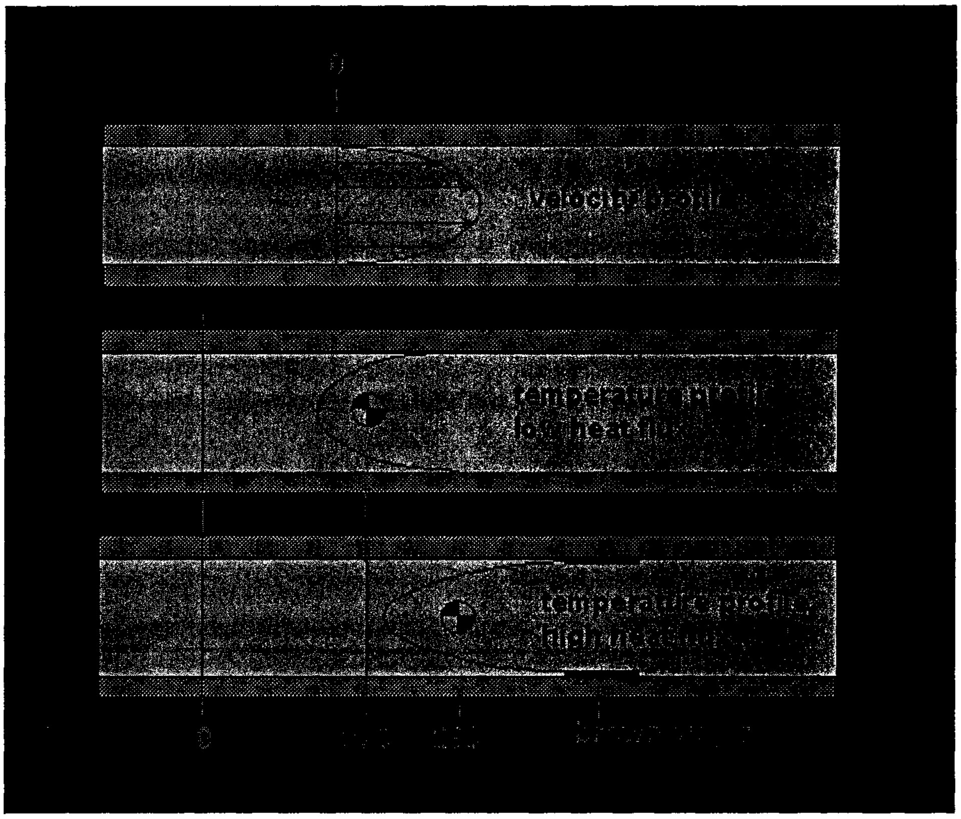


Figure 8. Temperature profiles of the heat exchanger tubes.

A temperature profile with low heat flux is illustrated in the middle tube. This might correspond to an exit temperature of 270°F. Note that there are temperatures above 270°F near the wall.

A temperature profile with high heat flux is illustrated in the bottom tube. This might correspond to an exit temperature of more than 280°F, which is known to cause fouling. Note that now the temperature near the tube wall is much higher than the flow-averaged temperature of 280°F. This causes brown (burned) sugar and fouling. Fouling increases the heat transfer coefficient from the steam to the sugar, requiring higher steam temperatures (i.e., pressures) to achieve 274°F. A minimum startup time of five minutes was set for this application in order to avoid the potential for fouling.

A product temperature RTD with 0.1°F accuracy is positioned 12 inches from the exit end of the heat exchanger and provides feedback on the product final cook temperature. Product exiting the heat exchanger is gravity fed over a weir into an atmospheric separation chamber (shaped like a bowl) positioned along side the heat exchanger section. Mounted directly below this holding chamber is a second chamber almost identical in size and shape to the top holding chamber. An access hole exists between the two chambers and a vertical plug valve is modulated to increase or decrease the amount of product flow, via gravity, from the upper chamber to the lower chamber. The lower chamber is under vacuum in the 6-12 inches of water range. As product drops into the lower chamber, additional moisture is removed with vacuum. The target moisture is 3.2%. The product is gravity fed from the lower chamber to a positive displacement pump that transfers the high-boil product to further processing.

The system steam header pressure was set at a minimum operating point of 110 psi to provide optimum system stability. A lower steam header pressure would not provide enough energy to adequately control the sugar syrup cooking process. The process steam was controlled via a Worcester Pulse Air III modulating valve with encoder position feedback mounted in the cooker steam supply line.

Process Parameters

There are several process parameters that define the control system. The process set point (SP) is the target for the process variable. For the Hohberger cooker in this application the SP was 274°F. The process variable (PV) is the variable being controlled by the system. In this application the PV was temperature in degrees Fahrenheit. The process control variable (CV) is the output from the control system to the actuator. In this application the CV was the steam valve position (open) in percent. The measured process disturbance variables (DV) are the parameters that are measured and included as feed forward variables to the process controller. In this application there is only one DV, the feed rate (measured as pump speed) in RPM. The unmeasured process disturbance variables (dV) are the parameters that are typically not measured by the system controller, and thus not accounted for in the control system. For this particular system one dV is measured for off-line analysis. This dV is the cookers internal steam pressure in pounds per square inch. However, the steam pressure is not accounted for in the control system, and so remains a true dV from a control perspective. Attempts were made to hold all other potential disturbance variables, such as product moisture and operational constraints, constant throughout the trials.

Temperature Control Loop

The primary goal of the sugar cooking process control was to maintain a constant temperature at the bridge RTD (TT4). The temperature recorded at RTD TT4 was regarded as the process variable (PV_T) for the temperature control loop. To obtain the desired cook temperature (temperature set point SP_T), the controller sent a signal to the modulating steam valve positioned in the steam line supplied to the cooker heat exchanger section. Steam valve position was designated as the temperature control loop control variable (CV_T). It was the variable that the controller was manipulating in order to maintain a precise process output relative to the process set point (SP_T).

Pressure Control Loop

An alternate control method for maintaining system cook temperature is to control the system pressure. An electronic pressure transmitter (PT2) mounted in the vent line recorded the heat exchanger internal vessel pressure. The pressure recorded at transmitter PT2 was regarded as the process variable (PV_P) for the pressure control loop. To indirectly obtain a desired cook temperature, a pressure control loop was used to maintain a predetermined pressure set point (SP_P). The pressure control loop generated a signal that was sent to the same modulating steam valve as with the temperature control loop. As with the temperature control loop, steam valve position was designated as the pressure control loop control variable (CV_P).

Research Temperature Control Method

Note that the temperature and pressure control options are mutually exclusive as only one can be in control of the modulating steam valve at any given time. For this application the researcher focused mainly on direct temperature measurement and control rather than controlling temperature via pressure measurement and control[2]. Therefore, throughout the remainder of this document the system variables will be referred to without the temperature subscript (for example CV instead of CV_T). However, for clarity when discussing pressure control, the subscript will be used (for example CV_P).

During operation the process is generally maintained at a constant production feed rate and must quickly react to environmental and product changes that disturb the process control system. The major exception to this occurs when the downstream equipment conditions change due to unexpected stoppages or increased production rate requirements. During these events the Hohberger cooker is required to adjust the production feed rate to compensate and maintain a rate equal to the downstream equipment. Maximum changes in production feed rate of ±25% are possible, though typical feed rate changes are between 5% and 10%.

The main concern for the system operator is to quickly achieve and then maintain a desired temperature set point while minimally overshooting the set point during the start-up ramp. The desired temperature of 274°F is optimal for the boiled sugar as it creates a candy that is clear and flavorful. Any temperature above 285°F is not acceptable as the

[2] The exception to this was Process Perfecter®. That controller used pressure control for starting the process and then switched to temperature control, as described in the appropriate section on page 89.

candy turns dark brown and tastes burned. Temperatures below 268°F lead to a cloudy candy without satisfactory taste. Additionally, the final moisture specification can be too high and product quality compromises such as texture variations due to insufficient caramelization may occur. The best range for temperature is between 273°F and 275°F. Maintaining this two-degree range had proven difficult for the existing control system operation and had caused the test duration to sometimes be shortened due to a lack of in-specification sugar. Finally, if the product temperature rises too quickly, the product may burn onto the inside of the tubes, causing the heat exchanger to lose its heat transfer characteristics. Temperature limits were set at ±5°F deviation from set point.

Controller Implementation

Figure 9 shows the overall system block diagram for the sugar cooking process. The system is comprised of a PLC; an operator interface; an industrial PC used for MPC control, programming, and data acquisition; and the sugar cooker as depicted in Figure 7. The various communication protocols used throughout the system are also shown.

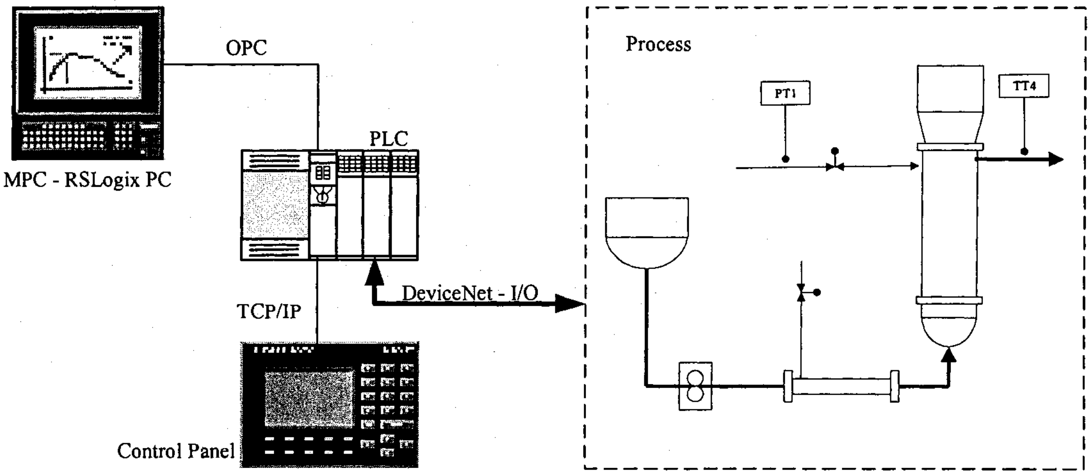


Figure 9. Sugar cooking process system diagram.

Controller Interface

Each PC-based process controller was required to interface with the existing PLC-based control system according to the following criteria:

1. Communications were to be established to interface with the Allen-Bradley ControlLogix® PLC using Allen-Bradley's RSLinx® OPC server. An Allen Bradley 6181 industrial PC equipped with the Windows NT® 4.0 (Service pack 3) operating system was provided to run the APC application software. The OPC server was configured for a 5 second update interval. Table 2 lists the control and instrumentation tags configured on the OPC server.

Table 2

OPC Group and Item Structure

Group	Group Type	APC Tags	PLC Tags
HOH_Input	Input, Polling	PLC Watchdog	PLC_Watchdog_Timer.ACC
		Manual Mode Status	N22_PV_STATUS[0].10
		Mode Status	APC_Control_ON
		Set Point	APC_TEMP_SP_FINAL
		Process Variable	APC_Hoh_Bridge_TEMP_TT4
		Control Variable Tieback	APC_Steam_Valve_Pos_Actual
		Feed Forward Variable	APC_Feed_Pump_RPM_CMD
		Disturbance Variable	APC_Vessel_Steam_PRES_PT2
HOH_Output	Output, Notify	Watchdog	APC_Heart_Beat
		Control Variable	APC_Valve_Position_CMD

2. An existing Allen-Bradley PanelView® 1000 touchscreen control panel was used to enable the interface between the operator and the process/controller. The PanelView® 1000 communicated with the ControlLogix® PLC Ethernet port using TCP/IP protocol. The operator entered the following parameters and commands from the control panel:

- The process set point (final product temperature of 274°F).
- The selected control mode (PID, MPC, or Manual) for the product temperature loop in the PLC logic.

Each MPC controller supplied the following interface with the existing PLC-based control system:

1. The MPC controller provided to the PLC the following control signals:

- The command output signal for the heat exchanger's steam supply modulating valve.
- A heartbeat pulse. The heartbeat was a pulse train with a 50% duty cycle and a 10 second period. It was generated and set to the PLC to allow the PLC to verify communication with the MPC controller. If the heartbeat pulse was lost the PLC automatically switched to PLC-based PID control.

2. The PLC provided to the MPC controller the following essential functionality to provide graceful startup and shutdown of the MPC controller:

- The process set point (final product temperature of 274°F).

- A request for MPC control.
- Bumpless transfer and set point tracking. PLC programming ensured proper (bumpless) switching between control modes. If MPC was switched on, the currently measured value for a PV became the new set point for the controller. This meant that the current process situation was frozen, and that no sudden automatic changes would happen. It was then up to the Operator to put in new set points, if needed. When MPC was switched off, the last position of the CV was be copied into the existing controller, effectively keeping the process where it was (set point tracking).

Test Method and Analysis Criteria

All controller trials on the sugar cooker were conducted with the technical engineering purpose of evaluating the overall accuracy of the model-based controllers used in this research for advanced regulatory control. The overall responses of quickly achieving product set point after start-up, and maintaining product temperature at a desired set point after the introduction of a product feed rate disturbance were main technical evaluation parameters for the study. Each potential solution was evaluated based on a set of fixed criteria, as outlined below.

To objectively evaluate controller performance the following objectives were established and measurable parameters were recorded:

1. Before the controller was initiated, the process was allowed to stabilize. The steam valve was moved to 30% open and TT4 was monitored until the temperature was approximately 220°F.

2. After PID/MPC control had been requested, the controller automatically ramped up the product temperature to 274°F. The goal was to do this in less than 15 minutes, but not less than five minutes. The actual response time for each trial was recorded.

3. The process was then held at the steady-state condition for approximately ten minutes. During this time the controller was evaluated on its ability to maintain a steady-state discharge product temperature at the bridge (TT4) of 274°F (±1°F on average). A constant product feed rate was assumed.

4. Disturbance model tests were performed once the controller showed that it could hold the process at the steady-state condition. The controller was required to maintain a discharge product temperature at the bridge (TT4) of 274°F (±1°F on average, with a maximum deviation of ±5°F) while the product feed rate was varied by approximately ±25% in a prescribed test pattern. System disturbances were introduced by altering the product feed rate through the heat exchanger. The feed rate was manually increased in two approximately 12% intervals, to 93 RPM and 103 RPM. At each interval the process was allowed to stabilize and then evaluated on its ability to maintain the process variable at the set point. The feed rate was then returned to the original speed and the process was allowed to stabilize before it was manually decreased in two

approximately 12% intervals, to 73 RPM and 63 RPM. At each interval the process was again allowed to stabilize, then evaluated on its ability to maintain the process variable at the set point. Finally, the feed rate was returned to the original speed and the process again stabilized and evaluated.

5. The controller was evaluated on its ability to demonstrate consistent start-up without fouling.

PID control response was recorded and used as a baseline for comparison against all four model-based controllers. The data from the individual trials was recorded using the standard on-line data logging functions in Allen-Bradley's RSLogix® 5000 programming software. The data analysis will be presented in Chapter 4.

PLC-Based PID Temperature Control

The first method employed to control the sugar cooking temperature was a standard PID (Proportional, Integral, Derivative) instruction in the PLC ladder logic code. The cooker and associated equipment were ultimately controlled using Allen-Bradley's ControlLogix® hardware platform with communications to distributed field I/O devices over a DeviceNet® network. The DeviceNet® network communicated with Allen-Bradley 1794 DeviceNet® distributed field I/O and served as the link to the Hohberger cooker analog and digital I/O points. The central ControlLogix® hardware platform also included an Ethernet/IP network link to communicate to higher-level computer-based systems. PLC ladder logic code for the Logix® 5550 processor was written using Allen-Bradley's RSLogix® 5000 software (version 10.00). DeviceNet® network configuration was established using Allen-Bradley's RSNetWorx® for DeviceNet® configuration software (version 4.01).

The PID instruction utilized in the PLC, like any traditional PID controller, has the limited capability of monitoring only one PV and controlling only one output or CV at a time. The PID instruction continuously calculates the process error based on the present value of the system process variable relative to the SP. Additionally, the PID instruction has no practical way of interpreting a long process deadtime that may be naturally inherent in the system. As a result, the process error continues to increase during the deadtime, which in turn causes the PID to continue to drive the CV well beyond what is required to reach the SP once the deadtime has passed. For the Hohberger cooking application this eventually lead to a severe temperature overshoot condition. To limit the amount of inevitable overshoot, an upper limit constraint was configured in the PID instruction logic. Additionally, the PID instruction cannot successfully predict how a randomly occurring measured (or unmeasured) disturbance will affect a future system PV value.

The ControlLogix® used the parallel PID controller configuration. The controller was setup for direct acting, dependent control with no biasing and a 0.25 second update frequency. The integral term was defined in repeats per second and the derivative term was defined in minutes. In order to avoid over cooking the product, the PID controller output was limited to a maximum of 55% open. It was also found that a controller output less than 30% open could not further cool the product due to the input from the preheater that held the product at 220°F. Therefore the PID controller output was clamped at a minimum of 30% open.

A ramp phase was programmed into the PID controller for implementation during the initial temperature rise to set point. The ramp function was used to prevent steam pressure saturation which caused the steam-modulating valve to fully open. A fully open valve would have added too much energy to the cooker and caused the temperature to rise too rapidly resulting in a loss of control. The ramp function increased the temperature set point in 4°F increments from 230°F (the temperature at the preheater) up to the set point. When the temperature of the batch was stable for 15 seconds, the next incremental increase was implemented.

The overall cooking process displayed a non-linear response in heating the sugar product from its initial temperature to its final temperature. A 5% steam valve change produced a different degree of temperature change depending on what temperature range of the process was being examined. An attempt was made to reduce the effect of the system non-linear gain by writing a basic gain schedule in the PLC code. This amounted to using the current PV value to decide what gain values the PID instruction would use to calculate the CV move. As the process variable increased to different predefined stages, a new set point and set of proportional, integral, and derivative parameters were written to PID gain registers in the PLC PID instruction. For a valve position between 30% and 40% moderate acting PID parameters were used to bring the temperature to within approximately 4°F of the set point. Slow acting PID parameters were used for a valve position between 40% and 50% to conservatively bring the temperature the rest of the way up to the set point with minimum overshoot. Fast acting PID parameters were then used to maintain the set point during steady-state operation.

PID Loop Tuning

A key element for any PID controller is proper tuning. Well-tuned PID loop control results in tighter tolerances in quality, reduced scrap and reduced downtime. Without the correct proportional, integral and derivative gains the controller will be unstable and will not be able to meet the control objective. There are several methods for acquiring correct PID parameters. It was not the purpose of this research to investigate these methods. Rather, ControlSoft's INTUNE® PID tuning software package was used to calculate an effective value for proportional, integral and derivative gain used in the PID instruction. INTUNE® is a collection of software tools that are used to start-up, diagnose and maintain the health of PID loops.

The initial, tuned PID controller results on the Hohberger cooker showed that PID control was not acceptable for driving the process to the temperature set point within the specified time and maintaining an overshoot limit of 5°F. It took about 25 minutes to stabilize the temperature within ±2°F of SP. Disturbances were generated in order to test PID control. The feed rate was increased to 93 RPM then dropped to 73 RPM. The increase of feed rate caused a temperature drop to 269°F (from 274°F). The drop in feed rate caused a large temperature increase and as the temperature reached and passed 281°F, the system was forced to shut down to avoid fouling the cooker. From this result, it was clear that PID controller could not compensate for the disturbances. It was, therefore, determined that advanced process control techniques were required to maintain the specified control of the Hohberger cooker throughout its range of operating conditions.

It should be noted that prior to any advanced process control investigation, proper PID tuning must be established to determine whether the physical system is able to be controlled using standard PID methods. Advanced control techniques should only be implemented when standard control techniques (such as PID) cannot perform acceptably. Using advanced control techniques when standard control techniques are sufficient incurs unwarranted additional system design and implementation costs.

Model-Based Temperature Control

The remaining software control platforms used in this research study were developed using model predictive control principles. The process model is based on differential equations that represent the dynamic relationship between the control actions and the resulting process variable responses. The first step in developing an accurate process model is to generate the model parameters. The mathematical structure of the model and the model parameters are obtained by performing a series of off-line step tests to the physical process. This is typically accomplished by manually changing the controller output (CV) while observing the response of the PV and recording the process deadtime, time constant and gain. Tuning is then achieved by adjusting these model parameters on-line.

A well-tuned MPC application will make adjustments to the process similar to those of an experienced operator – a few moves initially followed by a wait period with little or no further moves. If there were no disturbances acting on the process the differential equations would exactly predict the process response to the controlled moves, and could be resolved to determine the exact control moves required to achieve the process goal. However, due to naturally occurring disturbances there will always be some error between the set point and the process variable. Any error in the process state is corrected by feedback from the process variable measurements each control interval. This compounded feedback, from the prediction model and from the process measurements, is part of what makes model predictive control so robust. The result is that the dynamic models need not be infinitely precise for the controller to function well.

The first set of step tests was performed to discover the relationship between the product temperature and steam valve position. This involved manually commanding the steam valve position to move a specified percentage of its full range while measuring the cooker temperature response. For the trials a steam valve change was performed and the change in the sugar temperature at RTD TT4 was recorded. Figure 10 shows the results of these step tests. These resulted in an average process gain of approximately 3, an average process deadtime of approximately 30 to 40 seconds, and a process time constant of approximately 100 to 120 seconds.

The second set of step tests determined the relationship between the product temperature and the feed forward disturbance expressed as a feed rate in RPM. For the trials this disturbance was achieved by varying the speed of the Waukesha feed pump that pumped product to the inlet of the heat exchanger section. By varying feed pump speed, the change in product feed rate simulated a random system disturbance. As seen in Figure 11, this resulted in a gain of approximately –0.8, with a deadtime for the feed forward model of approximately 50 seconds, and a time constant of approximately 75 seconds.

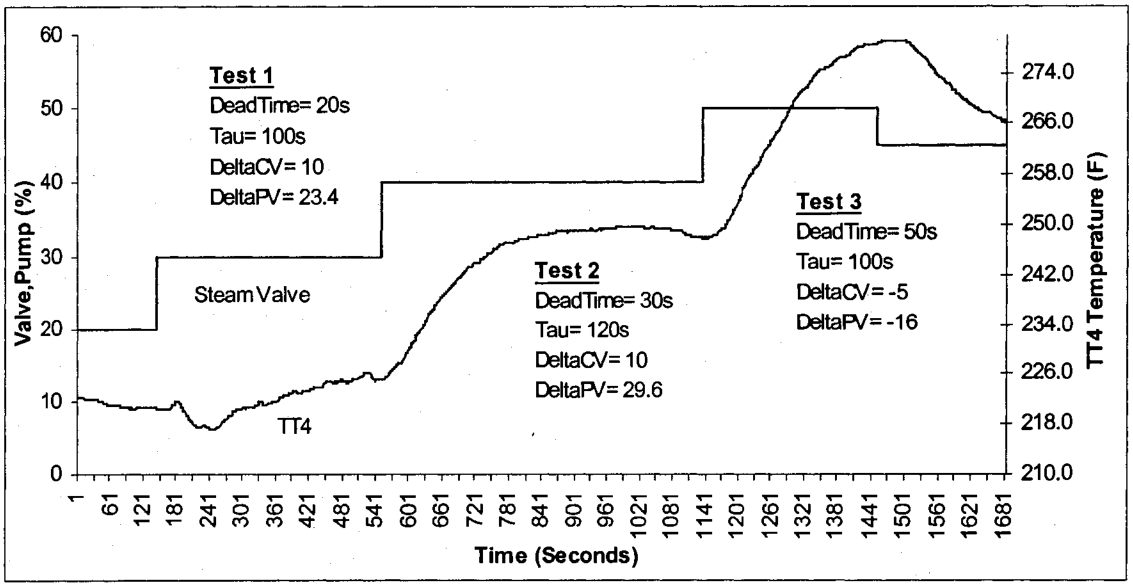


Figure 10. Feedback model parameter bump tests.

