



University of Kentucky
UKnowledge

University of Kentucky Doctoral Dissertations

Graduate School

2004

ROBUST GENERIC MODEL CONTROL FOR PARAMETER INTERVAL SYSTEMS

Joseph Michael Istre

University of Kentucky, jmistr01@engr.uky.edu

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Istre, Joseph Michael, "ROBUST GENERIC MODEL CONTROL FOR PARAMETER INTERVAL SYSTEMS" (2004). *University of Kentucky Doctoral Dissertations*. 342.
https://uknowledge.uky.edu/gradschool_diss/342

This Dissertation is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Doctoral Dissertations by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF DISSERTATION

Joseph Michael Istre

The Graduate School
University of Kentucky

2004

ROBUST GENERIC MODEL CONTROL FOR PARAMETER INTERVAL SYSTEMS

ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Joseph Michael Istre

Lexington, Kentucky

Director: Dr. YuMing Zhang, Associate Professor of Electrical
Engineering

Lexington, Kentucky

2004

Copyright © Joseph Michael Istre, 2004

ABSTRACT OF DISSERTATION

ROBUST GENERIC MODEL CONTROL FOR PARAMETER INTERVAL SYSTEMS

A multivariable control technique is proposed for a type of nonlinear system with parameter intervals. The control is based upon the feedback linearization scheme called Generic Model Control, and alters the control calculation by utilizing parameter intervals, employing an adaptive step, averaging control predictions, and applying an interval problem solution. The proposed approach is applied in controlling both a linear and a nonlinear arc welding system as well in other simulations of scalar and multivariable systems.

KEYWORDS: Generic Model Control, Predictive Control, Arc Welding,
Parameter Interval, Robust Control

Joseph Michael Istre

July 6, 2004

ROBUST GENERIC MODEL CONTROL FOR
PARAMETER INTERVAL SYSTEMS

By

Joseph Michael Istre

YuMing Zhang

Director of Dissertation

YuMing Zhang

Director of Graduate Studies

July 6, 2004

RULES FOR THE USE OF DISSERTATIONS

Unpublished dissertations submitted for the Doctor's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

DISSERTATION

Joseph Michael Istre

The Graduate School

University of Kentucky

2004

ROBUST GENERIC MODEL CONTROL FOR
PARAMETER INTERVAL SYSTEMS

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Joseph Michael Istre

Lexington, Kentucky

Director: Dr. YuMing Zhang, Associate Professor of Electrical Engineering

Lexington, Kentucky

2004

Copyright © Joseph Michael Istre, 2004

To Gabi for walking with me past this milestone and to God for
showing me the way.

ACKNOWLEDGEMENTS

This research is funded by the National Science Foundation under Grant DMI-0114982 and the University of Kentucky Center for Manufacturing. I would like to thank Dr. YuMing Zhang for his guidance, encouragement and support. I am also grateful to Drs. Bruce L. Walcott, Alan T. Male, and Larry Holloway for their helpful insight and supervision. I would also like to thank Wei Lu and Yuchi Lui for their helpful advice throughout my experiments. In addition, I want to thank my parents Bob and Cheryl for their prayer and support and lastly my wife, Gabi, for being the most wonderful person in my life. Lastly, I want to give my ultimate gratitude to Jesus Christ for making this possible.

TABLE OF CONTENTS

ACNOWLEDGEMENTS.....	iii
LIST OF FIGURES	vii
LIST OF FILES	x
CHAPTER 1 INTRODUCTION.....	1
1.1 OBJECTIVES.....	1
1.2 PROPOSED NONLINEAR CONTROL APPLICATIONS.....	2
1.3 OUTLINE OF DISSERTATION.....	2
CHAPTER 2 REVIEW OF NONLINEAR MODELLING AND	
CONTROL.....	4
2.1 NONLINEAR MODELLING	5
2.2 NONLINEAR CONTROL	8
2.2.1 Overview.....	8
2.2.2 Feedback Linearization [20]	10
2.2.3 Model Predictive Control [22-24]	14
2.2.4 Backstepping [25]	18
2.2.5 H_∞ Nonlinear Control [26].....	20
2.2.6 Other Nonlinear Controllers.....	22
CHAPTER 3 GENERIC MODEL CONTROL	24
3.1 A REVIEW OF GENERIC MODEL CONTROL	24
3.1.1 A Nonlinear Modeling Approach for GMC.....	30

3.2	A NUMERICAL METHOD IN APPLYING GMC.....	32
3.3	GMC ROBUSTNESS AND STABILITY	35
3.3.1	<i>GMC Robustness Properties</i>	35
3.3