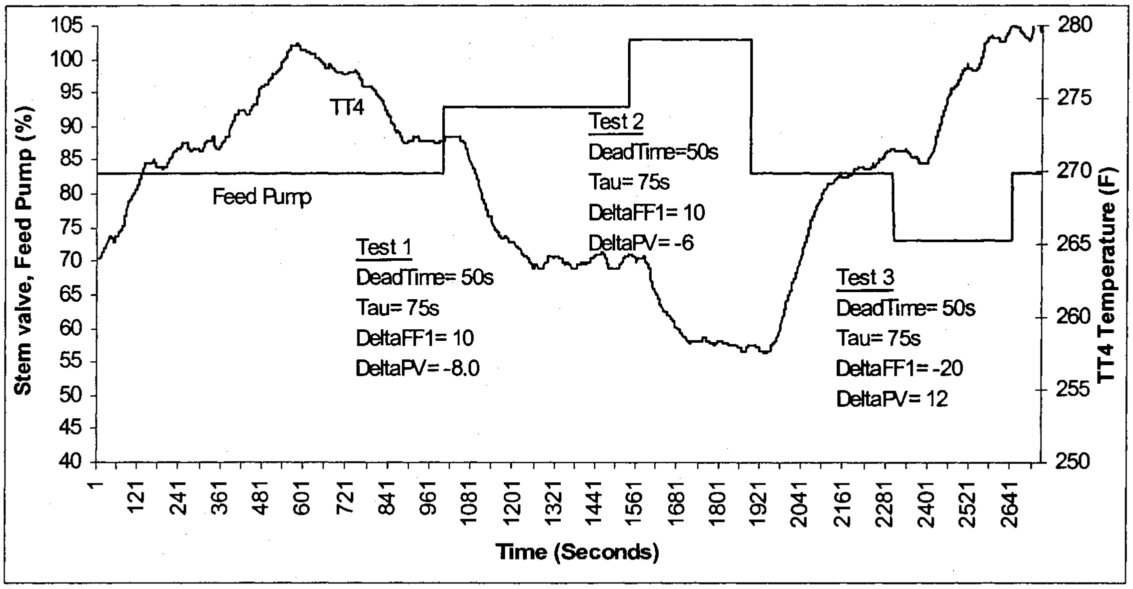


Figure 11. Feed forward model parameter bump tests.

These step tests were performed at the beginning of the MPC trials and were used in developing each of the various control strategies. However, there is no global design solution that exists for model predictive controllers. Each controller must be customized to achieve the desired behavior based on the characteristics of both the process and the applied MPC algorithm. There are several performance issues that need to be considered for each application. These include how aggressively and how long the controller should work to eliminate errors. An aggressive controller can rapidly minimize process errors, but requires dramatic control actions to do so. This can rapidly wear out or even damage control actuators. A less aggressive controller spares the actuator at the expense of rapidly eliminating the process errors. For the sugar cooking process the initial temperature rise time was limited to a minimum of five minutes. A more aggressive temperature increase would cause the product to burn and be unusable.

The bridge temperature was directly affected by changing steam pressure and mass flow of syrup through the system. The syrup feed pump speed was a good indication of the syrup mass flow through the sugar cooker. Using the data from the step tests of the sugar cooker, the transfer functions in Equations 6 and 7 were identified to model the process. These transfer functions represented all model information available to design, tune and integrate the controllers.

$$T_{SP} = \frac{2.9}{(100s+1)} e^{-25s} P \qquad (6)$$

$$T_{FF} = \frac{-0.8}{(75s+1)} e^{-40s} PS \qquad (7)$$

where,

T_{SP}:	set point temperature control
T_{FF}:	feed forward disturbance temperature control
P:	steam pressure
PS:	pump speed
s:	Laplace operator

MANTRA® Temperature Control

After the manually initiated step response tests were conducted on the sugar cooker system, MANTRA® Application Developer was used to program an advanced system controller for the sugar cooking process. Numerous input blocks were programmed to bring in all pertinent system variables such as the process variable (bridge temperature at TT4) and the product feed rate derived from the feed pump speed. A Ramp function block was programmed to maintain temperature control and reduce the possibility of temperature overshoot. The Ramp function block varied the heat exchanger steam valve position (system control variable output) by 2% every three minutes. The Ramp function remained in control until TT4 reached 260°F, or until the steam valve position achieved 50% open for a total of five consecutive minutes. At the completion of the Ramp function, the control automatically switched over to the Coordinated Control function block. Figure 12 shows the block diagram for the MANTRA® application.

For the sugar cooker trials, the Coordinated Control function block was chosen as the focal point of the control strategy. The Coordinated Control function block is used to control a single process variable using multiple controller outputs in automatic mode based on the PV - SP deviation, internal models and tuning. The controller used a first order lag with deadtime internal process model and first order filters (total of up to 12 tuning parameters) to calculate the controlled output. Each controlled output is calculated such that the process variable follows a first order lag trajectory when approaching the set point value. In the case of controlling sugar temperature, only one of the controller outputs was used. It was linked to the heat exchanger steam supply modulating valve.

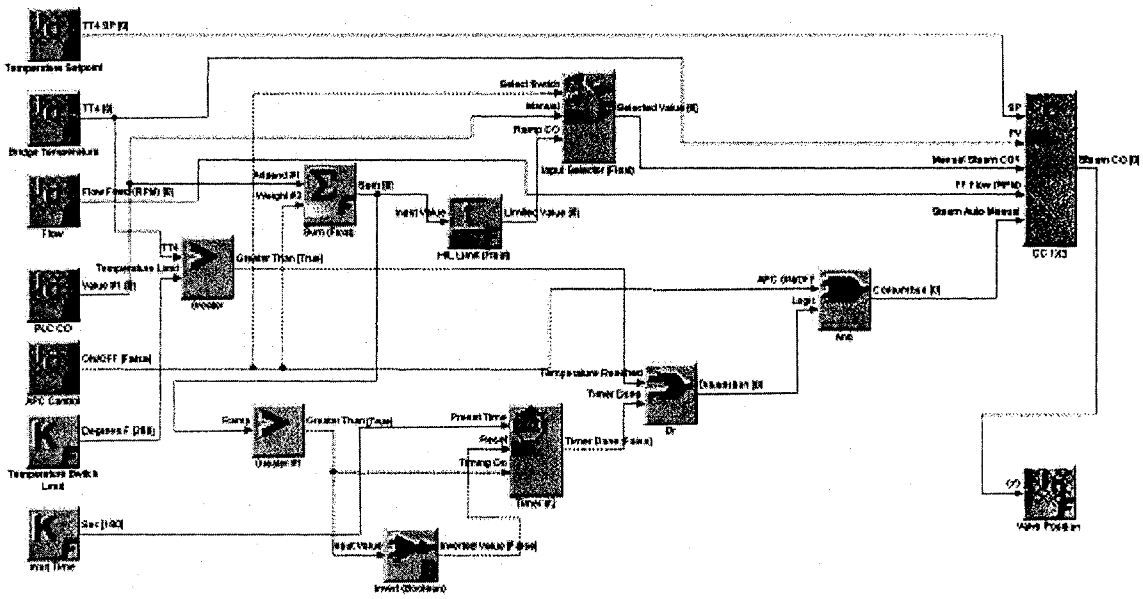


Figure 12. MANTRA® controller block diagram.

The Coordinated Control function block required measured tuning values to be entered in the configuration table. These parameter values were the overall model gain, model deadtime, and model time constant. The previously recorded step response fit the curve of a first order system with an approximate process gain of 3.3, a deadtime value of 30 seconds and a time constant value of 180 seconds. The overall system filter time constant was set at 50 seconds. The filter time constant determined the speed of the controller action.

The Coordinated Control function block has the capability to monitor a random system disturbance and factor this value into the control algorithm output. The Coordinated Control function block converts a system disturbance to an overall feed forward gain. The feed forward gain is also defined by the deadtime, time constant and gain due to a step command in the feed rate. A feed forwarded value for deadtime of 80 seconds, a time constant of 100 seconds and an overall process gain of –0.45 were used based on the previously outlined step test results.

After configuring the MANTRA® controller, the system was placed on-line. A total of five trials were conducted and the data recorded. For specific details on the MANTRA® controller functions and implementation see ControlSoft, Inc. (2000).

BrainWave® Temperature Control

For the sugar cooker trials, the BrainWave® software package was loaded on to the same PC on which the MANTRA® controller had been utilized. Figure 13 shows the general block diagram for the BrainWave® controller. Regarding integration, the BrainWave® controller does not have an associated software language protocol, so it did not have to be programmed to perform its control function. Instead of programming, there were three configuration steps that were required to place the controller online. In step one, OPC communication links were established to allow data transfer between the PLC processor and the BrainWave® controller. The controller received status inputs from the system through the PLC for real system parameters such as the bridge product temperature TT4 (process variable PV) and the product feed rate derived from the product feed pump speed (feed forward variable FF1[3]). The controller CV output signal was mapped to the PLC output register for the steam-modulating valve. Additional links were configured as watchdog timers to warn the PLC if the controller inadvertently shutdown.

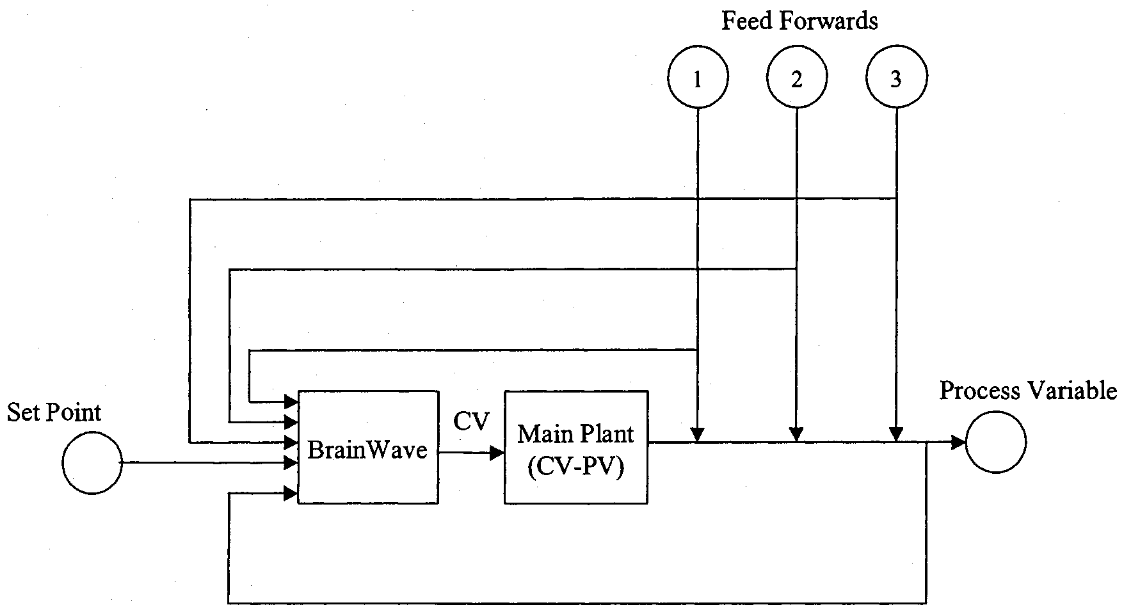


Figure 13. BrainWave® controller block diagram.

Step two consisted of entering scaling values in a data conditioning dialog screen. In this screen data formats were established for all of the tags configured during step one. Each tag's description was entered along with its raw value range, engineering value range and filtering criteria.

Step three required configuring parameter values for deadtime, time constant and process gain for both the steady-state model and the feed forward disturbance model.

[3] The loop controller had the capability to monitor two additional feed forward variables, but in the sugar cooking trials these two feed forward variables were not used.

These values were calculated from the manual step response curves generated in previous trials. For the main steady-state model the initial estimated values were 45 seconds for deadtime, 120 seconds for the time constant, and an approximate process gain of 2.6. For the feed forward model the initial estimated values were 50 seconds for deadtime, 75 seconds for the time constant, and an approximate process gain of –0.65. Based on the entered parameter values, BrainWave® automatically chose a model from a predefined group of over a hundred models to best control the process. For specific details on the BrainWave® controller and its implementation see Universal Dynamics Technologies Inc. (2000).

BrainWave® was then placed on-line to control the process. A total of five commissioning trials were conducted and the data from the trials were recorded. During the commissioning trials, the controller successfully predicted the position for the steam valve in order to reach the desired set point of 274°F for the bridge temperature (TT4). Steam valve positioning was based entirely on the selected model and did not require a ramp function, timing function, or control based indirectly on pressure feedback from the heat exchanger.

Process Perfecter® Temperature Control

The final commercial controller implemented was Process Perfecter® from Pavilion Technologies. It was also loaded on to the same PC as the previous software packages. As with the other MPC controllers, the step response models in Process Perfecter® include the process gain, the time delay before the PV responds, and a dynamic representation of how fast the PV moves to the next steady-state.

The overall Pavilion control strategy was divided into two distinct components. First, a pressure control model assisted by a preset timing function was used to control the initial increase of the sugar product temperature. When the preset six-minute timer expired, system control automatically switched from the pressure control model to a temperature control model. The temperature control dynamic model accounted for disturbance changes in the system such as the feed rate change produced by varying the speed of the product feed pump.

The logic for controlling the preset timing function and toggling between pressure and temperature control is administered through a Runtime Application Engine® (RAE®). RAE® applications are used as real-time moderators that can implement logic, equations or different models. Specifically in the cooker project, two RAE® applications enabled a one-button start of the sugar cooker process. Process Perfecter® gets instruction from one of the RAE® applications and starts up on pressure control with a desired pressure set point target. After a specified wait time for pressure to reach target, the other RAE® application triggers Process Perfecter® to switch to temperature control with a desired target. Tuning is also changed for temperature control at this point.

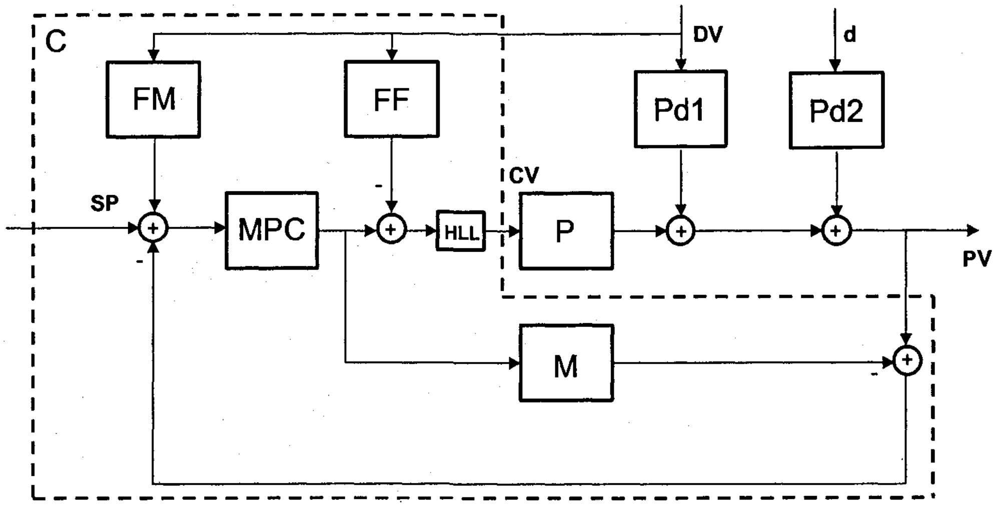
During the plant tests a nonlinear behavior was evident for the responses of pressure and temperature to CV changes. A nonlinear gain model using an artificial neural network was built from the data from those tests. Process Perfecter® is the only one of the MPC solutions with the ability to directly implement a non-linear model. The responses to changes in feed rate were noted to be linear and were implemented as a constant gain model.

Five trials were conducted using the Process Perfecter® controller. The data was recorded for off-line analysis with the other MPC solutions. For specifics on controller design using Process Perfecter® see Pavilion Technologies, Inc. (2002).

Model State Feedback Temperature Control

The main goal of the model state feedback controller was to demonstrate the capabilities of the ControlLogix® PLC to control temperature in the sugar cooker using the standard function block library. The focus was on linear model predictive feedback and feed forward control. The implementation consisted of two function block diagrams, one main ladder code routine, and input and output ladder routines. The function block diagrams provided the control output calculations in automatic and manual (tracking) modes. The main ladder routine addressed internal parameter calculations, initializations, bumpless transfer and standard interface to the controller configuration parameters. The input and output routines provided interface to the process variables. See Appendix B for ladder logic routines and Appendix C for function block routines.

The function block diagram of the controller implementation is shown on Figure 14, and the control parameters are given in Table 3. The functionality of the controller is similar to a Smith Predictor or Internal Model Controller with model predictive feed forward compensation.



where,

MPC:	model predictive process controller
P:	physical plant process to be controlled
M:	mathematical process model
FM:	compensator
FF:	feed forward
HLL:	high-low limiter
Pd1, Pd2:	process disturbance lags

Figure 14. Model state feedback controller block diagram.

Table 3

MSF Controller Parameters

Parameter	Value	Units
Feedback Gain	2.9	°F / %
Feedback Time Constant	100	Seconds
Feedback Deadtime	25	Seconds
Feed Forward Gain	-0.8	°F / RPM
Feed Forward Time Constant	75	Seconds
Feed Forward Deadtime	40	Seconds
Filter Time Constant	85	Seconds
CV Minimum	0	Percent
CV Maximum	100	Percent
CV Rate of Change Limit	100	% / Second

The advanced control routines were programmed as subroutines in the ControlLogix® program and were set up as a periodic task. Periodic tasks in ControlLogix® execute program subroutines deterministically. Deterministic execution of a program assures that the elapsed time between two subsequent executions of the program is kept constant by the processor. Deterministic execution is assumed in the implementation and must be assured when the program is executed so that the deadtime array timing is maintained. One-second controller sampling time was selected for the sugar cooker controller. The model predictive control program may be copied or duplicated to control as many loops as needed within a single ControlLogix® processor so long as deterministic execution is assured, i.e., without overloading the processor.

After the control algorithm was developed, five commissioning trials were made. As with the other solutions, the process was driven to 274°F, and the response time and overshoot were recorded. The controller was then evaluated on its ability to maintain the process at set point during feed rate disturbances.

Summary

This chapter presented a detailed description of the sugar cooking process, including the physical constraints and operational considerations that must be observed in order to produce an acceptable product. Traditional PID control of this process has proven inadequate based on production experience and analytical tests. Advanced process control techniques using a model of the sugar cooking system as the basis for control should be able to provide better control, in terms of both set point tracking and disturbance rejection, than PID control. Three commercial, PC-based MPC packages and one new PLC-based MPC solution were separately integrated into the sugar cooking process and tested for accuracy of the control objectives.

The MPC solutions were designed using actual process parameters obtained from open-loop step tests. The solutions were then integrated using standard communication protocols to send and receive data from the sugar cooking PLC. Necessary system information was tracked to provide a smooth transition between control modes and to ensure that the control algorithms were properly operating. In addition, an appropriate operator interface was developed as an add-on screen to the existing control panel.

A prescribed set of process tests were performed with each controller wherein the process was rapidly brought up to temperature from a known start condition in order to assess the set point tracking capability of each control method. A series of disturbance tests were then initiated with each controller to determine the controller's ability to reject disturbances and maintain the process set point. The data from these tests were collected using the on-line data logging functionality of the PLC programming software. The results of the process tests for each controller system and the analysis are presented in Chapter 4.

CHAPTER 4

PRESENTATION AND ANALYSIS OF DATA

This chapter presents the results of the research on applying model state feedback advanced control techniques directly on a ControlLogix® PLC to control an industrial sugar cooking process. The model state feedback MPC method was compared to the results of four industry accepted control methods applied to the same sugar cooking process. These methods included standard PLC-based PID control and three PC-based, commercially available MPC controllers. The PC-based controllers were ControlSoft's MANTRA® controller, Universal Dynamics' BrainWave® controller, and Pavilion Technologies' Process Perfecter®.

The purpose of this study was to determine the viability of applying MPC techniques directly on an industrial PLC. Six research hypotheses were developed as a result. The research hypotheses were:

1. It takes significantly less time for the PLC-based model state feedback implementation of the MPC controller to reach the final product temperature set point than it does for standard PLC-based PID control applied to the same industrial sugar cooker.

2. The PLC-based model state feedback implementation of the MPC controller experiences less temperature overshoot due to the initial product temperature rise than the standard PID control solution.

3. There is a smaller deviation in temperature around the set point, in the presence of system disturbances, during steady-state operation for the PLC-based model

state feedback implementation of the MPC controller than there is for the standard PID control solution.

4. The temperature rise time is shorter for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions applied to the industrial sugar cooker.

5. The PLC-based model state feedback implementation of the MPC controller exhibits less temperature overshoot as a result of the initial product temperature rise than the PC-based commercial MPC solutions.

6. The deviation in temperature around the set point, in the presence of system disturbances, is smaller for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions.

To summarize, these hypotheses stated that the MSF control implementation would perform better than the other control methods in regards to temperature rise time, initial temperature overshoot, and disturbance rejection during steady-state operation. Various statistics were used to test these hypotheses, as delineated in this chapter.

The standard PID control response was recorded and used as a baseline for comparison against all four model-based controllers. As expected, all four model-based controllers performed to specification. The standard PID control did not. There were two main areas on which the overall comparative analysis focused. These were the dynamic response of each strategy at start-up (temperature rise time and initial temperature overshoot), and the steady-state disturbance rejection capabilities of each strategy based on process feed rate changes. The comparison results are discussed in detail below.

Method of Data Collection and Analysis

The production run test set for each control strategy consisted of one trial set point tracking start-up run and four complete trial runs. Each complete trial run covered approximately two hours and included both dynamic set point tracking and disturbance rejection during steady-state conditions. Five different control strategies were investigated. The test data were thus collected from 25 trial production runs at one-second intervals.

Based on the production trials, summary statistics were calculated for rise time and overshoot during set point tracking, and for settling time and temperature deviation during disturbance rejection tests for each of the control strategies. The integrated squared error and normal error distribution were also determined for each control strategy. Reference Appendix D for the tabulated production trial data.

The research hypotheses were tested using analysis of variance (ANOVA). Tukey's post hoc test for honestly significant difference comparisons and the summary statistics were used to explore the differences resulting from the ANOVA. A one-way ANOVA was used to analyze the dynamic response of temperature rise time for the control strategies. A second one-way ANOVA analyzed the temperature overshoot for the five control strategies. A two-way ANOVA was used to analyze steady-state disturbance rejection based on a preset pattern of feed rate changes for each control strategy.

Analysis of the Collected Data

One of the most important aspects of process analysis is to have a clear understanding of what is to be achieved and how it will be measured. The goal of regulatory control is to maintain a controlled parameter at a specified set point. The performance criteria are centered on measuring the dynamic responses of the controlled parameter while attempting to achieve the set point, and the steady-state response while attempting to maintain the set point. According to Universal Dynamics Technologies Inc. (1998b), the most important criterion for measuring steady-state performance is offset from set point. In measuring dynamic performance there are three common criteria, these are peak error, rise time and settling time. The peak error is the maximum deviation from set point due to a transient response. Rise time is how long it takes the process to reach the set point the first time. Settling time is the amount of time required for the process variable to settle within a specific dead band around the set point after a process disturbance or change.

Another important performance criterion is integrated squared error (ISE). ISE is the square of the total net deviation from set point, and is calculated by integrating the square of the instantaneous error between the set point and the process variable (see Equation 8). The algorithm weights large errors more heavily than small errors, but does not differentiate between negative and positive errors. Unlike algorithms that average offset errors, the ISE returns a large value for oscillating loops. Dividing the total ISE by the total number of error samples normalizes the ISE. Normalized ISE was calculated and reported in this research.

$$\mathrm{ISE} = \int [e(t)]^2 \, dt \tag{8}$$

Performance of Final PID Control

The PID control strategy did not meet the control specification for both the temperature set point tracking phase as the product temperature was initially ramped up to the set point, and the disturbance rejection phase as the product feed rate was changed. Figure 15 shows a typical production run using the PID controller. The system showed unacceptable overshoot and oscillation. Over the five trials the controller achieved an average start up time of 28.75 minutes and recorded a maximum initial overshoot (due to temperature ramping) of 11°F with an average overshoot of 8.6°F. Due to the limited run time available for each trial, the PID controller was only able to run in a steady-state condition for approximately five minutes. However, during this time the controller did maintain control of the process to within ±1°F as required.

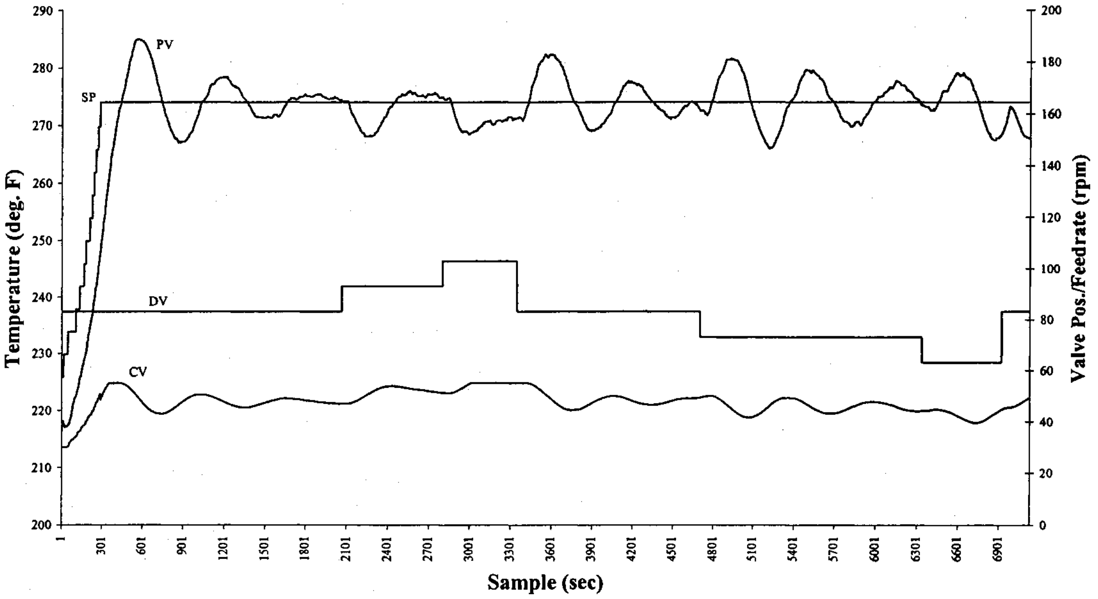


Figure 15. Standard PID system response.

The PID control strategy also did not meet the specified control response to the feed rate disturbances. The disturbance response to product temperature resulting from the product feed rate showed an average settling time of 5.8 minutes with an average deviation of 2.8°F and a maximum deviation of 8.3°F. The normalized integrated squared error was 12.8. Due to the limited run time available for each trial, the feed rate changes often had to be made before the system fully stabilized and returned to steady-state conditions. This was especially true when the feed rate was decreased. As a result, the data may be skewed favorably for the PID control strategy. The posted response time would have been greater had the run lengths afforded the necessary time for the system to fully stabilize. Additionally, the integrated squared error would have been higher. A visual inspection of the production run data shows that indeed the controller was not capable of controlling the system within the specified parameters.

As the product feed rate was decreased below the nominal feed rate, the disturbance rejection control during steady-state operation showed that the PID controller started to experience instability. Instability occurs when the controller loses the ability to reject a disturbance or maintain control of the process at the set point. As a result, the process variable oscillations continue to increase to an unacceptable level. For the Hohberger cooker, a decrease in feed pump speed resulted in a different disturbance process gain. This is due to the non-linear heating dynamics of the cooker. The PID controller could not compensate for the differing process gain between increasing and decreasing the product feed rate. The PID controller was tuned to compensate for product feed rate increases. As a result, the process never fully stabilized when the product feed rate was decreased. Additionally, the operator was unable to let the process run for an extended period of time at the lowest feed rate because the initial process variable oscillations were too large at that speed. Had the process been allowed to continue running in that condition, the product temperature would have exceeded acceptable limits and caused fouling.

It should also be noted that the product temperature in the cooker rose to unacceptable levels during both the initial temperature ramping phase and the feed rate disturbance phase. With temperatures exceeding 280°F the possibility existed that the process cooker tubes fouled, as well as the product cooked at these temperatures being unusable. In typical plant operations human intervention would have prevented the product from reaching these temperatures. However, for the purposes of these trials the PID controller was allowed to fully control the process, regardless of the product temperature.

Performance of Final MANTRA® Control

The MANTRA® control strategy exhibited marginally acceptable control performance in regards to the specified dynamic response parameters. The MANTRA® control strategy achieved an average process rise time of 14.5 minutes and recorded a maximum initial temperature overshoot (due to temperature ramping) of 0.4°F with an average temperature overshoot of 0.2°F. The system response from the final trial is shown in Figure 16.

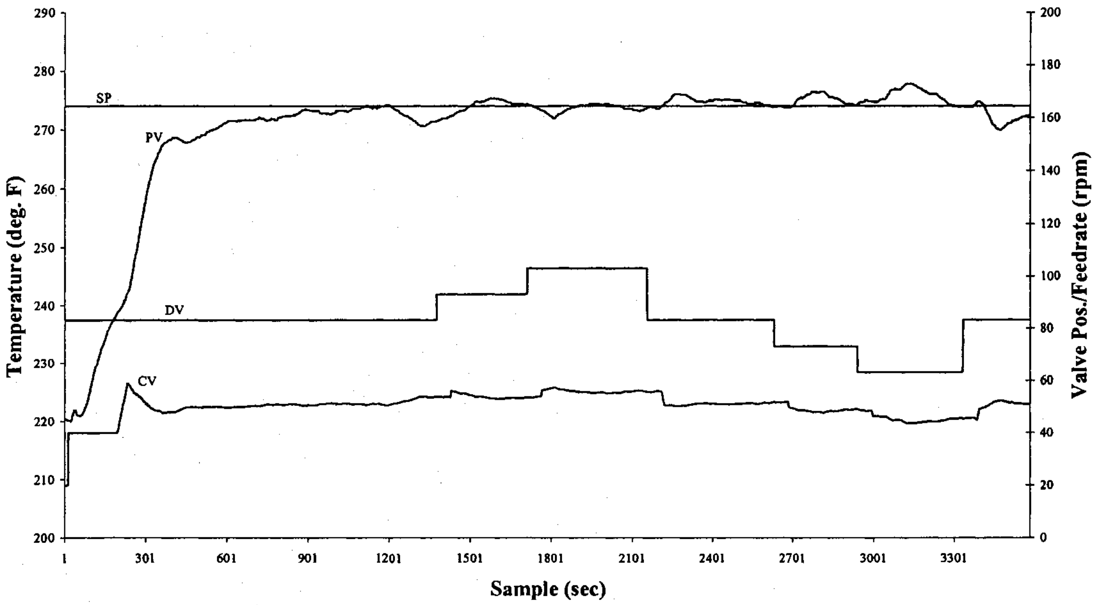


Figure 16. ControlSoft MANTRA® system response.

The MANTRA® control strategy met the control specification in regards to feed rate disturbances. The disturbance response in product temperature as a result of the product feed rate showed an average settling time of 4.3 minutes with an average deviation of 1.2°F and a maximum deviation of 4.1°F. The normalized integrated squared error was 2.2.

The MANTRA® controller did bring the product temperature up to set point with less than the specified maximum overshoot and within the specified rise time, however, the start up time only had a margin of approximately 0.5 minutes. The extended time for MANTRA® to achieve temperature set point can be attributed to how the control strategy was applied. A ramp function block with a pre-determined timing sequence was used to control the initial heating profile. This approach was used to simplify modeling the non-linear response of the steam valve position.

Additionally, MANTRA® exhibited less variation to an increase in product feed rate than in reacting to a decrease in product feed rate. A limitation with the MANTRA® controller is that only one set of process gains can be applied per feed forward model in the controller. The control strategy used with the Hohberger cooker tuned the model predictive parameters to reject an increase in feed rate with the assumption that a decrease in feed rate would have the same gain parameters. While this was an appropriate assumption for the MANTRA® control strategy, it resulted in less precise control during decreases in product feed rate.

Performance of Final BrainWave® Control

The BrainWave® control strategy achieved an average rise time of 5.5 minutes and recorded a maximum initial temperature overshoot of 3.0°F with an average overshoot of 1.7°F. The BrainWave® controller did cause some temperature overshoot for a time period of approximately 90 seconds, but the controller also took advantage of a model that ramped the process up in the fastest time. This was the fastest response time of all the APC strategies employed. Visual inspection showed that the product did not foul the cooker tubes as a result of the fast temperature rise. However, it is assumed that a slightly faster rise time would result in fouling. The system response from the final trial is shown in Figure 17.

The disturbance rejection response in product temperature, as a result of the product feed rate changes, showed an immediate response (settling time = 0) with an average deviation of 1.0°F and a maximum deviation from set point of 3.2°F. The normalized integrated squared error was 1.1.

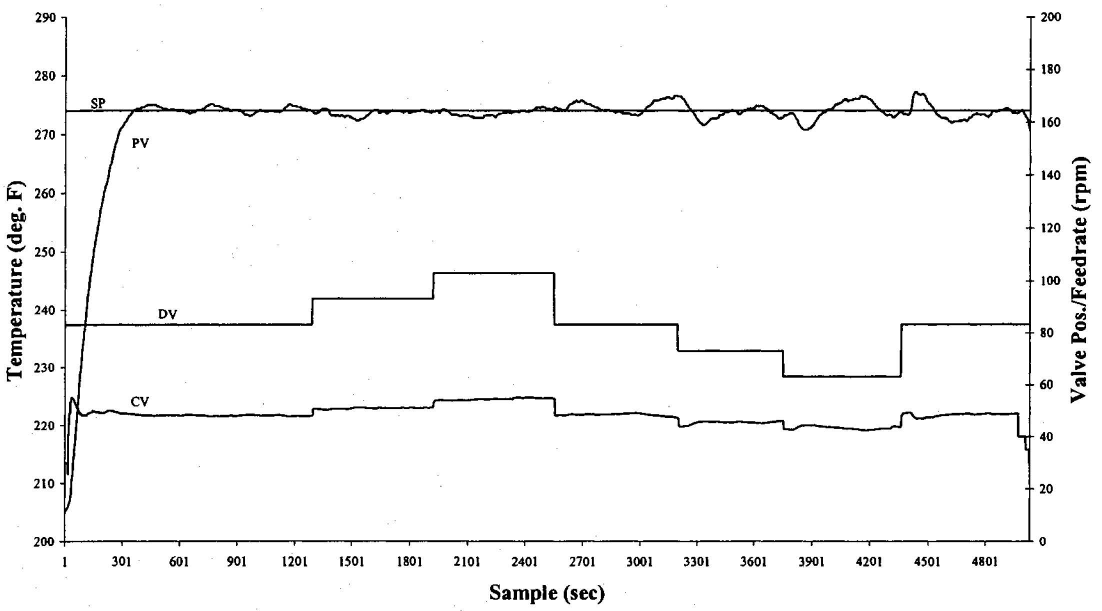


Figure 17. Universal Dynamics BrainWave® system response.

The BrainWave® model took into account the non-linear gain characteristic of the steam valve position (control output). This meant that the controller would be flexible enough to adjust to different temperature set points and still maintain target performance. For extremely non-linear processes several sub-models may be programmed to compensate for the changing gain requirements over preset regions of control. For example, the valve range could be divided into pseudo-linear segments and each segment could then be controlled via distinct model parameters. Switching logic would be used to operate through the various gain regions.

Performance of Final Process Perfecter® Control

The Process Perfecter® control strategy achieved an average rise time of 8.4 minutes and recorded a maximum initial overshoot (due to temperature ramping) of 1.7°F with an average temperature overshoot of 1.4°F. The system response from the final trial is shown in Figure 18.

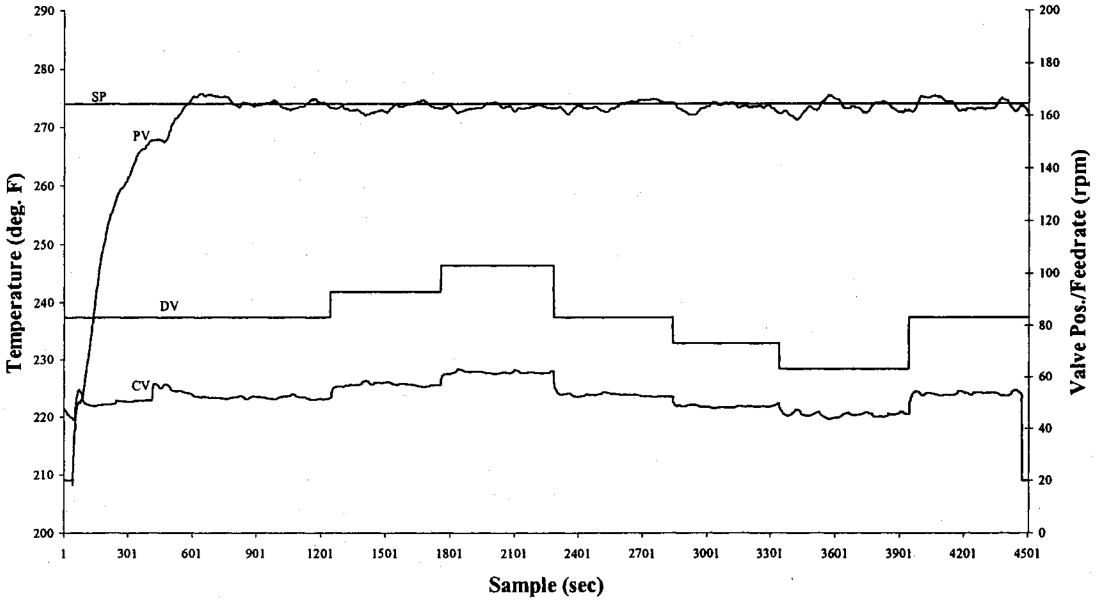


Figure 18. Pavilion Technologies Process Perfecter® system response.

The disturbance response in product temperature as a result of the product feed rate changes recorded an immediate response with an average deviation of 0.8°F and a maximum deviation of 2.8°F. Note that the Process Perfecter® neural network models rejected the feed rate decreases better than any of the other APC strategies. The normalized integrated squared error was 0.8.

An examination of the recorded results, particularly the integrated squared error, shows that the dynamic models used in Process Perfecter® maintained the tightest control during feed rate disturbances. Better control of the process during a decrease in feed rate could have been achieved by developing a neural model for both an increase and a decrease in product feed rate. For this application the decrease in feed rate was not specifically targeted, but was still learned to some extent by the neural network models. Additionally, the dynamic properties of the constructed steady-state models would allow it to maintain acceptable control for any set point change throughout the entire non-linear region of the steam valve range.

Performance of Final Model State Feedback Control

The Model State Feedback control strategy achieved an average rise time of 6.6 minutes and recorded a maximum initial overshoot (due to temperature ramping) of 2.9°F with an average overshoot of 2.3°F. The system response from the final trial is shown in Figure 19.

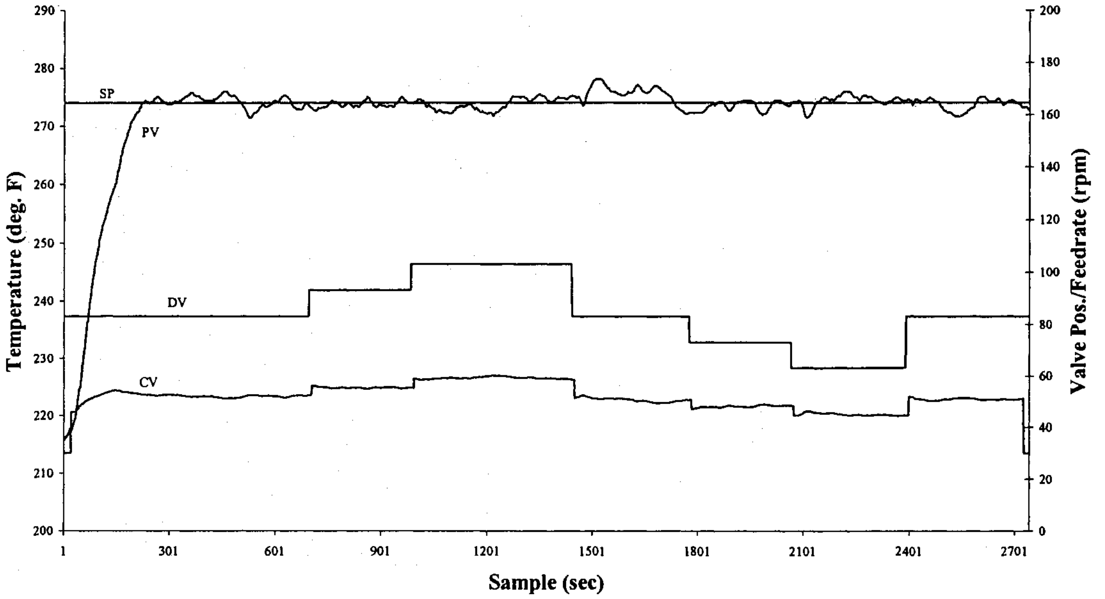


Figure 19. Model state feedback system response.

The disturbance response in product temperature as a result of the product feed rate changes recorded an immediate response with an average deviation of 0.9°F and a maximum deviation of 4.2°F from set point. The normalized integrated squared error was 1.4.

The Model State Feedback (MSF) control strategy did not have the fastest, most accurate dynamic response, or the best disturbance rejection during steady-state operation. Visual inspection showed that it did meet the control specifications and was comparable to the other APC control strategies. As with the other regulatory-based APC methods, the MSF control strategy had more difficulty rejecting a decrease in product feed rate than in rejecting an increase in product feed rate, due to the process non-linearity.

Overall Performance Analysis

Figure 20 is a compilation of the individual performance graphs for each of the control strategies using a typical production run for each. This graphical comparison shows visually that the four MPC strategies significantly outperformed the standard PID control with gain scheduling. It additionally shows that the four MPC strategies performed similarly, and that each of them would be an acceptable strategy for the sugar cooker.

Figure 21 compares the normal distribution of error for each of the control strategies evaluated. When comparing recorded data, especially integrated squared error (see Figure 22), it is seen that Process Perfecter® performed with the most precise control. However, Process Perfecter® was not as accurate as the other methods; its control tended towards a lower operating temperature than specified by 0.4°F. The most accurate control was achieved by the MANTRA® controller, though it did not have as tight or responsive control as the other methods. It is apparent from these graphs that PID control was much less precise than the MPC strategies. It is also seen that the MPC strategies exhibited much tighter control than PID, resulting in less overall system error.

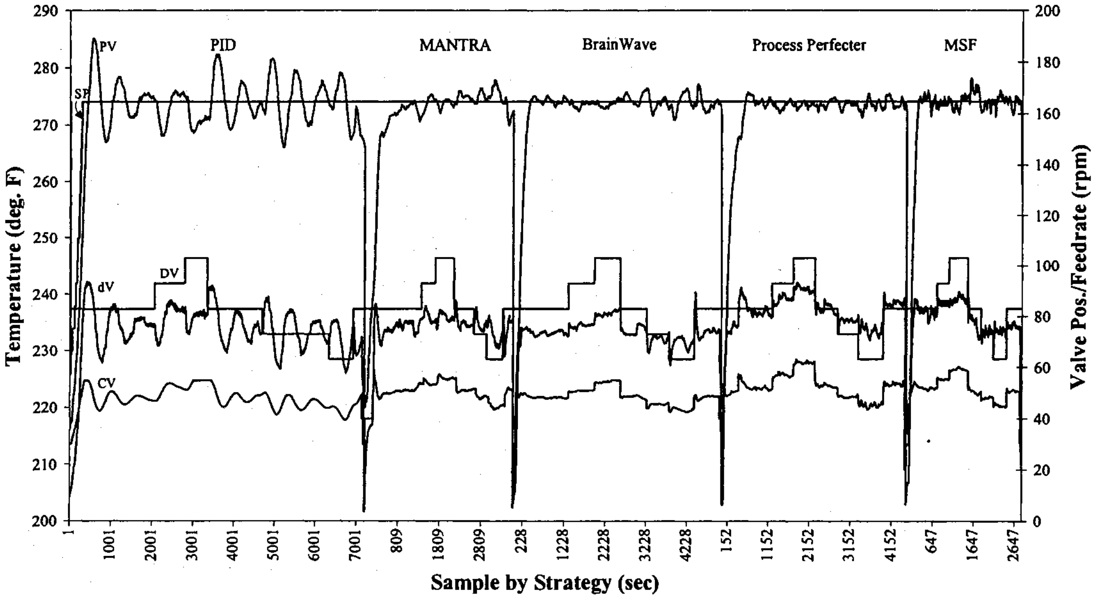


Figure 20. Overall system response comparison.

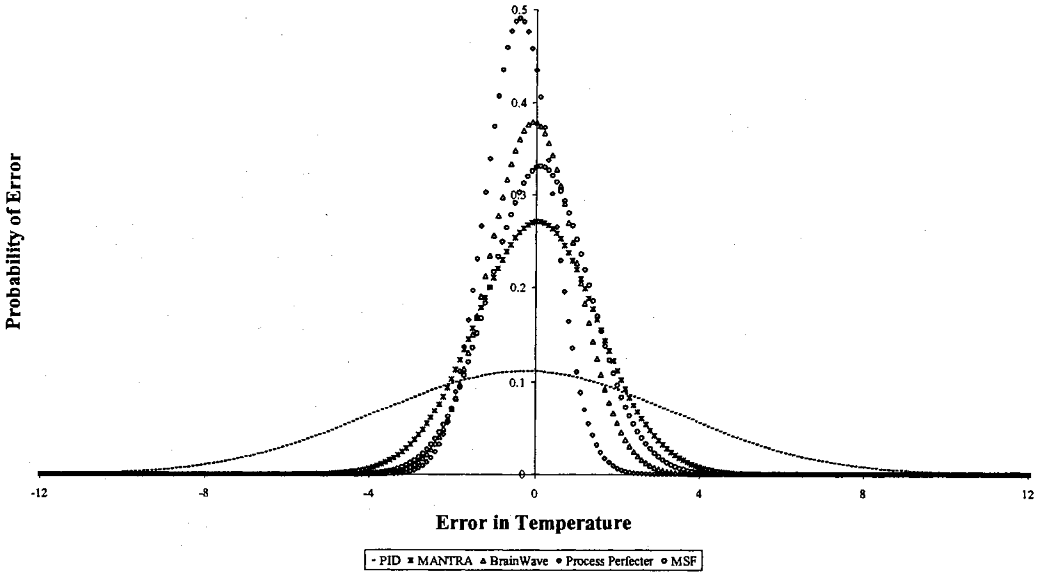


Figure 21. Error distribution comparison.

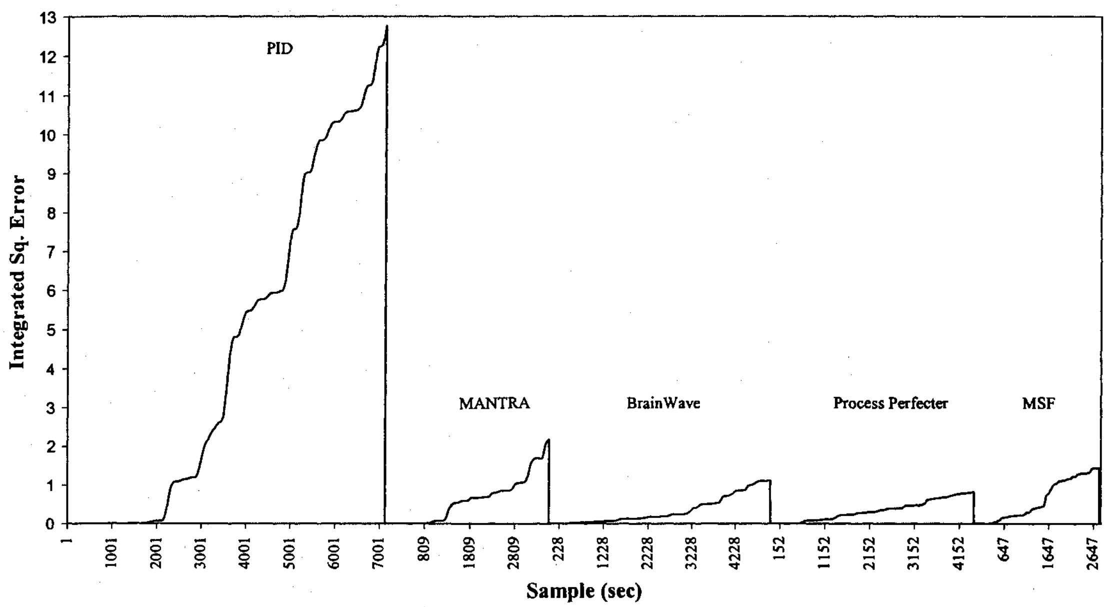


Figure 22. Integrated squared error comparison.

Analysis of the Hypotheses Tests

Three analyses were performed to support the research hypotheses. The collected data was analyzed using the S-PLUS® Student Edition version 4.5 release 2 statistical software, published by International Thompson Publishing Company. The following test statistics were used:

1. A one-way fixed ANOVA was used to test for a difference in the mean rise times for the different control strategies at the $\alpha = .05$ significance level. Specific differences among treatments were examined using Tukey's post hoc test for honestly significant difference comparisons. Summary statistics were then used to determine the direction of the observed differences.

2. A one-way fixed ANOVA ($\alpha = .05$) was used to test for a difference in temperature overshoot as a result of the initial product temperature rise for the different control solutions. Again, the specific differences among treatments were examined using a Tukey's post hoc test for honestly significant difference comparisons and summary statistics were used to determine the direction of the observed differences.

3. To test for a difference in deviation in temperature around the set point in the presence of system disturbances during steady-state operation for the different control solutions, two independent variables were examined. These were the control strategy and the product feed rate. The product feed rate was defined as the system disturbance, and was varied in a prescribed, step-wise fashion. A two-way ANOVA ($\alpha = .05$) was used. Both test method and feed rate were treated as fixed factors. The feed rate was treated as a fixed factor due to the fact that speed adjustments were only made in fixed increments based on the baseline speed. Only main effects were examined. The interactions were not examined because the independent variables were not manipulated simultaneously. A Tukey's post hoc test for honestly significant difference comparisons and summary statistics were used to determine the direction of the observed differences.

Table 4 lists the dynamic control summary statistics calculated based on the production trials for each control strategy. The steady-state control summary statistics are shown in Table 5. The results of each hypothesis test are detailed in the following sections.

Table 4

Application Strategy Dynamic Control Results and Comparison

Method	$M_{Rise\ Time}$	$SD_{Rise\ Time}$	Max. Initial Overshoot	$M_{Overshoot}$	$SD_{Overshoot}$
PID	28.8 min.	2.5 min.	11.0°F	8.6°F	2.7°F
MANTRA®	14.5 min.	1.0 min.	0.4°F	0.2°F	0.12°F
BrainWave®	5.5 min.	0.29 min.	3.0°F	1.7°F	0.82°F
Perfecter®	8.4 min.	0.29 min.	1.7°F	1.4°F	0.45°F
MSF	6.6 min.	0.54 min.	2.9°F	2.3°F	0.39°F

Table 5

Application Strategy Steady-State Results and Comparison

Method	$M_{Settling\ Time}$	M_{Temp}	SD_{Temp}	$M_{Abs\ Err}$	$SD_{Abs\ Err}$	Max. Deviation	ISE
PID	5.8 min.	273.8°F	3.6°F	2.1°F	1.1°F	8.3°F	12.8
MANTRA®	4.3 min.	274.0°F	1.5°F	1.2°F	0.52°F	4.1°F	2.2
BrainWave®	N/A[a]	273.9°F	1.0°F	0.96°F	0.39°F	3.2°F	1.1
Perfecter®	N/A[a]	273.6°F	0.8°F	0.78°F	0.24°F	2.8°F	0.8
MSF	N/A[a]	274.1°F	1.2°F	0.86°F	0.36°F	4.2°F	1.4

[a] Process responded to disturbances too quickly to see a process lag.

Temperature Rise Time Hypothesis Test

Research Hypotheses 1 and 4 (Chapter 1, p. 5) were tested by running an ANOVA on the difference in the mean rise times for the five control strategies. The ANOVA showed that there was a highly significant difference between the different control strategies ($F_{4,20} = 301.8, p << .05$, Table 6).

Table 6

Fixed Effects ANOVA Results for Temperature Rise Time

Source	*df*	*SS*	*MS*	*F*	*p*
Control Strategy	4	6640799	1660200	301.8238	$p << .05$
Residuals	20	110011	5501		

The Tukey honestly significant difference comparison was used to generate 95% simultaneous confidence intervals for linear combinations of the control methods. Tukey's method found no significant difference between model state feedback control and BrainWave® control, or between model state feedback control and Process Perfecter® control. All other control methods were determined to be different from one another at the .05 level (Tukey's, critical point = 2.99, Table 7).

Table 7

Tukey's Post Hoc Test for Temperature Rise Time

Comparison	Estimate	Std. Error	Lower Bound	Upper Bound
PID - MANTRA®	853	46.9	713.0	993.0[a]
PID - BrainWave®	1400	46.9	1260.0	1540.0[a]
PID - Perfecter®	1220	46.9	1080.0	1360.0[a]
PID - MSF	1330	46.9	1190.0	1470.0[a]
MANTRA® - BrainWave®	543	46.9	403.0	683.0[a]
MANTRA® - Perfecter®	369	46.9	229.0	510.0[a]
MANTRA® - MSF	476	46.9	336.0	616.0[a]
BrainWave® - Perfecter®	-174	46.9	-314.0	-33.2[a]
BrainWave® - MSF	-67	46.9	-207.0	73.4
Perfecter® - MSF	107	46.9	-33.8	247.0

[a] Comparisons differ at p < .05 in the Tukey honestly significant difference comparison.

Based on the observed differences in Table 7, an examination of the mean rise time (Table 4) for each control strategy showed that the MSF, BrainWave®, and Process Perfecter® control methods reached the set point temperature faster than either standard PID control or MANTRA® control. Each of these methods, as well as the MANTRA® strategy, were within the required rise time criterion of 15 minutes. The PID control strategy was unable to achieve the required criterion.

Temperature Overshoot Hypothesis Test

A second ANOVA was run on the difference in the peak temperature overshoot for the five control strategies in order to address Research Hypotheses 2 and 5. The ANOVA showed a highly significant difference between the control strategies ($F_{4,20} = 32.53, p <<$ .05, Table 8).

Table 8

Fixed Effects ANOVA Results for Temperature Overshoot

Source	*df*	*SS*	*MS*	*F*	*p*
Control Strategy	4	219.4536	54.86340	32.52822	$p <<$ .05
Residuals	20	33.7328	1.68664		

The Tukey honestly significant difference comparison was used to generate 95% simultaneous confidence intervals for linear combinations of the control methods. Tukey's method found significant differences only between PID control and each of the MPC control strategies (Tukey's, critical point = 2.99, Table 9). Significant differences were not observed between the different MPC strategies.

Table 9

Tukey's Post Hoc Test for Temperature Overshoot

Comparison	Estimate	Std. Error	Lower Bound	Upper Bound
PID - MANTRA®	8.38	0.821	5.92	10.800[a]
PID - BrainWave®	6.90	0.821	4.44	9.360[a]
PID - Perfecter®	7.24	0.821	4.78	9.700[a]
PID - MSF	6.36	0.821	3.90	8.820[a]
MANTRA® - BrainWave®	-1.48	0.821	-3.94	0.978
MANTRA® - Perfecter®	-1.14	0.821	-3.60	1.320
MANTRA® - MSF	-2.02	0.821	-4.48	0.438
BrainWave® - Perfecter®	0.34	0.821	-2.12	2.800
BrainWave® - MSF	-0.54	0.821	-3.00	1.920
Perfecter® - MSF	-0.88	0.821	-3.34	1.580

[a] Comparisons differ at $p < .05$ in the Tukey honestly significant difference comparison.

An examination of the peak temperature overshoot for each control strategy showed that the PID controller experienced significantly more temperature overshoot than any of the MPC strategies as shown in Table 4. The PID control method experienced a maximum temperature overshoot of 11°F, and thus was unable to achieve the required criterion of less than 5°F temperature overshoot. Each of the MPC strategies did meet this criterion.

Feed Rate Disturbance Hypothesis Test

A two-way ANOVA was run to examine the disturbance rejection capabilities for the five control strategies in order to address Research Hypotheses 3 and 6. The ANOVA showed a highly significant difference between the control strategies ($F_{4,131} = 22.40, p <<$ .05, Table 10). The ANOVA also showed a significant difference between the speed disturbances ($F_{4,131} = 3.26, p = .014$, Table 10).

Table 10

Fixed Effects, 2-Way ANOVA Results for Feed Rate Disturbance

Source	*df*	*SS*	*MS*	*F*	*p*
Control Strategy	4	31.10278	7.775696	22.40414	$p << .05$
Speed	4	4.52304	1.130761	3.25806	0.013889
Residuals	131	45.46554	0.347065		

The 95% simultaneous confidence intervals for linear combinations of the control methods were calculated using the Tukey honestly significant difference comparison. Significant differences between PID control and each of the MPC control strategies were observed (Tukey's, critical point = 2.77, Table 11). The MPC strategies did not show significant differences between each other in regards to rejecting feed rate disturbances.

Table 11

Tukey's Post Hoc Test for Feed Rate Disturbance Rejection

Comparison	Estimate	Std. Error	Lower Bound	Upper Bound
PID - MANTRA®	0.875	0.157	0.439	1.31[a]
PID - BrainWave®	1.11	0.157	0.675	1.55[a]
PID - Perfecter®	1.3	0.157	0.86	1.73[a]
PID - MSF	1.21	0.157	0.779	1.65[a]
MANTRA® - BrainWave®	0.236	0.157	-0.199	0.672
MANTRA® - Perfecter®	0.421	0.157	-0.0147	0.856
MANTRA® - MSF	0.34	0.157	-0.0959	0.775
BrainWave® - Perfecter®	0.185	0.157	-0.251	0.62
BrainWave® - MSF	0.104	0.157	-0.332	0.539
Perfecter® - MSF	-0.0812	0.157	-0.517	0.354

[a] Comparisons differ at p < .05 in the Tukey honestly significant difference comparison.

Confidence intervals for linear combinations of the feed rate changes were not examined. It was not important to evaluate one feed rate compared to another across all control strategies (for example 83 RPM for all methods vs. 103 RPM for all methods). In actual application a single control method would be implemented and would be required to perform to specification for any valid feed rate. It is not the case that multiple control methods would be applied to the process. Therefore, it was only important to observe differences in the control strategies across all feed rates.

Examination of mean absolute error, due to feed rate disturbances, (Table 5) for each control strategy showed that the PID controller experienced more temperature deviation than the MPC strategies. The PID control strategy was unable to achieve the required criterion of less than 1°F deviation on average. The MANTRA® controller was also unable to maintain and average deviation of less than 1°F, though it did experience significantly less average deviation than the PID controller. Each of the other MPC methods met this criterion.

It should be noted that for each ANOVA the variation in the PID controller was more than four times larger than the variation of the smallest MPC strategy variation for the variable under test. This violates the homogeneity of variance assumption. However, when groups are of equal sample size the ANOVA is typically robust enough to overcome this departure from homogeneity (Box, 1954).

Summary

Industrial sugar cooker production trials using five different control strategies were carried out. The control strategies included standard PLC-based PID control, three PC-based commercial MPC controllers, and PLC-based MPC control using the model state feedback algorithm. Statistical analyses were performed on the resulting data sets from each control strategy.

Analysis of the data showed that the PID control strategy was not able to meet the defined performance criteria. The PID controller did not successfully bring the product up to the set point temperature without significant overshoot within the required time. Additionally, the PID controller was not able to maintain the process set point within the required dead band during system disturbances.

All of the implemented MPC control strategies did meet the defined performance criteria. The data analysis also showed that there was not a significant difference in the operational results of the different MPC control strategies with regard to temperature overshoot and disturbance rejection during feed rate changes. A significant difference was demonstrated in rise time for the MANTRA® controller.

The hypotheses that the MSF control strategy would reach the final temperature faster, with less overshoot, and would reject system disturbances better than the PID controller were validated. However, the hypotheses that the MSF control strategy would outperform the three commercial MPC strategies for these same parameters were not proven. It was shown that the four MPC strategies operated similarly (i.e., no significant statistical differences) in regards to the performance criteria.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This study focused on the implementation of model predictive control techniques on an industrial sugar cooking process. The goal was to test an MPC solution directly on a PLC rather than on a PC. This study implemented and evaluated three PC-based, commercial MPC technologies for the sugar cooking process, and a new MPC implementation using a combination of ladder logic code and function blocks directly on Rockwell Automation's Allen-Bradley ControlLogix® PLC. The implementation results from these MPC solutions outperformed traditional PLC-based PID control. Additionally, the PLC-based MPC solution compared favorably to the PC-based commercial applications, though it did not outperform the commercial MPC strategies as hypothesized.

The results and data analysis reported in Chapter 4 provide several interesting insights into the use of model predictive control technologies, specifically as applied to the sugar cooking process. The following section reviews the experimental research carried out in this dissertation. Observations will be discussed, including both positive and negative outcomes resulting from the application of the MPC technologies used in this research study. System development issues for each MPC strategy will also be detailed. Conclusions will be drawn based on the specific research findings. Recommendations for further study and refinement will be made, along with a brief look at the industrial potential of the MSF algorithm implemented directly on the PLC as detailed in this research.

Study Overview

The sugar cooking process mixed a batch of product consisting of corn syrup, water and selected dry ingredients in a steam-jacketed mixing tank. The product inside the mixing tank was heated to 170°F. The syrup mixture was pumped from the tank through a preheater that raised the product temperature to approximately 220°F. A Waukesha positive displacement pump nominally running at 83 RPM was used to pump the syrup mixture to the Hohberger cooker heat exchanger section at a rate of 8.5 pounds per minute. The Hohberger cooker was a shell and tube style heat exchanger designed to take product at 220°F and produce a maximum of 20 pounds per minute of product at a final cook temperature of 274°F. A steam line served as the heating media for the cooker. Steam was metered to the heat exchanger through a Worcester steam-modulating flow valve. By controlling the position of the steam-modulating flow valve, the heat exchanger shell pressure was directly controlled. High-boil product temperature is a function of the heat exchanger shell pressure.

This study was done to test an alternate and superior control method to the PLC-based PID controller utilized on the sugar cooking process. The existing PID control solution was shown to be inadequate for controlling the dynamics of the sugar cooking process, or for rejecting process disturbances during steady-state operation. Model-based control strategies have been utilized throughout industry to overcome these control deficiencies seen in traditional PID controllers. The existing model-based controllers require a PC for operation rather than operating on the existing PLC. The additional hardware and software required for the MPC solution adds to the implementation cost, reducing the profitability, for the sugar cooker. Therefore, this study was additionally pursued to address shortfalls in the existing advanced process control solution possibilities. Although there are many commercially available MPC controllers for implementation on a stand-alone PC, to date there are no control packages for realizing model-based control techniques directly on the ubiquitous PLC.

The following model-based control strategies were evaluated in this research:

1. MANTRA® is an advanced control system from ControlSoft, Inc. It is a full-blown, configurable control system with more than 90 function blocks, configurable faceplates, and function block programming. The MPC functionality is based on the internal model control algorithm.

2. BrainWave®, from Universal Dynamics Inc., is an advanced regulatory controller designed specifically to deal with deadtime dominant systems and systems with a high degree of variable interactions. The model incorporates feed forward signals to compensate for the variable interactions. While it is model based, the mathematical technique used allows the model to be updated during on-line, closed loop operation. This means the models stay correct and change if the process does.

3. Pavilion Technologies' Process Perfecter® is an artificial neural network based solution that offers fully dynamic models, simultaneous solution of multivariable control problems, and fully integrated optimization techniques to drive operations to maximum profit. It supports nonlinear process representation on any model parameter.

4. The model state feedback algorithm provides model predictive control functionality using a combination of ladder logic code and function block programming

on the ControlLogix® PLC using the native PLC programming tools. The function blocks execute the model predictive functions. The ladder code provides for parameter assignments, auto-manual-track bumpless switching, controller initialization, communications, real-time parameter changes, fault handling, and redundancy-watchdog. Deterministic timing, as supported by periodic tasks in the ControlLogix® PLC, was required for accurate calculations.

Using these control strategies, the following research hypotheses were tested:

1. It takes significantly less time for the PLC-based model state feedback implementation of the MPC controller to reach the final product temperature set point than it does for standard PLC-based PID control applied to the same industrial sugar cooker.

2. The PLC-based model state feedback implementation of the MPC controller experiences less temperature overshoot due to the initial product temperature rise than the standard PID control solution.

3. There is a smaller deviation in temperature around the set point, in the presence of system disturbances, during steady-state operation for the PLC-based model state feedback implementation of the MPC controller than there is for the standard PID control solution.

4. The temperature rise time is shorter for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions applied to the industrial sugar cooker.

5. The PLC-based model state feedback implementation of the MPC controller exhibits less temperature overshoot as a result of the initial product temperature rise than the PC-based commercial MPC solutions.

6. The deviation in temperature around the set point, in the presence of system disturbances, is smaller for the PLC-based model state feedback implementation of the MPC controller than it is for the PC-based commercial MPC solutions.

To summarize the hypotheses, there were three main areas on which the overall comparative analysis focused. These comparison areas were the dynamic response of each strategy at startup, including both temperature rise time and overshoot, and the steady-state disturbance rejection capabilities of each strategy. The comparison results are presented here.

Tests of the sugar cooking control strategies were performed wherein the process temperature was ramped up to the set point as quickly as possible while still maintaining control of the system and resulting in minimal temperature overshoot. A disturbance was then introduced to the process by changing feed rate in a predefined manner, during which the process controller was evaluated on its ability to reject the disturbance and maintain the process temperature at the set point. Each of the five control strategies was subjected to the same test procedure.

The test results showed that the MPC strategies controlled the sugar cooking process better than the traditional PID control method in regards to temperature rise time, temperature overshoot, and disturbance rejection based on feed rate disturbances. It was seen that the differences between the various MPC strategies was not significant relative to temperature overshoot and disturbance rejection. However, it was shown that the MANTRA® controller did not perform as well as the other MPC strategies for temperature rise time. Based on these results, Research Hypotheses 1-3 above were supported by the data. However, Hypotheses 4-6 were not supported. The PLC-based MPC strategy was shown to be comparable, but not superior, to the PC-based commercial MPC applications.

Constructive Observations

There are several advantages to implementing model-based control techniques directly on the system PLC. This section discusses advantages discovered with this application of the model state feedback algorithm. Also noted are general advantages of model predictive control discovered as a result of this study.

One of the major advantages of the model state feedback control strategy employed directly on the PLC was the fact that no external communication protocols were required. Each of the other MPC control strategies required the communication protocols between the PLC and the PC to be properly configured prior to development and implementation. All three of the PC-based MPC strategies experienced difficulties during the communication configuration process that negatively affected development time. These difficulties were avoided with the PLC-based solution.

It should be noted that the internal steam pressure fluctuations of the shell and tube heat exchanger were not considered in any of the implemented control strategies. The pressure was assumed to be constant throughout the trials. However, the pressure did not remain constant during the trials, as shown by the disturbance variable (dV) in Figure 20

(Chapter 4, p. 98). This variable was used as an unmeasured disturbance to the controllers to verify robustness; the actual vessel pressure was recorded for analytical purposes. The MPC strategies were shown to be robust by accurately controlling the process temperature without accounting for all of the process system disturbances, such as internal vessel pressure.

During the test runs, it was observed that the actual process dynamics were somewhat different than the model developed for the MPC strategies. The actual gain of the steam valve was lower than the calculated model gain; the dead time for feed rate was also shorter than calculated from the step tests. These parameters were not modified for the operational model parameters with any of the MPC strategies. This was for the purpose of testing the robustness of the MPC solutions. A good, robust MPC strategy should be able to handle reasonable modeling errors. In contrast, using incorrect PID parameters will cause a PID-based control system to become unstable. Acceptable performance from a model-based controller should be obtained simply by changing the tuning time constant (Chia & Lefkowitz, 1997). This robustness feature is critical for real-time plant applications, as often plant engineers may not have the time and resources to modify the selected model. The test cases in this study indicate that the MPC solutions were robust.

It is possible to tune any of the MPC strategies to target specific control responses. For example, if it were determined that no overshoot was preferable to a minimum startup time, provided that the startup time met the 15-minute requirement, then the controller could be tuned to ramp up to temperature more slowly and use the set point as a constraint rather than a target. Conversely, the controller could be tuned to ramp up to temperature as quickly as possible and allow some degree of overshoot if the goal is a rapid startup. This tuning capability is not constrained to the dynamic aspects of the control system. Particular control responses can be tuned for steady-state operation as well. To illustrate, the system could be tuned such that the set point is treated as a hard constraint rather than an operating target. This would cause the process to trend below the set point, but not exceed the set point on a feed rate change. No such specific control responses were targeted for any of the trial runs in this study.

Implementation Issues

While the MPC strategies are much more accurate and capable of controlling systems that are deadtime dominant, have excessive noise profiles, contain interacting variables, and so forth, there are also a few disadvantages with this technology. One side effect of model-based control that is apparent throughout the various MPC trial runs is valve jitter. Jitter is when process actuator movements come in rapid succession. The valve (or other actuator) appears not to settle at a particular operating position. Jitter is a result of continuously predicting a new operating set point for the actuator, based on the process model, and subsequently commanding the actuator to move to this new set point. This causes increased wear on the mechanical components of the actuator, possibly resulting in increased maintenance requirements. It is possible to reduce jitter by lengthening the controller update interval. The slower the update interval, the less often the actuator will be commanded to move. However, slowing down the controller will also result in less accurate control. Considering the performance charts in Chapter 4 it is seen that the valve had considerably more jitter for the MPC solutions than it did for the PID solution. However, it is these predictive actuator moves that allow the MPC strategies to maintain tighter control than the PID controller.

Each control strategy exhibited poorer control when the feed rate was decreased, as opposed to increasing the feed rate. This was due to unexpected non-linear process dynamics. As stated in chapter one, the process was assumed to be linear. The controllers were designed based on this assumption. No specific efforts were made in this study to overcome this difficulty.

It was also noted that the control accuracy for all of the control strategies seemed to deteriorate throughout the day, likely due to process equipment heating. As the equipment was run continuously it tended to hold an increased amount of heat. This additional heat load changed the process characteristics, which caused increasing inefficiencies in each control strategy over the course of consecutive trials. It is believed that using the current ambient process equipment temperature as a feed forward variable in each of the control strategies would have solved this problem. Without this additional process information the controller had no way of compensating for the changes to the process model caused by heating.

System Development Characteristics

While not part of the statistical analysis, system development characteristics are of key importance to the long-term implementation potential of each strategy. Advanced process control techniques are often able to control difficult processing systems better than conventional control algorithms. However, MPC strategies are not all targeted at controlling the same types of processes or providing the same development tools. MANTRA® is a viable software solution for deadtime and coordinated control problems of minimal size and complexity. It is also designed to be a complete control strategy, not just a model-based controller. BrainWave® provides an elegant replacement to difficult PID problems, including deadtime and coupled-variable problems. However, it is only designed for regulatory control. Process Perfecter® is most appropriate for large systems with multiple coupled actions and poor coordination of set points. It is typically used for system optimization problems. The model state feedback algorithm is a viable solution for single-output only systems, implemented directly on a ControlLogix® PLC, where space or cost is a factor. However, this solution requires extensive process and electrical engineering expertise to understand and modify.

Development time required to build and implement each separate control strategy is another important development characteristic. MANTRA® and BrainWave® performed well as regulatory controllers and each required less than one week engineering time to commission a trial for this project. The pre-established model bases allows for quicker integration with these two control packages. A distinct disadvantage of the artificial neural network based approach used by Pavilion was that it took a considerable amount of engineering effort to develop and implement the control strategy. A typical application takes from weeks to months of engineering time. The initial MSF strategy took considerable time to design, and required an in-depth understanding of control theory. However, similar future implementations of the MSF strategy will require a similar amount of time as the MANTRA® and BrainWave® solutions.

Lastly, the cost of the developed system needs to be taken into account. The most cost effective control solution is one that can be applied directly on the PLC where no additional hardware or software is required, such as with PID control and the MSF algorithm in this research, and thus no additional system costs are incurred. The cost of other MPC solutions tends to vary in direct proportion to the strategy's scope of control and capabilities. In this research the MANTRA® product was the least expensive PC-based solution at approximately $5,000 per control loop. The BrainWave® solution was list priced at approximately $20,000 per system, although this includes control capabilities for replacing up to four PID control loops. The most expensive, and most capable system, was Process Perfecter® at approximately $75,000 per installation for an unlimited system application. It should be noted that these prices are list prices for the software. Additional costs for hardware and engineering also need to be considered.

Conclusions

Based on the results of the study, it is concluded that the model state feedback algorithm may be successfully implemented on a ControlLogix® PLC to control an industrial shell and tube style heat exchanger applied to cooking sugar syrup used in confectionary products. The MSF controller exhibited superior operational results compared to the standard PID-based controller in regards to reduced temperature rise time, overshoot minimization, and feed rate disturbance rejection. The MSF model predictive controller compared favorably to the commercially available MPC strategies studied for these same parameters.

Using an advanced control solution for the sugar cooking process resulted in several benefits, both financial and product related. These benefits included:

1. The tighter control limits reduced the amount of product that was wasted due to improper processing.

2. Energy usage was reduced due to less large temperature swings in the process.

3. More consistent product was produced that was closer to the target recipe.

4. The advanced automation reduced the amount of manual labor required to run the process.

5. More stable process control was established.

6. Process efficiency was increased.

Although the MSF solution was not found to perform superiorly compared to the PC-based MPC solutions, it was found to perform equally as well, thus making it a viable control solution for the industrial sugar cooker. Additional benefits were realized by implementing the advanced control solution directly on the ControlLogix® PLC. These included:

1. No external hardware (such as a PC) was required.

2. No additional programming software was required.

3. No special communications protocols (such as OPC) were required.

4. It was easier to maintain at the plant level because the solution was developed on a well-known platform using standard tools.

5. Implementation costs were significantly reduced due to the elimination of the external hardware and software.

Recommendations for Further Study

The results of this study warrant the following recommendations for future research. First, it should be noted that the shell and tube type heat exchanger used in this research is only one type of industrial cooker, and that sugar cooking is only one cooking process. Another common cooker is a scrape surface heat exchanger. This type of cooker is commonly used for cooking dairy-based products. The operation of the scrape surface type cooker is very similar to the shell and tube style cooker. However, there are differences in the process dynamics as a result of the different mechanics in the system. The application of the PLC-based MSF algorithm to other types of industrial cookers and products is recommended.

As previously noted, the internal vessel pressure in the sugar cooker was considered constant and was not compensated in any of the control strategies. However, the recorded steam pressure was not constant. It is believed that the MPC strategies would have controlled the process temperature with even greater accuracy if the internal vessel pressure were used as a feed forward variable in the control scheme. It is recommended that the internal vessel pressure fluctuations be integrated into the MPC controllers, and system accuracy be reevaluated.

The temperature control for the sugar cooking process was unexpectedly found to be non-linear. However, the controllers were all designed based on a linear system. For each controller this resulted in process control degradation for feed rate reductions. It is believed that this is a solvable problem. It is recommended that for the Process Perfecter® system the artificial neural network models be trained to recognize and compensate for the non-linearity. It is recommended for the other MPC strategies that the control strategy be divided into two feed rate disturbance models, one for increasing feed rate and one for decreasing feed rate. The appropriate model algorithm would be applied depending on process conditions.

In order for the PLC-based MSF algorithm to become an efficient solution for widespread utilization, there needs to be some additional human interface tools developed. It is recommended that further study be commenced to develop an engineering interface that includes on-line model development, configuration tools for response targeting, and tuning tools for optimizing the installed controller. An operator interface should also be investigated. This interface should include typical operational interface tools to control the process, as well as financial tools to quantify the savings realized by utilizing the MPC strategy. The implementation of these interface tools may create a control package that can compete with the commercial PC-based MPC strategies.

Summary

It has been shown that model-based controllers can be used in specific processing applications to improve the existing control quality achieved using standard PID loop controllers. Model-based controllers can successfully control processes with relatively long response delays (deadtime) or processes that have randomly occurring disturbances such as changes in production rate. Other processes have controls that may frequently drift from a desired set point forcing system operators to use manual control to bring these processes back into correct specifications. Model predictive control can be used as a viable solution for these difficult control requirements.

The model state feedback algorithm applied directly on an industrial PLC has successfully controlled the sugar cooking process with proficiency equal to the commercial PC-based strategies. This strategy has several benefits, such as requiring no external hardware and software, which results in less expensive implementation than the currently available commercial MPC strategies. It is recommended that this strategy be further developed and refined for general application throughout the food and beverage industry.

REFERENCES

Anderson, K. L., Blankenship, G. L., & Lebow, L. G. (1988). A rule-based adaptive PID controller. *Proceedings of the 27th IEEE Conference on Decision and Control, 1,* 564-569.

Barnes, R. (1996, June). Hybrid-based PLC control predicts, optimizes product quality. *Control Engineering, 43,* 75-78.

Bequette, W. B. (1998). *Process dynamics modeling, analysis, and simulation.* Upper Saddle River, NJ: Prentice Hall PTR.

Blevins, T. (2001, June). Applying embedded model predictive control successfully. *Control Solutions,* (Suppl. 1), 2-8.

Blevins, T. L., McMillan, G. K., Wojsznis, W. K., & Brown, M. W. (2003). *Advanced control unleashed: Plant performance management for optimum benefit.* Research Triangle Park, NC: ISA.

Box, G. E. P. (1954). Some theorems on quadratic forms applied in the study of analysis of variance problems. *Annals of Statistics, 25,* 290-302.

Boyce, J., & Brosilow, C. (1996). Multivariable cascade control for processes with output constraints. *Proceedings of European Symposium on Computer Aided Process Engineering, 6,* 1-10.

Brosilow, C., & Joseph, B. (2002). *Techniques of model based control.* Upper Saddle River, NJ: Prentice Hall PTR.

Bryan, L. A., & Bryan, E. A. (1997). *Programmable controllers theory and implementation.* Atlanta, GA: Industrial Text Company.

Buckbee, G. (2002, March). Poor controller tuning drives up valve costs. *Control Magazine, XV,* 47-51.

Chia, T. L., & Brosilow, C. B. (1991, June). Modular multivariable control of a fractionator. *Hydrocarbon Processing,* 61-66.

Chia, T. L., & Lefkowitz, I. (1997, September). Add efficiency with multivariable control. *Intech, 44,* (9), 85-87.

ControlSoft, Inc. (2000). *Mantra NT user manual* (Publication No. MN301-02/00 NT). Cleveland, OH: ControlSoft, Inc.

Coulibaly, E., Maiti, S., & Brosilow, C. (1992). *Internal model predictive control (IMPC).* Paper presented at the Annual Meeting of the AIChE, Miami, FL.

Demoro, E., Axelrud, C., Johnson, D., & Martin, G. (1997). *Neural network modeling and control of polypropylene process.* Paper presented at the Society of Plastic Engineers Polyolefins X International Conference, Houston, TX.

Dormer, A. (1998, April). Advanced control and optimization of process units. *Hydrocarbon Engineering,* 1-4.

Eder, H. (2003a, April). Advanced process control: What is it really? *Control Solutions International,* 34, 37-38.

Eder, H. (2003b, June). Where to find payback for advanced control. *Control for the Process Industries, XVI.* Retrieved June 12, 2003, from http://www.controlmagazine.com/Web_First/CT.nsf

EnTech Control Engineering Inc. (1993, November). *Automatic controller dynamic specification.* Retrieved April 24, 2003, from http://www.emersonprocess.com/ entechcontrol/ download/publications/control.pdf

Gerry, J. (1998, March). How to control processes with large deadtimes. *Control Engineering Online, 45.* Retrieved February 27, 2002, from http://www.controleng.com/archives/ 1998/ctl0301.98/03a306.htm

Gerry, J., & Shinskey, F. G. (2000). *PID controller specification.* Retrieved March 25, 2002, from http://www.expertune.com/PIDspec.htm

Godena, G., & Colnaric, M. (2000, August). Exception handling for PLC-based process control software. *Microprocessors and Microsystems, 24,* 407-414.

Gough, W. A. (2003). *Predictive adaptive regulatory control with Brainwave.* Manuscript submitted for publication.

Griffen, B. T. (1995). *Artificial neural network prediction of injection molding parameters.* Unpublished master's thesis, Brigham Young University, Provo, UT.

Group Engineering. (2001). *Factory control specification* (Electrical Specification No. 16979). Glendale, CA: Author.

Guinan, T., Kelly, G., & Semrad, D. (1998). *Neural network NOx optimization project.* Paper presented at Pavilion Technologies User Group Meeting, Austin, TX.

Harrold, D. (1999, August). Process controller tuning guidelines. *Control Engineering Online, 46.* Retrieved April 25, 2003, from http://www.controleng.com/archives/ 1999/ ct10801.99/9908we1.htm

Hartman, E. (2002). *A (very) brief history of artificial neural networks.* Austin, TX: Pavilion Technologies, Inc.

Havener, J., & Unrau, E. (2001, November). Reducing energy costs in a central utility system through optimization – impact for a pharmaceutical company. *Pharmaceutical Engineering, 21,* (6), 1-6.

Helps, C. R. G., & Griffen, B. T. (1994). Predicting mold-cavity temperature with an artificial neural network. *Plastics Engineering, L* (10), 25-27. Brookfield, CT: Society of Plastics Engineers.

Huzmezan, M., Dumont, G. A., Gough, W. A., & Kovac, S. (2001). *Adaptive control of integrating time delay systems: A PVC batch reactor.* Manuscript submitted for publication.

Huzmezan, M., Gough, B., Kovac, S., Le, L., & Roberts, G. (1999). *A new generation of adaptive model based predictive controllers applied in batch reactor temperature control.* Paper presented at the IEEE APC '99 Workshop, Vancouver, Canada.

Isaksson, A. (1999, September). *Model based control design.* Retrieved March 5, 2003, from http://www.s3.kth.se/control/kurser/2E1200/downloads/IMCkomp.pdf

Johnson, C. D., & Malki, H. A. (2002). *Control systems technology.* Upper Saddle River, NJ: Pearson Education, Inc.

Kay, J., & Thompson, A. (1998, December). Dead time is death to PID control. *Intech, 45,* (12), 60-62.

Keeler, J. D., Hartman, E., & Piche, S. (1996). *Process modeling and optimization using focused attention neural networks* (Technical Report Turning Your Data Into Gold). Austin, TX: Pavilion Technologies, Inc.

Lefkowitz, I., & Beiler, E. (1996). *Intune.* Cleveland, OH: ControlSoft, Inc.

Manesis, S. A., Sapidis, D. J., & King, R. E. (1998). Intelligent control of wastewater treatment plants. *Artificial Intelligence in Engineering, 12,* 275-281.

Martin, P. G. (2002). *Bottom-line automation.* Research Triangle Park, NC: ISA.

McMillan, G. (2000). *Good tuning: A pocket guide.* Research Triangle Park, NC: ISA.

Mhatre, S., & Brosilow, C. (n.d.). *Multivariable model state feedback: A computationally simple, easy to tune alternative to MPC.* Retrieved March 5, 2003, from http://www.cwru.edu/cse/eche/people/faculty/brosilow/papers/mvmsf.pdf

Parker, P. (2002, May). High-turndown process control. *Control Magazine, XV,* 49-52.

Pavilion Technologies, Inc. (2002). *Process Perfecter 6.5 off-line user's guide.* Austin, TX: Pavilion Technologies, Inc.

Piche, S., Sayyar-Rodsari, B., Johnson, D., & Gerules, M. (2000, June). Nonlinear model predictive control using neural networks. *IEEE Control Systems Magazine,* 53-62.

Rivera, D. E. (1999a). *ChE 461/598: Process dynamics and control; Introduction to internal model control with application to PID controller tuning.* Unpublished manuscript, Arizona State University.

Rivera, D. E. (1999b). *Internal model control: A comprehensive view.* Tempe, AZ: Arizona State University, Department of Chemical, Bio and Materials Engineering.

Rockwell International Corporation. (2002). *Allen-Bradley ControlLogix system* (Publication 1756-UM001E-EN-P). Milwaukee, WI: Rockwell Automation.

Shridhar, R., & Cooper, D. J. (1998). A novel tuning strategy for multivariable model predictive control. *ISA Transactions, 36,* 273-280.

Stryczek, K. (1996). *Mp tuning of internal model control systems under parametric uncertainty.* Unpublished doctoral dissertation, Case Western Reserve University, Cleveland, OH.

Studebaker, P. (1995, September). Advanced control in the real world. *Control Magazine, VIII,* 75-80.

Tay, M. E. (1996, September). Validating plant data. *Chemical Engineering,* 82-88.

Tzovla, V., & Mehta, A. (2000). *A simplified and integrated approach to model predictive control implementation.* Research Triangle Park, NC: ISA.

Universal Dynamics Technologies Inc. (1998a). *Getting control of deadtime intensive loops.* Vancouver, BC: Universal Dynamics Technologies Inc.

Universal Dynamics Technologies Inc. (1998b, April). *Methods of defining the success of a control strategy* (Technical Note 004). Vancouver, BC: Universal Dynamics Technologies Inc.

Universal Dynamics Technologies Inc. (2000). *Brainwave 2000 user's guide.* Vancouver, BC: Universal Dynamics Technologies Inc.

VanDoren, V. J. (1998, August). Ziegler-Nichols methods facilitate loop tuning. *Control Engineering, 45.* Retrieved February 27, 2002, from http://www.controleng.com/archives/ 1998/ct10801.98/08abas.htm

VanDoren, V. J. (2001, September). Real-world applications of advanced process control. *Control Engineering, 48,* 51-56.

VanDoren, V. J. (2002, October). Adaptive controllers. *Control Engineering, 49,* 20-30.

VanDoren, V. J. (Ed.). (2003). *Techniques for adaptive control.* New York: Butterworth-Heinemann.

Willis, M. J., & Tham, M. T. (1994, April). *Advanced process control.* Unpublished manuscript, University of Newcastle upon Tyne, United Kingdom. Retrieved January 3, 2002, from http://lorien.ncl.ac.uk/ming/advcontrl/apc.htm

Wojsznis, W., Blevins, T., Gudaz, J., & Mehta, A. (2001). *Practical approach to tuning MPC.* Paper presented at the ISA 2001 Conference, Houston, TX.

APPENDICES

APPENDIX A

MODEL STATE FEEDBACK ALGORITHM DERIVATION

In developing the model state feedback algorithm it is convenient to begin with the concept of a perfect controller. To do this it is assumed that there are no process disturbances or influences other than the external set point. A perfect controller is the exact inverse of the process to be controlled. Under these conditions the control system operates in open loop control. Equation A9 shows that the open loop transfer function equals one. Therefore the process variable follows the set point perfectly.

$$\mathrm{MPC} = \mathrm{P}^{-1} \Rightarrow \mathrm{PV} = (\mathrm{MPC} \cdot \mathrm{P}) \cdot \mathrm{SP} \tag{A9}$$

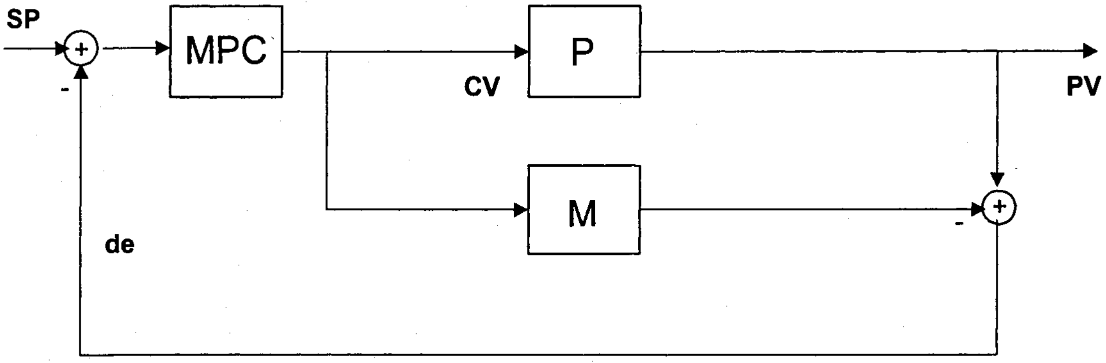
For model-based control the ideal model is a perfect replica of the process under control. For a first order lag plus deadtime process, such as the one found in this research, the model is determined by the process gain (K), time constant (τ), and deadtime (θ) as described in Chapter 2 and shown in Equation A10.

$$\mathrm{P} = \mathrm{M} = \frac{\mathrm{K}e^{-\theta s}}{\tau s + 1} \tag{A10}$$

Again assuming a perfect controller, the model-based controller would be the inverse of the FOPDT model. However, inverting this model would result in two problems: (a) the inverse of the deadtime component generates an unrealizable prediction of future process disturbances or changes, and (b) the inverse of the (τs + 1) term in the numerator would require a pure differentiation of the process output that would result in excessive noise amplification and could potentially cause a numerical overflow or divide by zero error in the PLC. To generate a realizable controller the unrealizable demand that the controller would have to predict (anticipate) the operator's intentions to change set point in the future needs to be removed (Brosilow & Joseph, 2002). Equation A11 shows the closest realizable function to the FOPDT model inverse. In this controller, ϵ is a tuning parameter chosen to avoid excessive noise amplification, compensate for modeling errors, and select the system response speed. Increasing ϵ slows the response speed, while decreasing ϵ increases the response speed. System robustness is increased with a higher the value of ϵ (Rivera, 1999b).

$$\mathrm{MPC} = \frac{\tau s + 1}{K(\varepsilon s + 1)} \tag{A11}$$

In real applications there are always process disturbances and the controller is never a perfect inverse of the process. Therefore, it is necessary to use feedback to compensate for these discrepancies. Figure A23 depicts the closed loop control system. The system follows the standard closed loop transfer function, with the additional compensation for the difference between the process model and the actual process (see Equation A12).



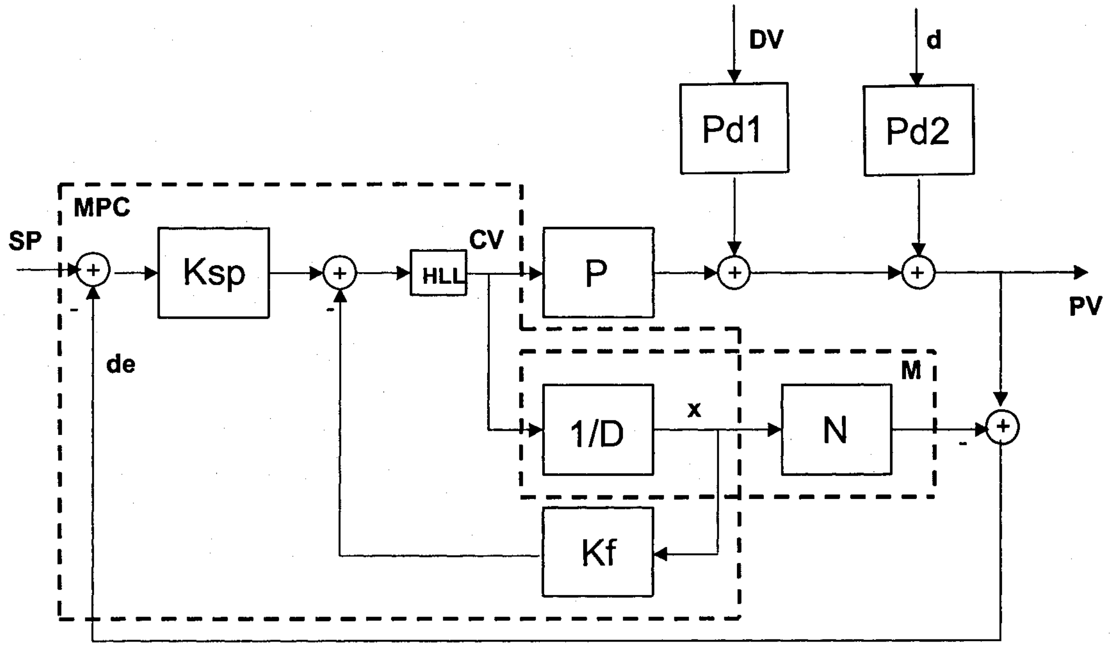
where,
MPC: model predictive process controller
P: physical plant process to be controlled
M: mathematical process model

Figure A23. Model predictive controller with feedback.

$$\mathrm{PV} = \frac{\mathrm{P} \cdot \mathrm{MPC}}{1 + (\mathrm{P} - \mathrm{M}) \cdot \mathrm{MPC}} \cdot \mathrm{SP} \qquad \text{(A12)}$$

The controller shown in Figure A23 assumes that the calculated control effort has been applied and that any future calculated control efforts could be applied. However, the reality of the situation is that the controller output is limited and can become saturated if unrealized control efforts remain unaccounted. This leads to process overshoot and sluggish responses. To compensate, a high-low limiter (HLL) is applied to provide anti-reset windup. The HLL function constrains the output effort to within the limits of the physical process. When the calculated control effort is outside these limits the controller updates the internal model states and predictions according to the actual output sent to the process and recalculates the control effort to comply with the imposed system output constraints (Coulibaly, Maiti, & Brosilow, 1992). This allows the controller to accurately control the process based on the applied control efforts.

Brosilow and Joseph (2002) explain how the MSF implementation is developed by splitting the model into its numerator and denominator, splitting the controller gain, and reworking the controller block diagram as shown in Figure A24. Equations A13 through A16 are the individual block components that make up the MSF controller. K_{SP} and K_f are the MSF gains and K is the process model gain determined from the step tests. Based on these calculations, the controller output (CV) is determined by Equation A17, where x is an internal model state (see Figure A24).



where,

K_{SP}:	set point gain
K_f:	state multiplier
P:	physical plant process to be controlled
D, N:	mathematical process model components
HLL:	high-low limiter
Pd1:	measured process disturbance lag
Pd2:	unmeasured process disturbance lag

Figure A24. Feedback portion of the model state feedback controller.

$$N = Ke^{-\theta s} \tag{A13}$$

$$D = \tau s + 1 \tag{A14}$$

$$K_{SP} = \frac{\tau}{K \cdot \varepsilon} \tag{A15}$$

$$K_f = \frac{\tau}{\varepsilon} - 1 \tag{A16}$$

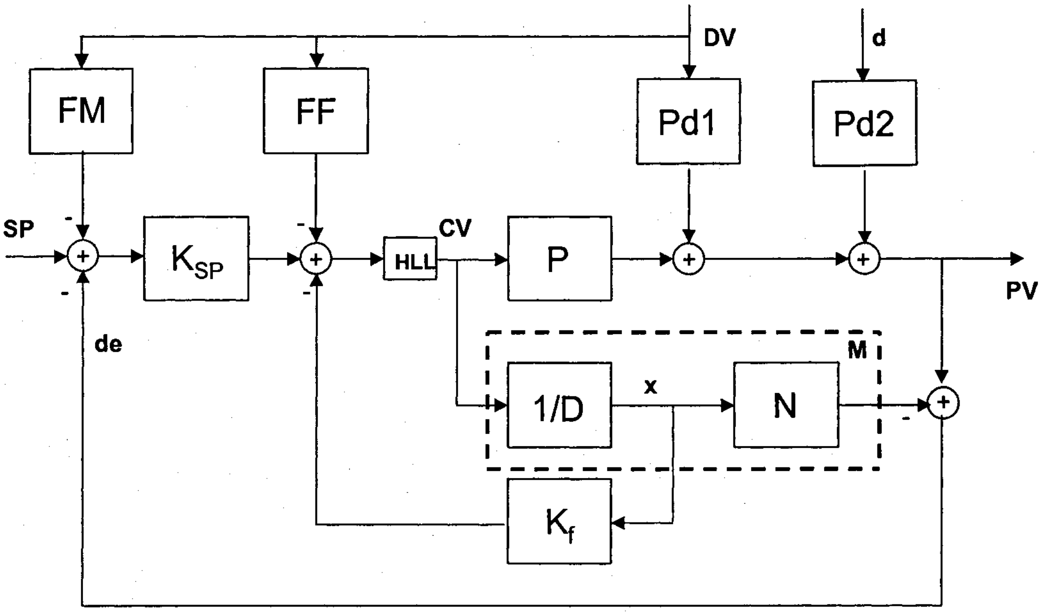
$$CV = K_f \cdot x + K_{SP}(SP - de) = \frac{\tau s + 1}{K(\varepsilon s + 1)}(SP - de) \tag{A17}$$

Once the feedback portion of the controller has been developed using these equations, the feed forward model is implemented to compensate for measured process disturbances. The feed forward FOPDT transfer function depends only on the process model (M) and measured process disturbance (Pd1) parameters. The process disturbance parameters are found in the same manner as the model parameters, and are in the form of Equation A10. Equation A18 gives the calculation for the feed forward compensator.

$$FF = \frac{Pd1}{P} \tag{A18}$$

For ratios of Pd1/M, where the deadtime of the feedback signal is longer than the deadtime of the feed forward signal, the model predictive feedback and feed forward components compensate for the same process disturbances. Therefore, the FM compensator must be included in the control scheme to bias the set point during transient responses to a process disturbance. Equation A19 gives the calculation for FM in this case. For instances where the deadtime of the feed forward signal is longer than the deadtime for the feedback signal the FM compensator is zero. Figure A25 shows the final MSF controller implementation.

$$\begin{aligned} FM &= Pd1 - FF \cdot P \qquad &&\text{for } \theta_P > \theta_{Pd1} \\ FM &= 0 && \theta_P < \theta_{Pd1} \end{aligned} \tag{A19}$$



where,

K_{SP}:	set point gain
K_f:	state multiplier
P:	physical plant process to be controlled
N, D:	process model numerator and denominator components
FM:	compensator
FF:	feed forward function
HLL:	high-low limiter
Pd1:	measured process disturbance lag
Pd2:	unmeasured process disturbance lag

Figure A25. Final MSF controller implementation.

It should be noted here that for the MSF controller to generate physically realizable control variable responses, it must satisfy the following criteria (Rivera, 1999b):

1. The controller must be stable. It must generate bounded responses to bounded inputs. To achieve this criterion all poles of the MSF controller must lie in the open left-half plane. The first-order MSF controller developed in this study contains no right-half plane poles ($\epsilon > 0$, $K > 0$) and is therefore stable. The process under control is also a first-order, self-regulating system that is inherently stable.

2. The controller must be proper (either strictly proper or semi-proper), which avoids pure differentiation of signals. The MSF controller in this study meets this criterion because the limit as s approaches infinity of the controller is greater than zero, and the transfer function denominator order is equal to the numerator order (see Equation A11). Thus the controller is semi-proper.

3. The controller must be causal, relying on previous and current plant measurements only, and not requiring prediction of future events. Using the controller function given in Equation A11 to approximate the inverse of the deadtime component in the FOPDT process meets this condition.

The MSF controller developed for this study meets the requirements for being stable, proper, and causal. The controller is therefore capable of generating physically realizable control responses to set point changes and process disturbances.

Tuning of the controller is reduced to selecting the desired transition speed between set points under closed loop conditions (setting the value of ϵ). The maximum rate of change for the control effort may also be selected. The maximum rate of change is used as a safe guard against possible spikes in the control effort due to inaccurate sensor readings or other sudden changes to which the controller should not over react. The controller is tuned and the maximum rate of change is selected such that the rates of change limits are imposed only when these anomalies occur. Mhatre and Brosilow (n.d.), and Stryczek (1996) discuss several other algorithmic and heuristic methods for determining the value of ϵ and tuning the MSF controller.

APPENDIX B

MODEL STATE FEEDBACK PLC LADDER LOGIC

There are three associated ladder logic code routines, the main routine, input routine, and output routine. These routines perform a number of the necessary MPC functions. These functions are:

1. Calculates the model parameters of 1/D, N, Kf, Ksp, FF, FM based on the user-entered system gain, deadtime, and time constant. These parameters are calculated each controller scan (5 second intervals) prior to execution of the control algorithm.

2. Provides bumpless control transfer between the following control modes:
   a. Automatic – Calculates the modulating steam valve position based on temperature feedback and feed rate feed forward signals to control to the cooking temperature set point.
   b. Manual – The operator sets the steam valve position.
   c. Tracking – The model predictive controller tracks the actual steam valve position determined by another source (operator or other algorithm) to provide for bumpless transfer back to automatic mode when selected.

3. Initializes the controller after a processor restart or modifications in the program, such as adjustments to the gain, deadtime, or time constant.

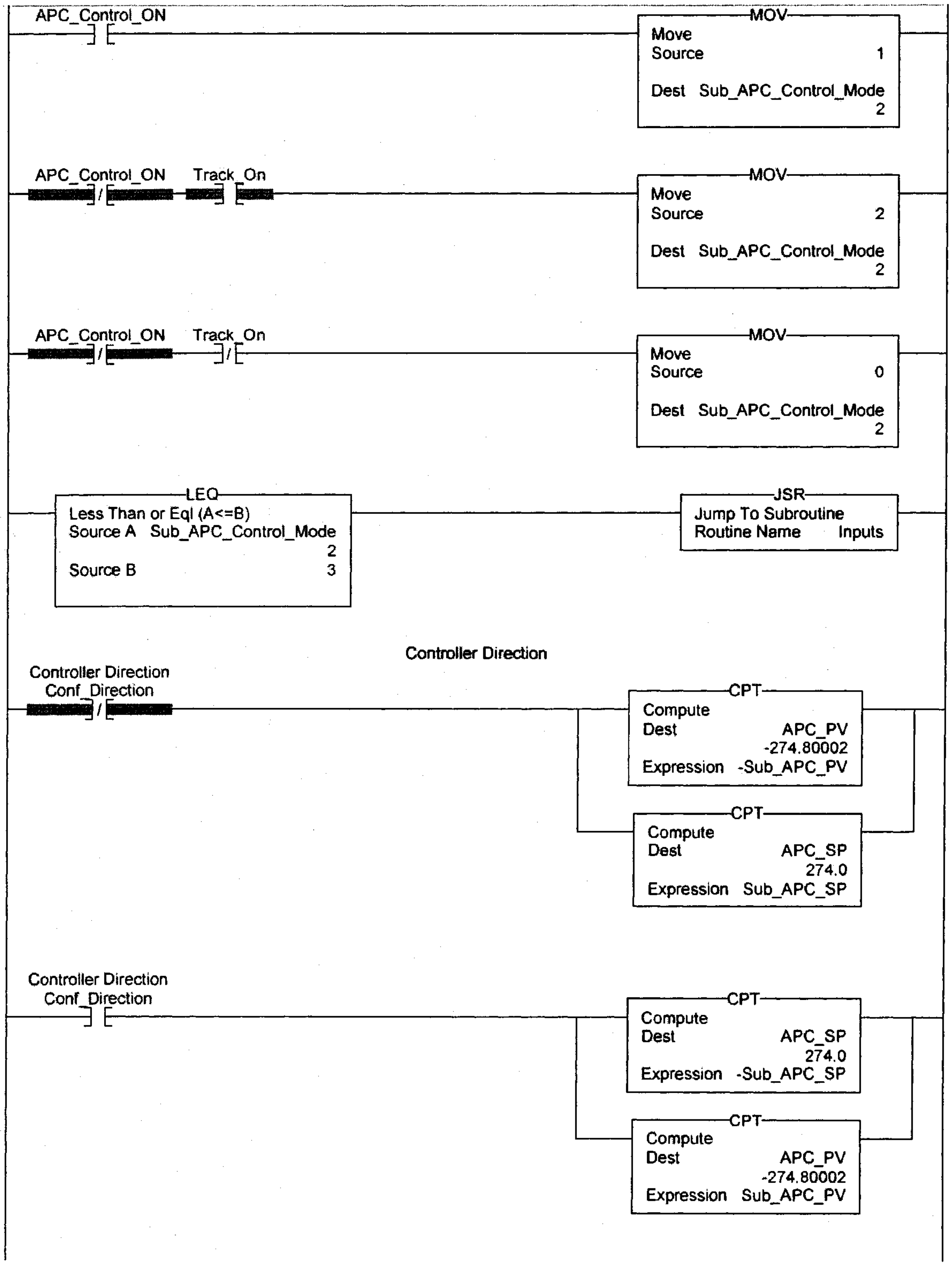
4. Maintains the system heartbeat timer between the processor and other relevant systems (I/O, communications, and so forth). If the heartbeat fails then a communications alarm is set and the control defaults to a PID control mode.

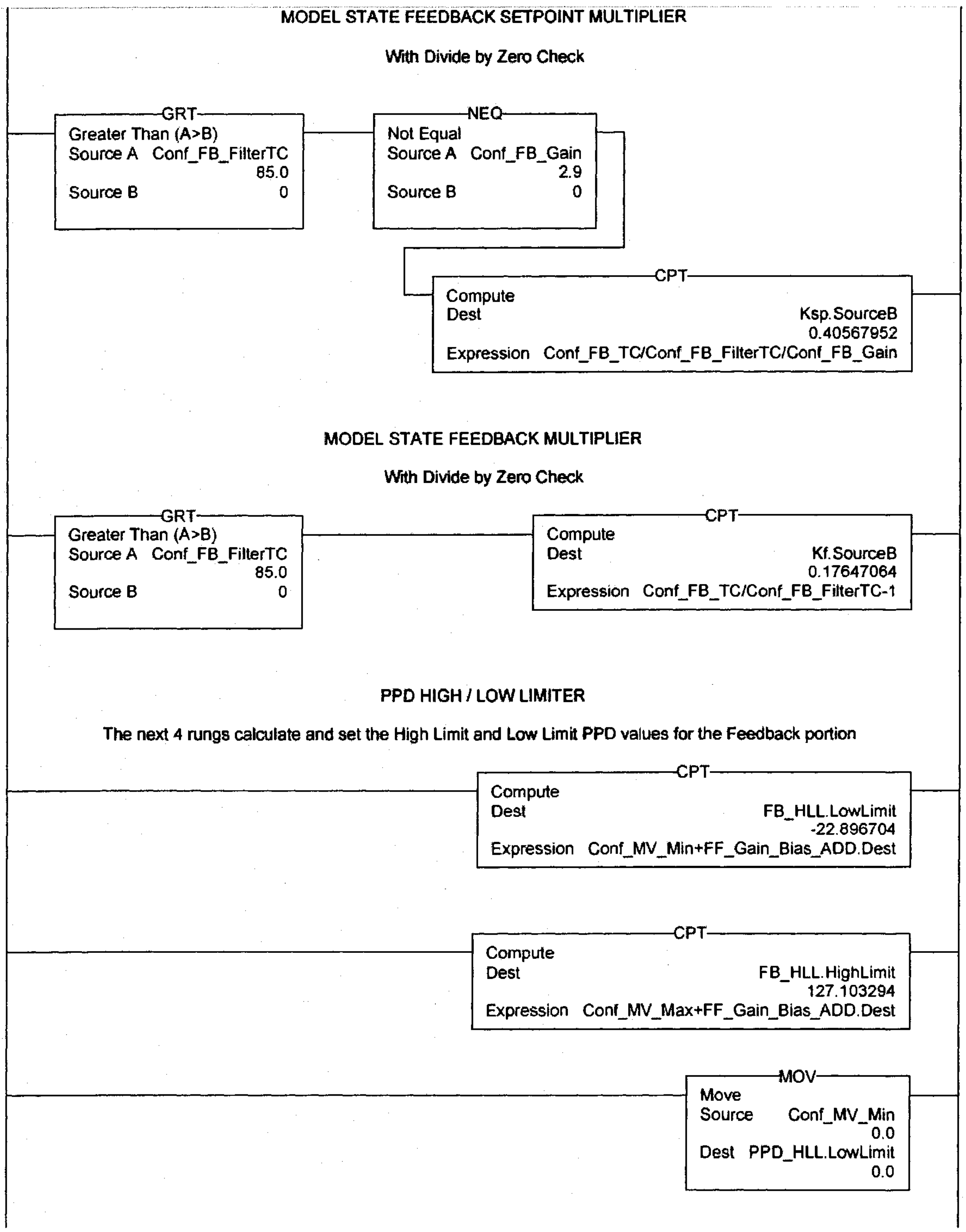
5. Communicates to the system inputs and outputs (I/O).

6. Provides fault handling.

7. Calls the controller deterministic function block diagram.

Main Routine





MODEL STATE FEEDBACK SETPOINT MULTIPLIER

With Divide by Zero Check

```
GRT
Greater Than (A>B)
Source A   Conf_FB_FilterTC
                       85.0
Source B                  0

NEQ
Not Equal
Source A   Conf_FB_Gain
                    2.9
Source B              0

CPT
Compute
Dest                                     Ksp.SourceB
                                          0.40567952
Expression   Conf_FB_TC/Conf_FB_FilterTC/Conf_FB_Gain
```

MODEL STATE FEEDBACK MULTIPLIER

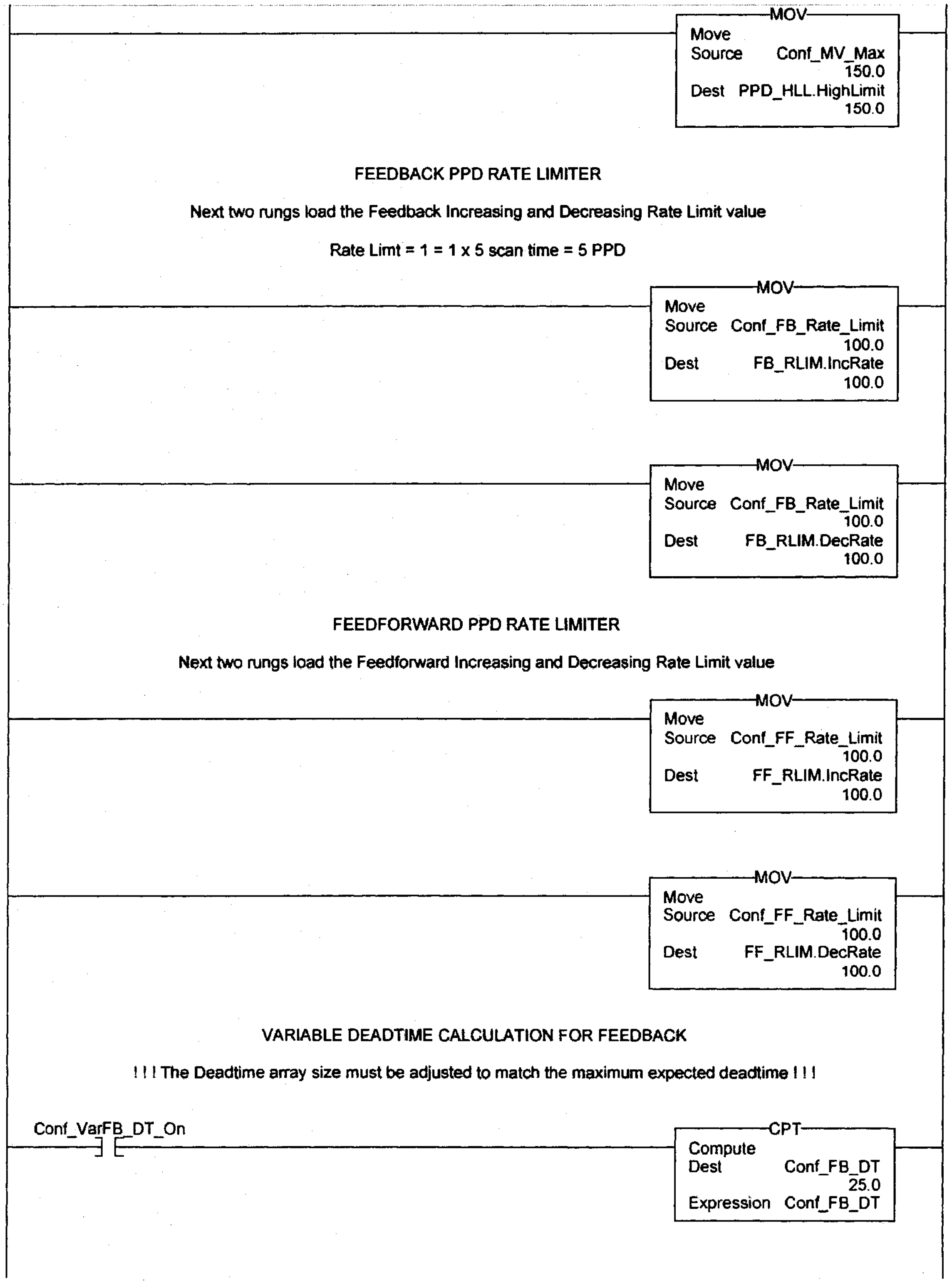
With Divide by Zero Check

```
GRT
Greater Than (A>B)
Source A   Conf_FB_FilterTC
                       85.0
Source B                  0

CPT
Compute
Dest                           Kf.SourceB
                               0.17647064
Expression   Conf_FB_TC/Conf_FB_FilterTC-1
```

PPD HIGH / LOW LIMITER

The next 4 rungs calculate and set the High Limit and Low Limit PPD values for the Feedback portion

```
CPT
Compute
Dest                            FB_HLL.LowLimit
                                     -22.896704
Expression   Conf_MV_Min+FF_Gain_Bias_ADD.Dest

CPT
Compute
Dest                           FB_HLL.HighLimit
                                     127.103294
Expression   Conf_MV_Max+FF_Gain_Bias_ADD.Dest

MOV
Move
Source        Conf_MV_Min
                      0.0
Dest   PPD_HLL.LowLimit
                      0.0
```



```
MOV
Move
Source      Conf_MV_Max
                  150.0
Dest  PPD_HLL.HighLimit
                  150.0
```

FEEDBACK PPD RATE LIMITER

Next two rungs load the Feedback Increasing and Decreasing Rate Limit value

Rate Limt = 1 = 1 x 5 scan time = 5 PPD

```
MOV
Move
Source  Conf_FB_Rate_Limit
                     100.0
Dest       FB_RLIM.IncRate
                     100.0
```

```
MOV
Move
Source  Conf_FB_Rate_Limit
                     100.0
Dest       FB_RLIM.DecRate
                     100.0
```

FEEDFORWARD PPD RATE LIMITER

Next two rungs load the Feedforward Increasing and Decreasing Rate Limit value

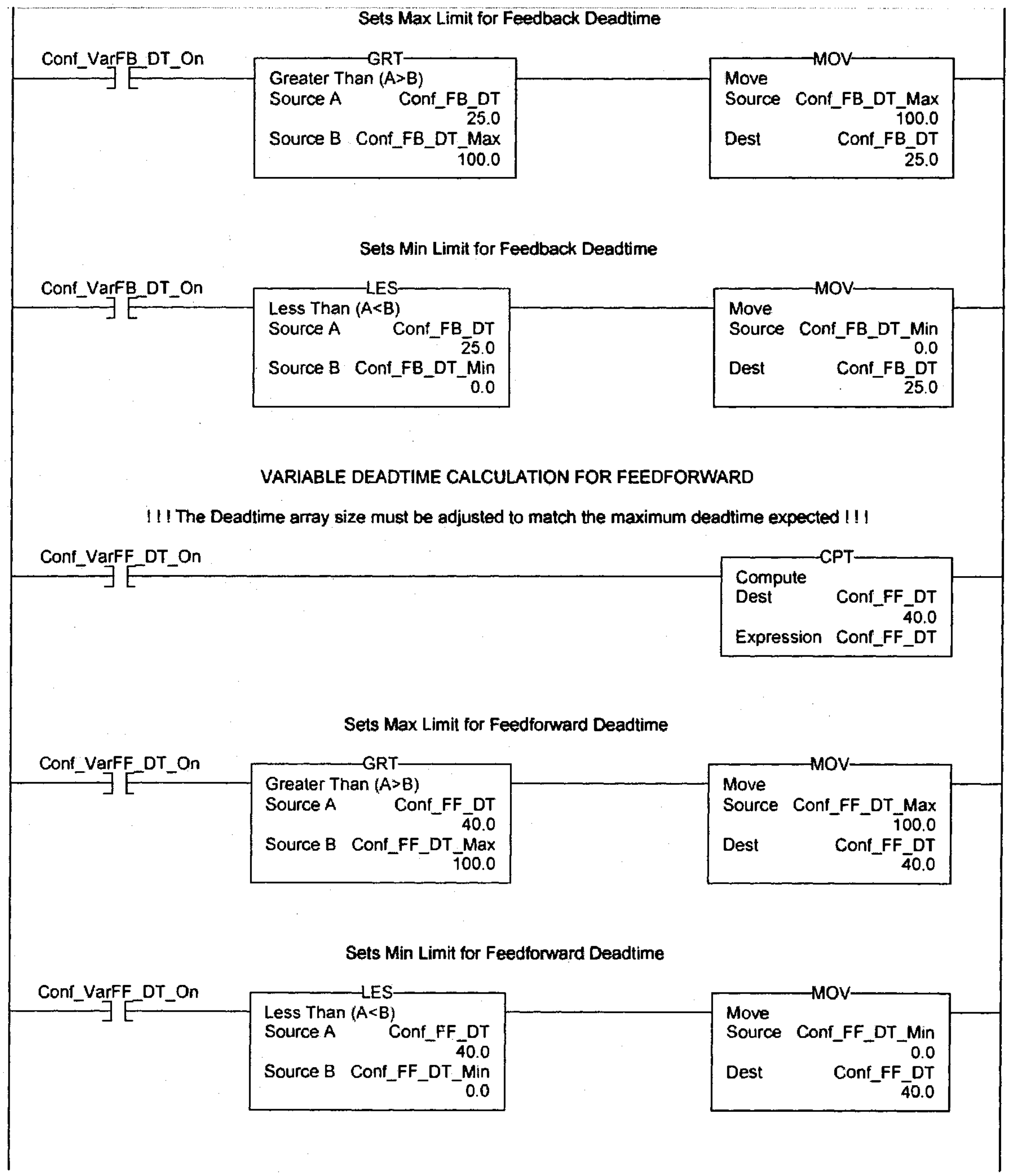
```
MOV
Move
Source  Conf_FF_Rate_Limit
                     100.0
Dest       FF_RLIM.IncRate
                     100.0
```

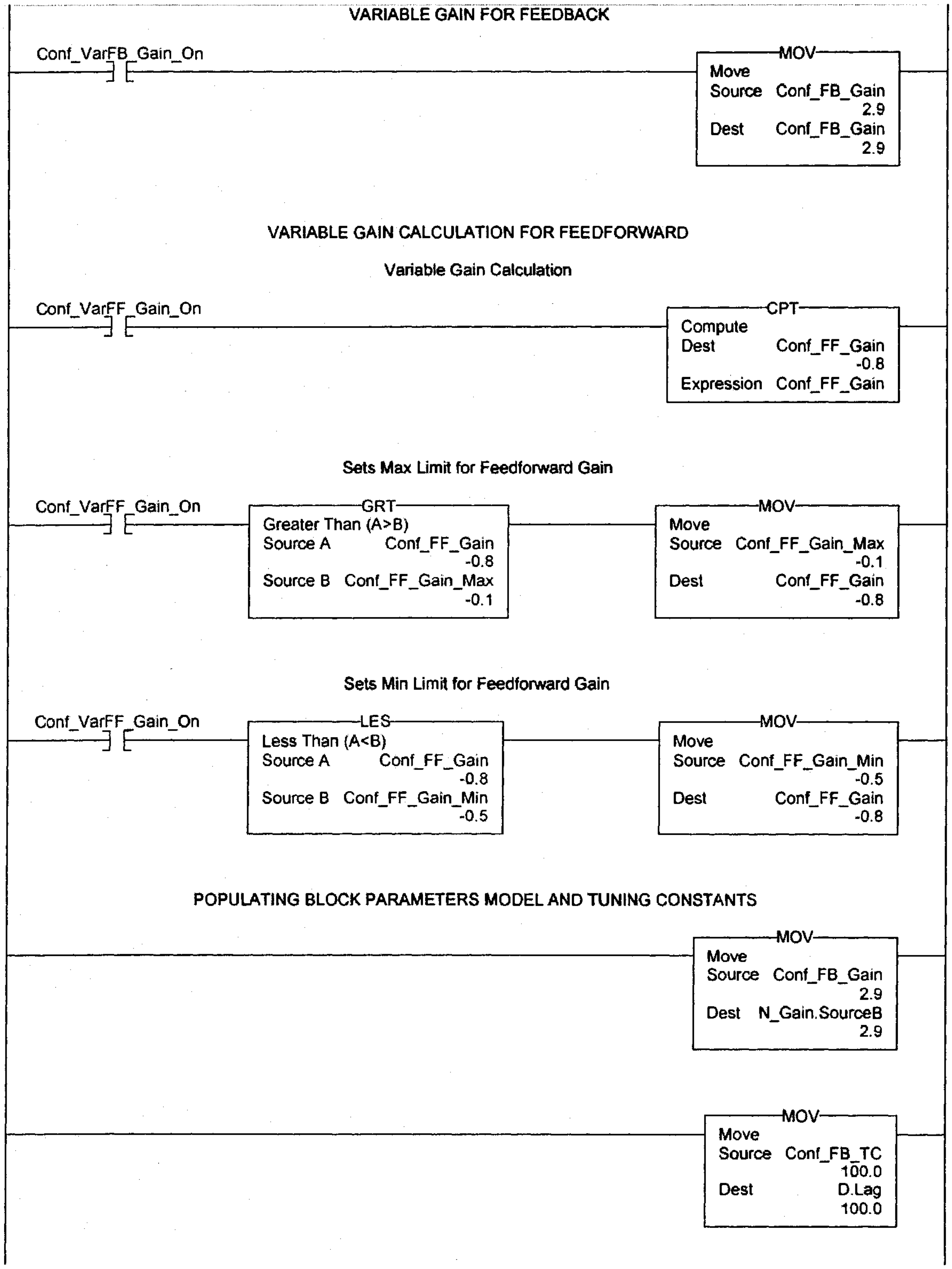
```
MOV
Move
Source  Conf_FF_Rate_Limit
                     100.0
Dest       FF_RLIM.DecRate
                     100.0
```

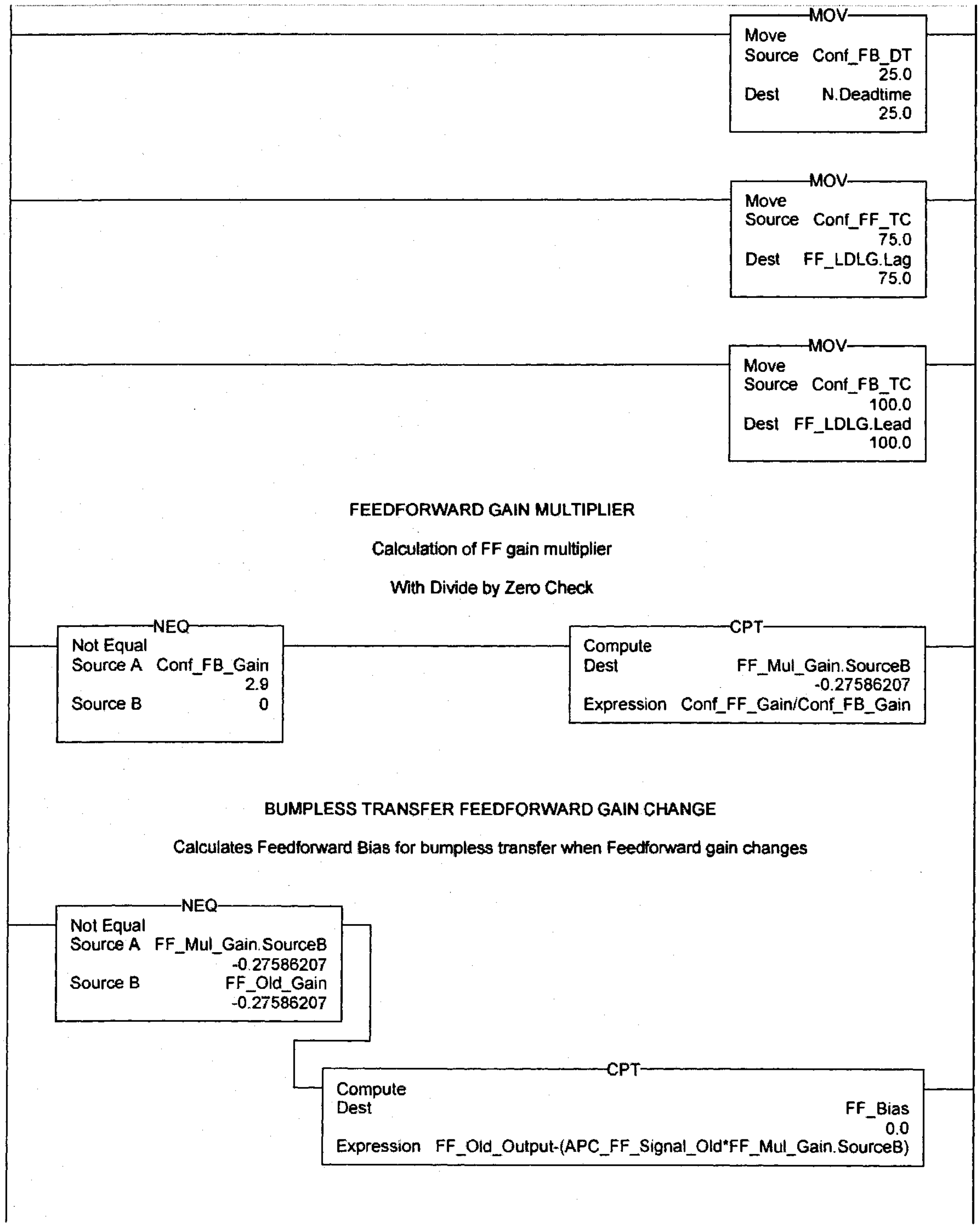
VARIABLE DEADTIME CALCULATION FOR FEEDBACK

! ! ! The Deadtime array size must be adjusted to match the maximum expected deadtime ! ! !

```
Conf_VarFB_DT_On

CPT
Compute
Dest          Conf_FB_DT
                    25.0
Expression    Conf_FB_DT
```


Sets Max Limit for Feedback Deadtime
Conf_VarFB_DT_On
GRT
Greater Than (A>B)
Source A Conf_FB_DT
25.0
Source B Conf_FB_DT_Max
100.0
MOV
Move
Source Conf_FB_DT_Max
100.0
Dest Conf_FB_DT
25.0
Sets Min Limit for Feedback Deadtime
Conf_VarFB_DT_On
LES
Less Than (A<B)
Source A Conf_FB_DT
25.0
Source B Conf_FB_DT_Min
0.0
MOV
Move
Source Conf_FB_DT_Min
0.0
Dest Conf_FB_DT
25.0
VARIABLE DEADTIME CALCULATION FOR FEEDFORWARD
! ! ! The Deadtime array size must be adjusted to match the maximum deadtime expected ! ! !
Conf_VarFF_DT_On
CPT
Compute
Dest Conf_FF_DT
40.0
Expression Conf_FF_DT
Sets Max Limit for Feedforward Deadtime
Conf_VarFF_DT_On
GRT
Greater Than (A>B)
Source A Conf_FF_DT
40.0
Source B Conf_FF_DT_Max
100.0
MOV
Move
Source Conf_FF_DT_Max
100.0
Dest Conf_FF_DT
40.0
Sets Min Limit for Feedforward Deadtime
Conf_VarFF_DT_On
LES
Less Than (A<B)
Source A Conf_FF_DT
40.0
Source B Conf_FF_DT_Min
0.0
MOV
Move
Source Conf_FF_DT_Min
0.0
Dest Conf_FF_DT
40.0


VARIABLE GAIN FOR FEEDBACK
Conf_VarFB_Gain_On
MOV
Move
Source Conf_FB_Gain
2.9
Dest Conf_FB_Gain
2.9
VARIABLE GAIN CALCULATION FOR FEEDFORWARD
Variable Gain Calculation
Conf_VarFF_Gain_On
CPT
Compute
Dest Conf_FF_Gain
-0.8
Expression Conf_FF_Gain
Sets Max Limit for Feedforward Gain
Conf_VarFF_Gain_On
GRT
Greater Than (A>B)
Source A Conf_FF_Gain
-0.8
Source B Conf_FF_Gain_Max
-0.1
MOV
Move
Source Conf_FF_Gain_Max
-0.1
Dest Conf_FF_Gain
-0.8
Sets Min Limit for Feedforward Gain
Conf_VarFF_Gain_On
LES
Less Than (A<B)
Source A Conf_FF_Gain
-0.8
Source B Conf_FF_Gain_Min
-0.5
MOV
Move
Source Conf_FF_Gain_Min
-0.5
Dest Conf_FF_Gain
-0.8
POPULATING BLOCK PARAMETERS MODEL AND TUNING CONSTANTS
MOV
Move
Source Conf_FB_Gain
2.9
Dest N_Gain.SourceB
2.9
MOV
Move
Source Conf_FB_TC
100.0
Dest D.Lag
100.0



```
MOV
Move
Source  Conf_FB_DT
              25.0
Dest    N.Deadtime
              25.0

MOV
Move
Source  Conf_FF_TC
              75.0
Dest    FF_LDLG.Lag
              75.0

MOV
Move
Source  Conf_FB_TC
             100.0
Dest    FF_LDLG.Lead
             100.0
```

FEEDFORWARD GAIN MULTIPLIER

Calculation of FF gain multiplier

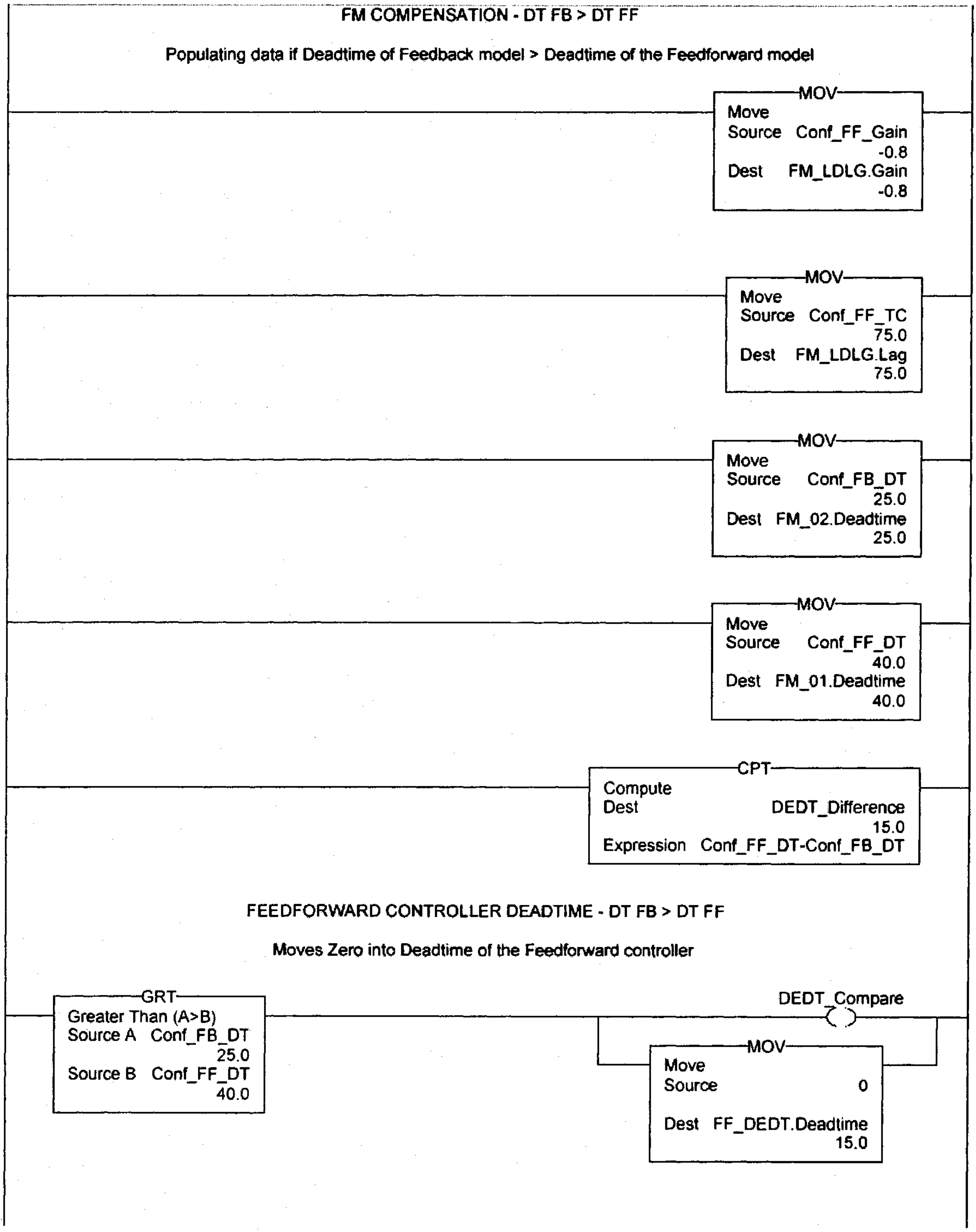
With Divide by Zero Check

```
NEQ
Not Equal
Source A  Conf_FB_Gain
                   2.9
Source B             0

CPT
Compute
Dest                FF_Mul_Gain.SourceB
                            -0.27586207
Expression  Conf_FF_Gain/Conf_FB_Gain
```

BUMPLESS TRANSFER FEEDFORWARD GAIN CHANGE

Calculates Feedforward Bias for bumpless transfer when Feedforward gain changes

```
NEQ
Not Equal
Source A  FF_Mul_Gain.SourceB
                  -0.27586207
Source B          FF_Old_Gain
                  -0.27586207

CPT
Compute
Dest                                                  FF_Bias
                                                          0.0
Expression  FF_Old_Output-(APC_FF_Signal_Old*FF_Mul_Gain.SourceB)
```



FM COMPENSATION - DT FB > DT FF

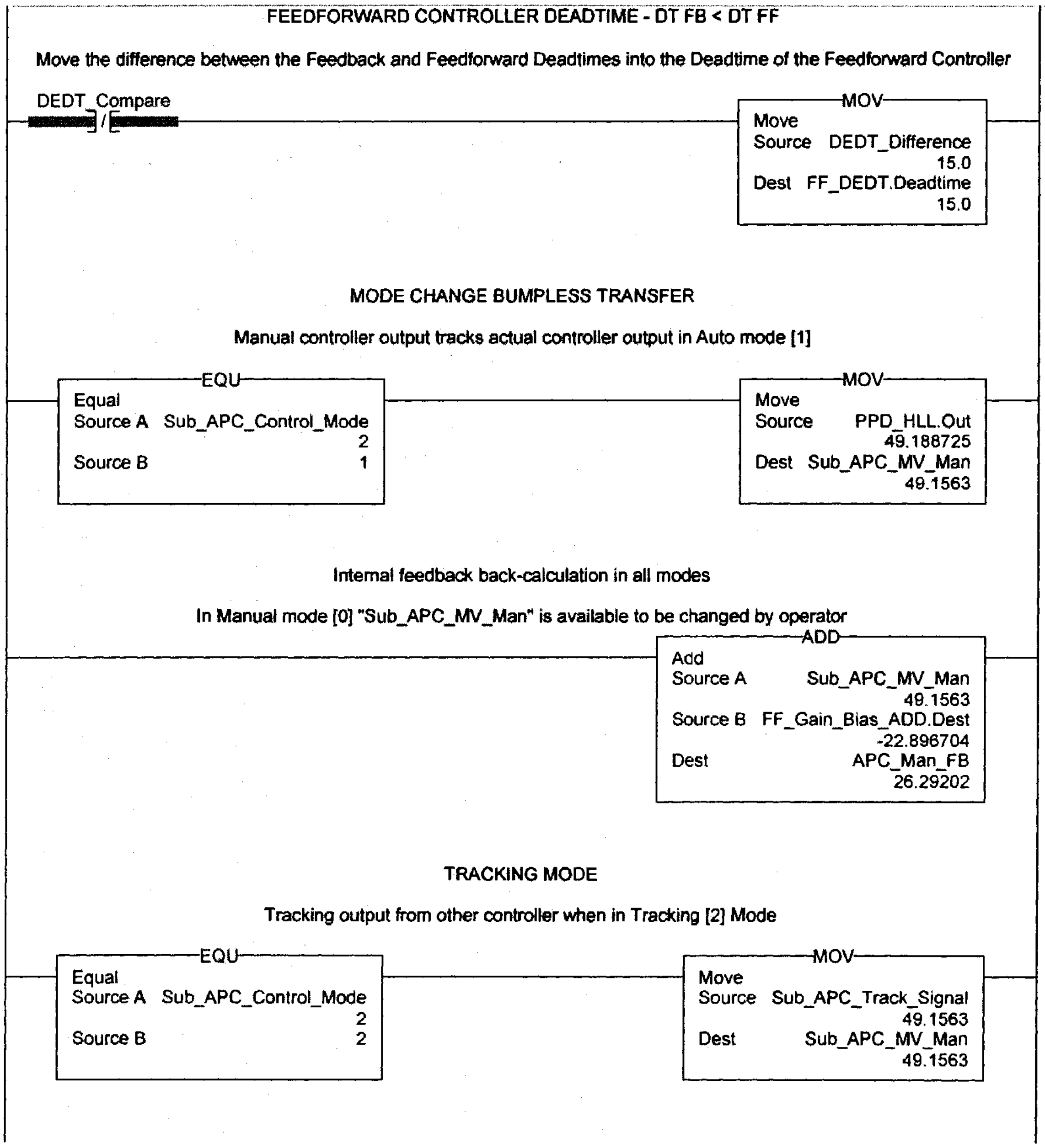
Populating data if Deadtime of Feedback model > Deadtime of the Feedforward model

```
MOV
Move
Source  Conf_FF_Gain
                -0.8
Dest    FM_LDLG.Gain
                -0.8

MOV
Move
Source  Conf_FF_TC
              75.0
Dest    FM_LDLG.Lag
              75.0

MOV
Move
Source     Conf_FB_DT
                 25.0
Dest  FM_02.Deadtime
                 25.0

MOV
Move
Source     Conf_FF_DT
                 40.0
Dest  FM_01.Deadtime
                 40.0

CPT
Compute
Dest                 DEDT_Difference
                                15.0
Expression  Conf_FF_DT-Conf_FB_DT
```

FEEDFORWARD CONTROLLER DEADTIME - DT FB > DT FF

Moves Zero into Deadtime of the Feedforward controller

```
GRT
Greater Than (A>B)
Source A  Conf_FB_DT
                25.0
Source B  Conf_FF_DT
                40.0

DEDT_Compare
( )

MOV
Move
Source                 0

Dest  FF_DEDT.Deadtime
                   15.0
```



FEEDFORWARD CONTROLLER DEADTIME - DT FB < DT FF

Move the difference between the Feedback and Feedforward Deadtimes into the Deadtime of the Feedforward Controller

```
DEDT_Compare                                   MOV
--] / [-------------------------------  Move
                                        Source  DEDT_Difference
                                                          15.0
                                        Dest  FF_DEDT.Deadtime
                                                          15.0
```

MODE CHANGE BUMPLESS TRANSFER

Manual controller output tracks actual controller output in Auto mode [1]

```
              EQU                                    MOV
Equal                                  Move
Source A  Sub_APC_Control_Mode         Source       PPD_HLL.Out
                             2                        49.188725
Source B                     1         Dest   Sub_APC_MV_Man
                                                        49.1563
```

Internal feedback back-calculation in all modes

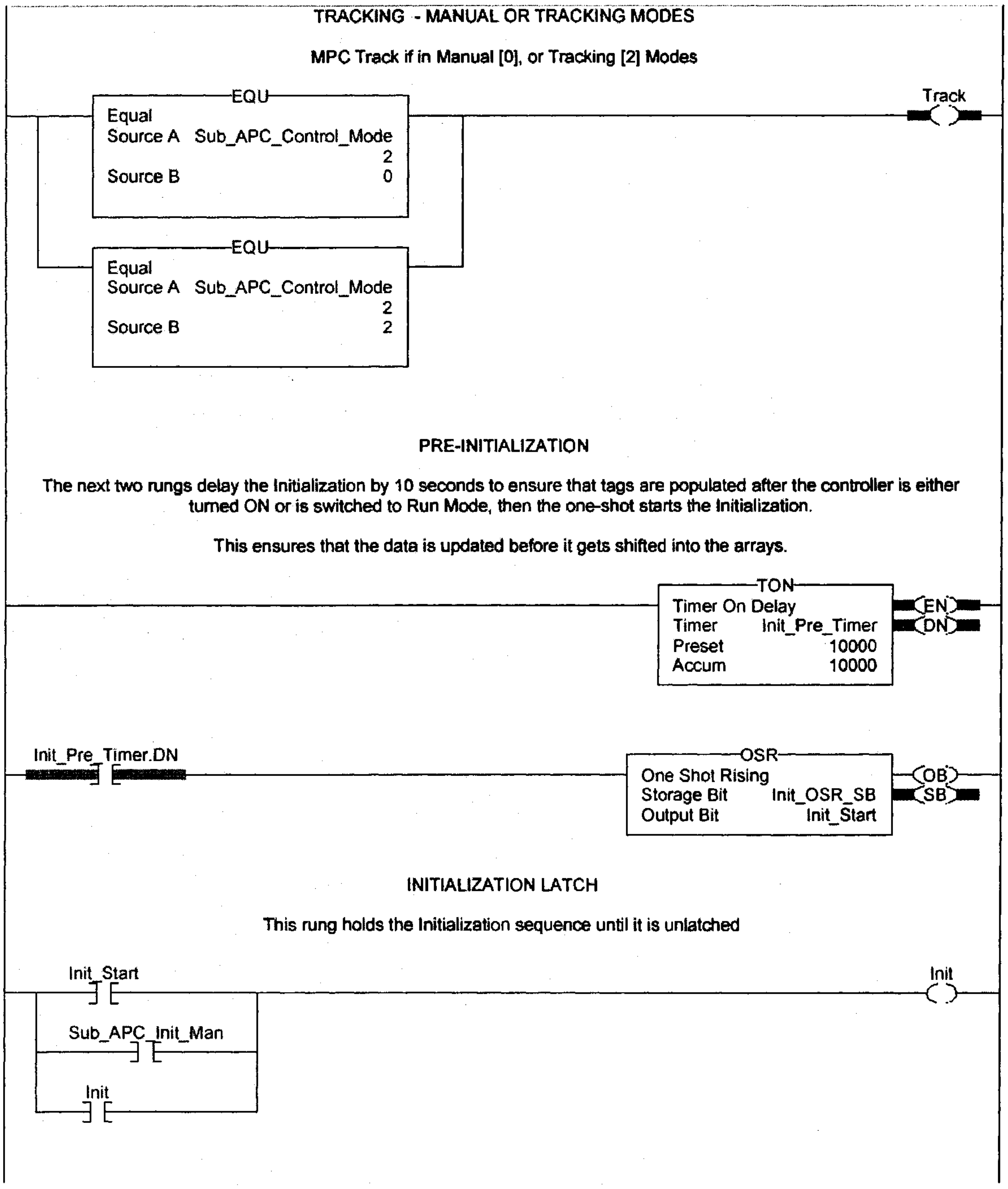
In Manual mode [0] "Sub_APC_MV_Man" is available to be changed by operator

```
                                   ADD
                     Add
                     Source A          Sub_APC_MV_Man
                                              49.1563
                     Source B  FF_Gain_Bias_ADD.Dest
                                           -22.896704
                     Dest                  APC_Man_FB
                                             26.29202
```

TRACKING MODE

Tracking output from other controller when in Tracking [2] Mode

```
              EQU                                    MOV
Equal                                  Move
Source A  Sub_APC_Control_Mode         Source  Sub_APC_Track_Signal
                             2                              49.1563
Source B                     2         Dest          Sub_APC_MV_Man
                                                            49.1563
```



TRACKING - MANUAL OR TRACKING MODES

MPC Track if in Manual [0], or Tracking [2] Modes

PRE-INITIALIZATION

The next two rungs delay the Initialization by 10 seconds to ensure that tags are populated after the controller is either turned ON or is switched to Run Mode, then the one-shot starts the Initialization.

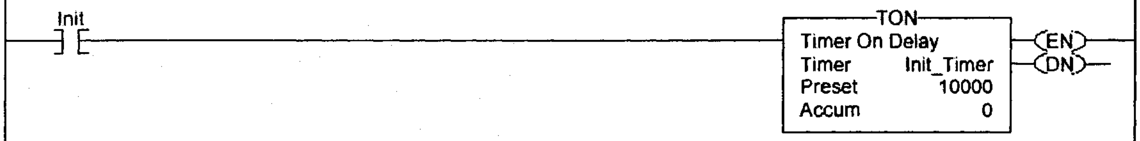
This ensures that the data is updated before it gets shifted into the arrays.

INITIALIZATION LATCH

This rung holds the Initialization sequence until it is unlatched

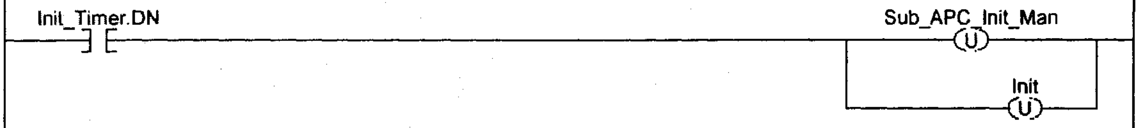
INITIALIZATION TIMER

The next two rungs create an Init pulse 10 seconds for Init Manual.
This ensures that the data is updated before it gets shifted into the arrays.



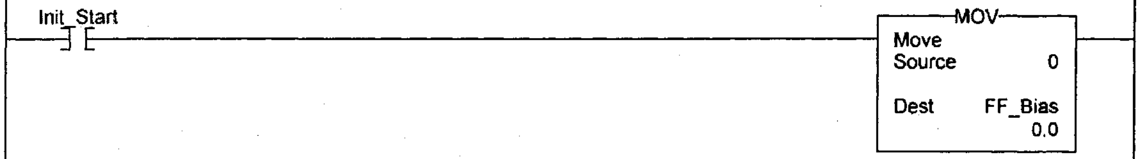
INITIALIZATION UN-LATCH

This rung un-latches the Initialization sequence



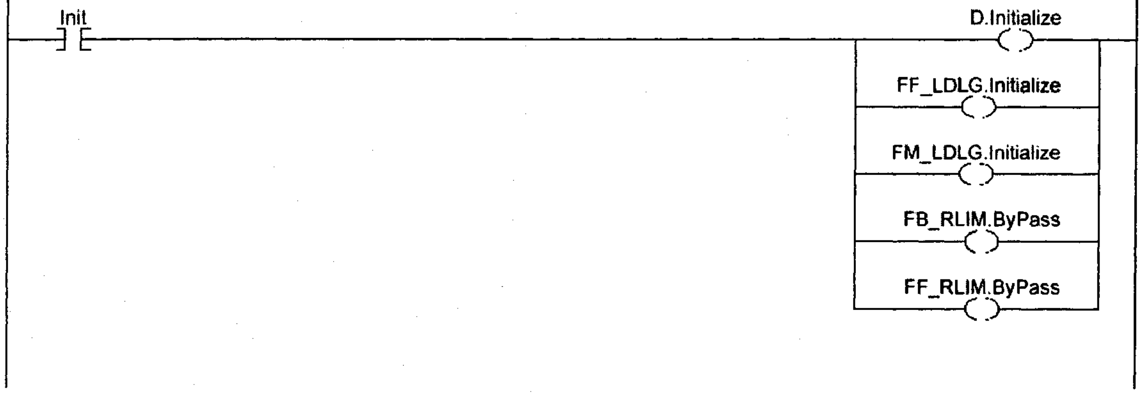
FF BIAS ZERO INITIALIZATION

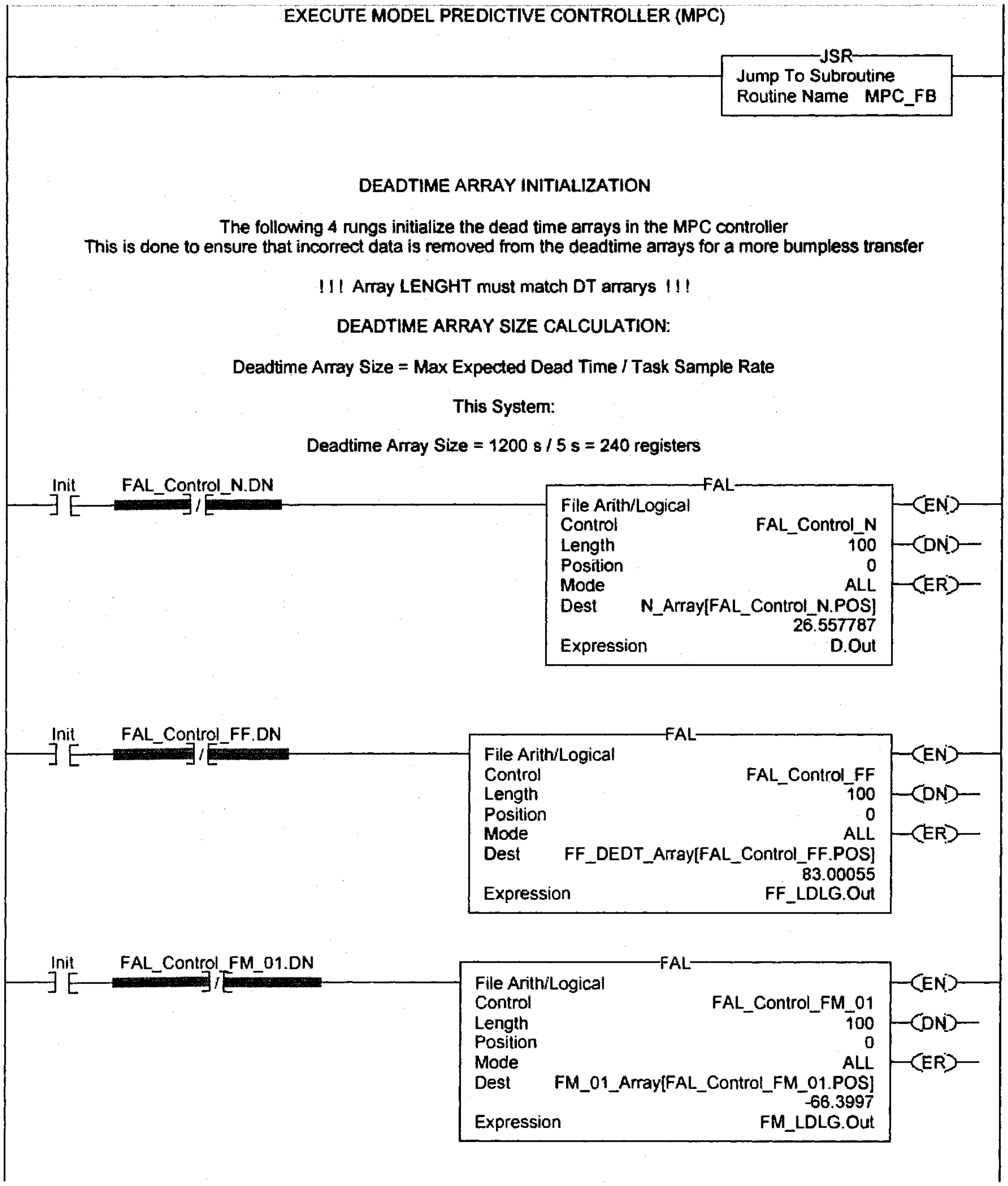
This rung initializes the Feedforward Bias on Init_Start only



LDLG AND RATE LIMITER INITIALIZATION

This rung initializes the Lead / Lag and Rate Limit blocks making the Output = Input in the MPC Controller





EXECUTE MODEL PREDICTIVE CONTROLLER (MPC)

JSR
Jump To Subroutine
Routine Name MPC_FB

DEADTIME ARRAY INITIALIZATION

The following 4 rungs initialize the dead time arrays in the MPC controller
This is done to ensure that incorrect data is removed from the deadtime arrays for a more bumpless transfer

! ! ! Array LENGHT must match DT arrarys ! ! !

DEADTIME ARRAY SIZE CALCULATION:

Deadtime Array Size = Max Expected Dead Time / Task Sample Rate

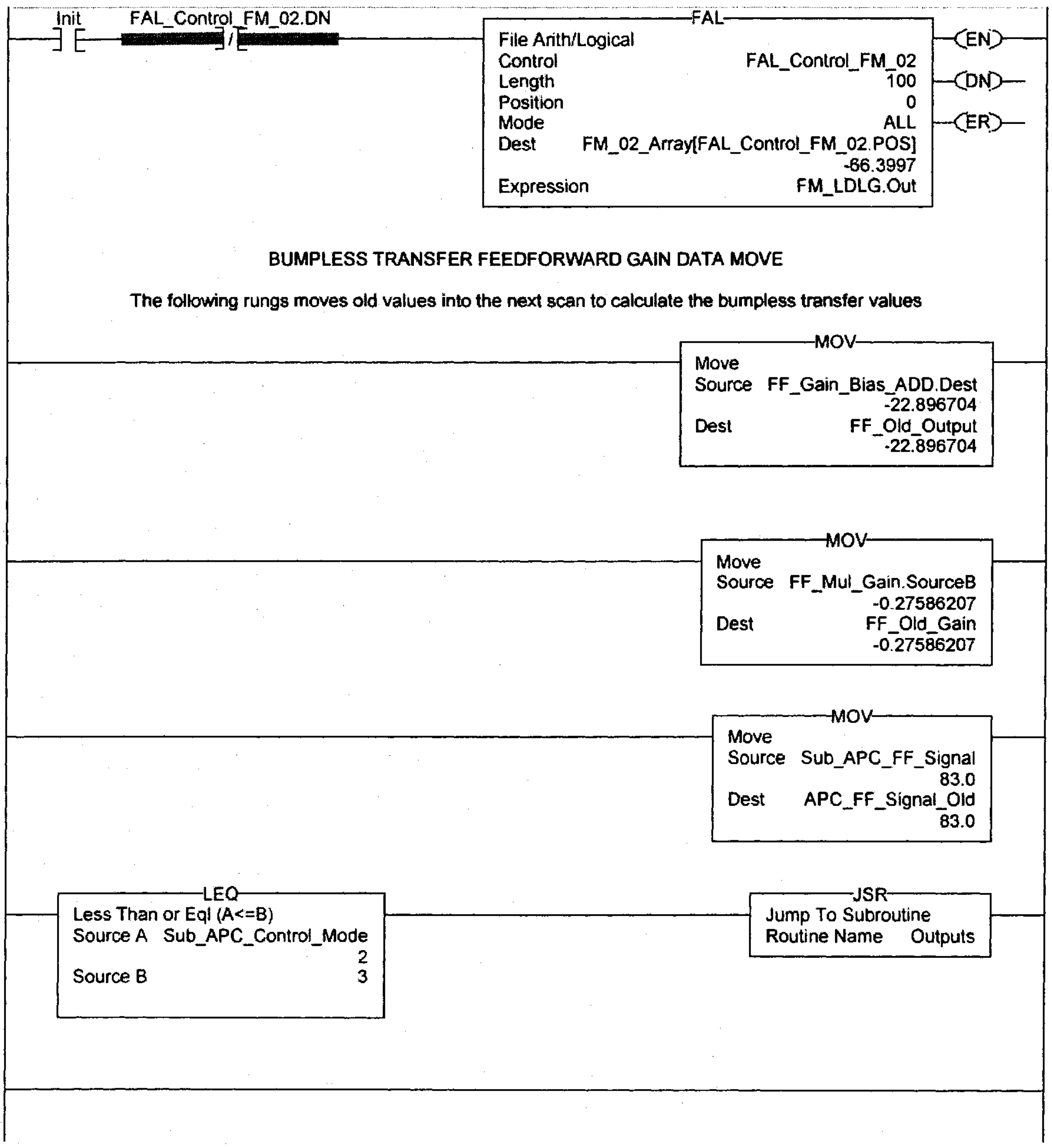
This System:

Deadtime Array Size = 1200 s / 5 s = 240 registers

FAL
File Arith/Logical
Control FAL_Control_N
Length 100
Position 0
Mode ALL
Dest N_Array[FAL_Control_N.POS]
26.557787
Expression D.Out

FAL
File Arith/Logical
Control FAL_Control_FF
Length 100
Position 0
Mode ALL
Dest FF_DEDT_Array[FAL_Control_FF.POS]
83.00055
Expression FF_LDLG.Out

FAL
File Arith/Logical
Control FAL_Control_FM_01
Length 100
Position 0
Mode ALL
Dest FM_01_Array[FAL_Control_FM_01.POS]
-66.3997
Expression FM_LDLG.Out



Init FAL_Control_FM_02.DN

FAL
File Arith/Logical
Control FAL_Control_FM_02
Length 100
Position 0
Mode ALL
Dest FM_02_Array[FAL_Control_FM_02.POS]
-66.3997
Expression FM_LDLG.Out

BUMPLESS TRANSFER FEEDFORWARD GAIN DATA MOVE

The following rungs moves old values into the next scan to calculate the bumpless transfer values

MOV
Move
Source FF_Gain_Bias_ADD.Dest
-22.896704
Dest FF_Old_Output
-22.896704

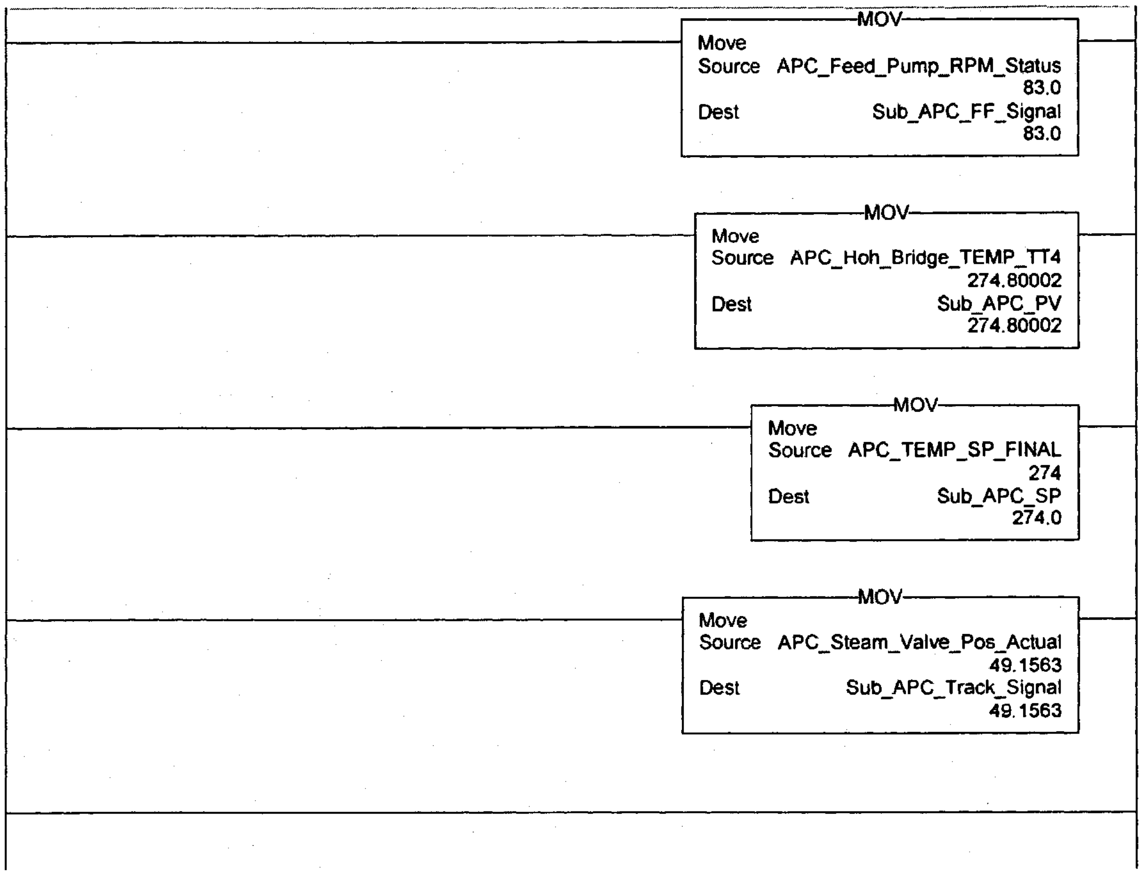
MOV
Move
Source FF_Mul_Gain.SourceB
-0.27586207
Dest FF_Old_Gain
-0.27586207

MOV
Move
Source Sub_APC_FF_Signal
83.0
Dest APC_FF_Signal_Old
83.0

LEQ
Less Than or Eql (A<=B)
Source A Sub_APC_Control_Mode
2
Source B 3

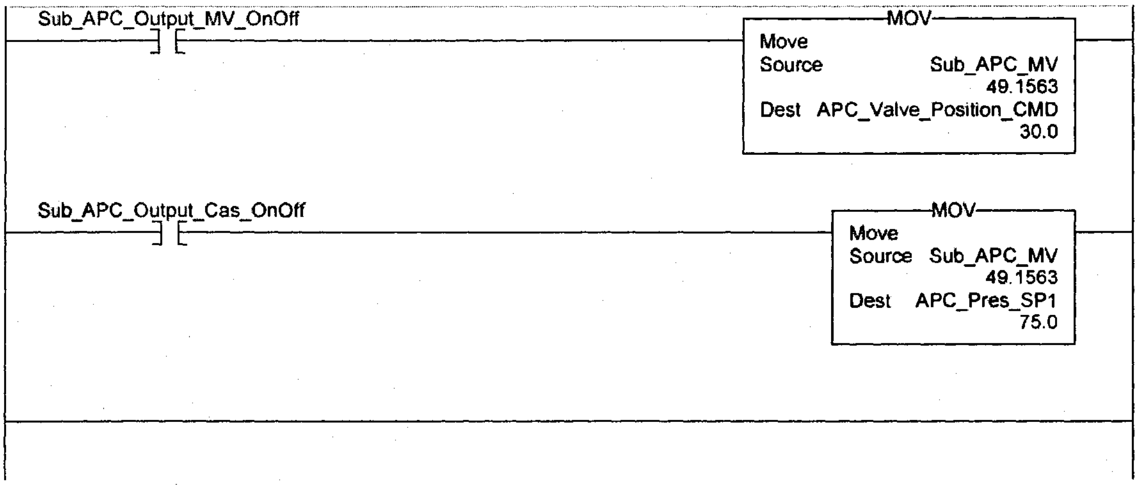
JSR
Jump To Subroutine
Routine Name Outputs

Input Subroutine



```
MOV
Move
Source  APC_Feed_Pump_RPM_Status
                            83.0
Dest           Sub_APC_FF_Signal
                            83.0

MOV
Move
Source  APC_Hoh_Bridge_TEMP_TT4
                       274.80002
Dest                  Sub_APC_PV
                       274.80002

MOV
Move
Source  APC_TEMP_SP_FINAL
                      274
Dest           Sub_APC_SP
                    274.0

MOV
Move
Source  APC_Steam_Valve_Pos_Actual
                           49.1563
Dest           Sub_APC_Track_Signal
                           49.1563
```

Output Subroutine



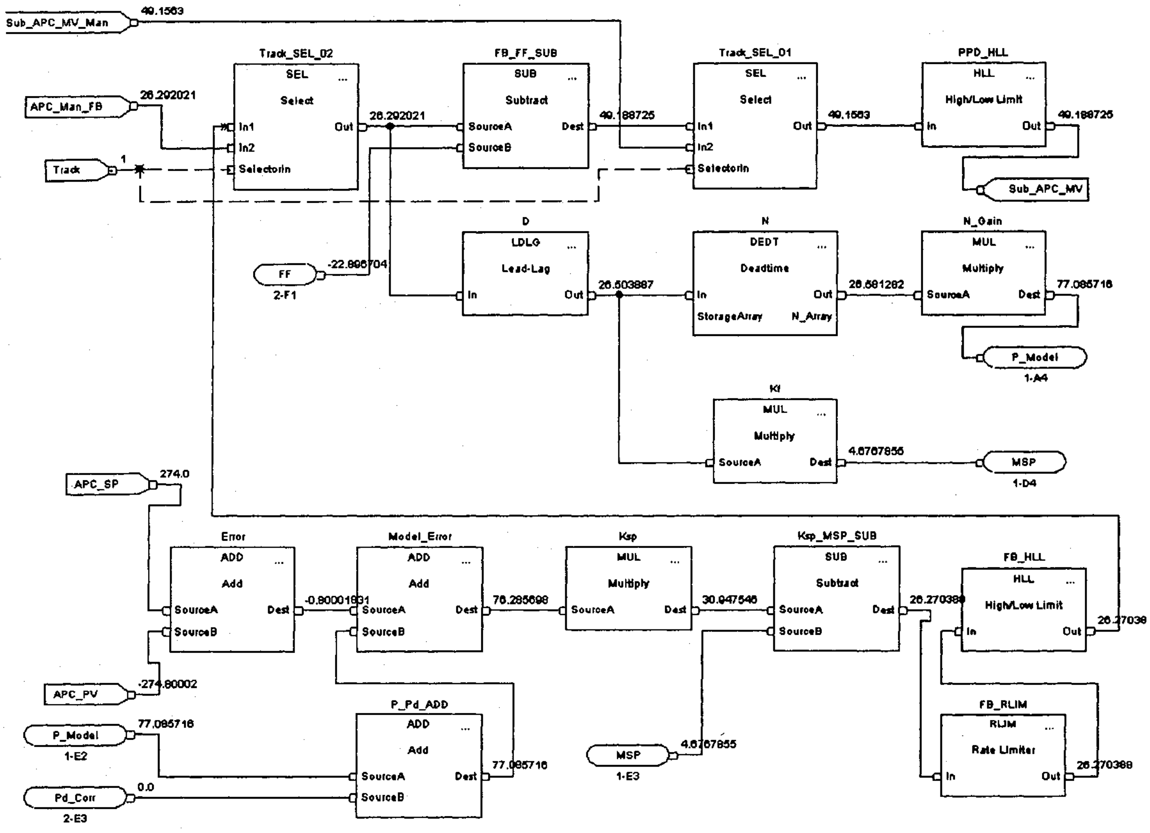
```
Sub_APC_Output_MV_OnOff
        ] [
MOV
Move
Source            Sub_APC_MV
                     49.1563
Dest  APC_Valve_Position_CMD
                        30.0

Sub_APC_Output_Cas_OnOff
        ] [
MOV
Move
Source  Sub_APC_MV
           49.1563
Dest    APC_Pres_SP1
              75.0
```

APPENDIX C

MODEL STATE FEEDBACK PLC FUNCTION BLOCKS

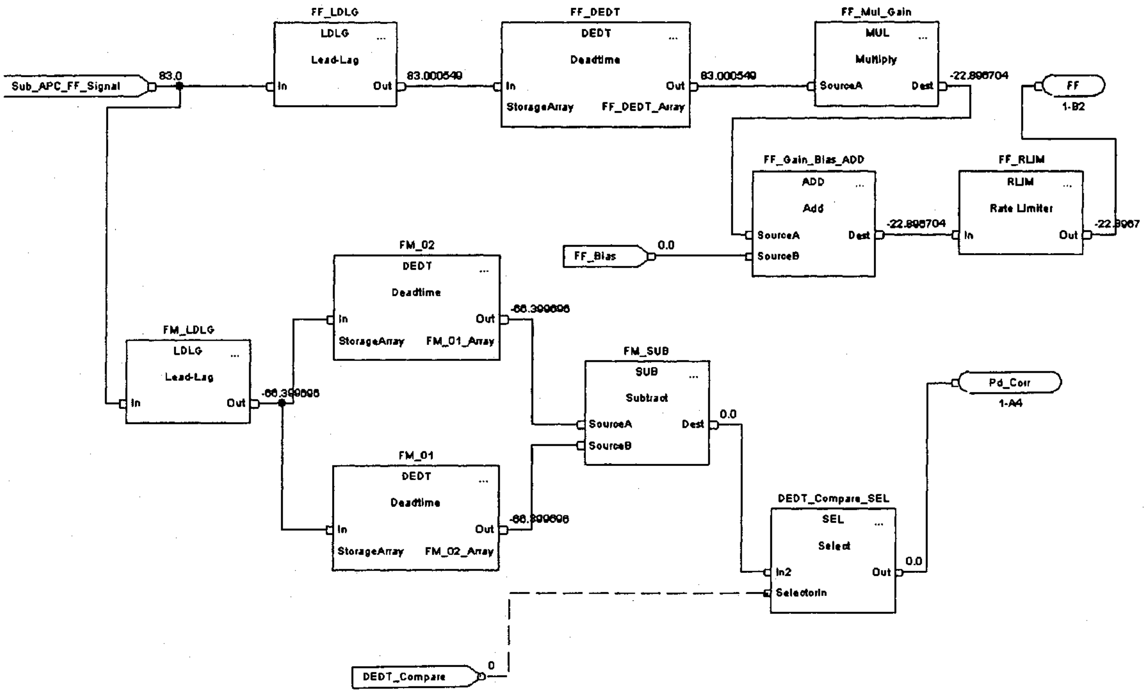
Model State Feedback Controller Function Block Subroutine

There are two function block code subroutines used in the MPC implementation. The first routine represents the model state feedback controller. The inputs to the controller routine are the temperature set point (APC_SP), the current temperature value (APC_PV), the feed forward component (FF), the feedback component (P_Model), and the FM contribution (Pd_Corr). The modulating steam valve percent open position (Sub_APC_MV) is the output from the controller routine. This value is sent directly to the output ladder logic routine for transfer to the valve.



Feed Forward Compensation Function Block Subroutine

The feed forward compensator is the second routine. It takes the product feed rate (Sub_APC_FF_Signal) as the primary input. The FM compensation switch (DEDT_Compare) is also used as an input. The outputs from the feed forward compensator are the feed forward signal (FF) and the FM contribution (Pd_Corr). Both signals are sent to the MSF controller subroutine to contribute to the final output to the modulating steam valve.



The key function blocks used to generate the model predictive controller were: Lead/Lag (LDLG), Deadtime (DEDT), Selector (SEL), Gain multiplication (MUL), Addition (ADD), Subtraction (SUB), and High/Low limit (HLL). These are all standard function blocks in the ControlLogix® PLC programming library.

APPENDIX D

SUMMARIZED TRIAL DATA FOR ANALYSIS OF VARIANCE

Table D12

Summary of Temperature Rise Time for Production Trials

Control Method	Rise Time (seconds)
PID Controller	1931
	1589
	1589
	1703
	1813
MANTRA® Controller	976
	808
	852
	852
	872
BrainWave® Controller	312
	312
	343
	349
	329
Process Perfecter® Controller	479
	496
	510
	526
	502
Model State Feedback Controller	416
	432
	347
	384
	401

Table D13

Summary of Temperature Overshoot for Production Trials

Control Method	Overshoot (°F)
PID Controller	4.4
	11.0
	8.0
	11.0
	8.7
MANTRA® Controller	0.4
	0.32
	0.1
	0.2
	0.18
BrainWave® Controller	1.0
	1.0
	1.9
	3.0
	1.7
Process Perfecter® Controller	0.6
	1.7
	1.4
	1.5
	1.7
Model State Feedback Controller	2.0
	2.9
	1.9
	2.3
	2.2

Table D14

Summary of Disturbance Rejection Trials for PID

Trial No.	Feed Rate (RPM)	Temperature Error (°F)
1	83	0.850586715
	93	2.256871645
	103	3.385865602
	83	2.963414972
	73	3.291371303
	63	3.171158723
	83	3.865528291
2	83	0.860403922
	93	1.651436312
	103	1.351937821
	83	2.297625016
	73	1.512637003
	63	2.664458795
	83	3.408226486
3	83	0.454345149
	93	0.998766581
	103	0.361718534
	83	1.128385059
	73	1.163041902
	63	0.909466283
	83	0.977871763
4	83	0.815302097
	93	2.075335697
	103	3.243902921
	83	2.737835765
	73	3.070804659
	63	2.972442518
	83	3.682607569

Table D15

Summary of Disturbance Rejection Trials for MANTRA®

Trial No.	Feed Rate (RPM)	Temperature Error (°F)
1	83	1.155396226
	93	0.987128708
	103	0.566589493
	83	0.833478039
	73	1.06149213
	63	1.714329868
	83	2.158712404
2	83	1.153130744
	93	0.984182056
	103	0.565313391
	83	0.831727035
	73	1.058067962
	63	1.709956577
	83	1.889957044
3	83	0.91723617
	93	0.833830376
	103	0.44046924
	83	0.700540081
	73	0.857261414
	63	1.416900456
	83	1.82710153
4	83	1.393556283
	93	1.140427041
	103	0.692709746
	83	0.966415998
	73	1.265722846
	63	2.01175928
	83	2.490323277

Table D16

Summary of Disturbance Rejection Trials for BrainWave®

Trial No.	Feed Rate (RPM)	Temperature Error (°F)
1	83	0.431635924
	93	0.522377763
	103	0.541143876
	83	0.927050158
	73	0.915396041
	63	1.348273417
	83	1.099177833
2	83	0.431279532
	93	0.521549906
	103	0.540282181
	83	0.925619525
	73	0.913740713
	63	1.346063132
	83	1.084668704
3	83	1.130779594
	93	0.801319181
	103	0.758057549
	83	1.636808811
	73	1.385973884
	63	1.06080535
	83	1.855554996
4	83	0.510562923
	93	0.633212015
	103	0.629302432
	83	1.10153759
	73	1.090975466
	63	1.568712779
	83	1.302871578

Table D17

Summary of Disturbance Rejection Trials for Process Perfecter®

Trial No.	Feed Rate (RPM)	Temperature Error (°F)
1	83	0.622641509
	93	0.83079922
	103	0.652099237
	83	0.65557554
	73	0.676
	63	1.016611296
	83	0.824952741
2	83	0.62173913
	93	0.829182879
	103	0.650857143
	83	0.654398564
	73	0.674650699
	63	1.014925373
	83	0.827580071
3	83	0.877829457
	93	0.610720887
	103	0.3536
	83	0.825966851
	73	0.657805907
	63	0.965432099
	83	0.839482759
4	83	0.567343336
	93	0.84040404
	103	0.335041322
	83	1.017723577
	73	1.586303191
	63	0.951547352
	83	0.85872891

Table D18

Summary of Disturbance Rejection Trials for Model State Feedback

Trial No.	Feed Rate (RPM)	Temperature Error (°F)
1	83	0.504412972
	93	0.517357719
	103	0.943076533
	83	1.954494728
	73	0.740339924
	63	0.843171872
	83	0.772836531
2	83	0.782781956
	93	0.515567554
	103	0.941008383
	83	1.948660416
	73	0.7377958
	63	0.840561433
	83	0.775498697
3	83	1.273370782
	93	0.735239937
	103	0.476781979
	83	1.060183866
	73	0.742489555
	63	0.967708057
	83	0.709143894
4	83	0.78000234
	93	0.398246469
	103	0.724226199
	83	0.878741597
	73	1.082839761
	63	0.78163012
	83	0.686333672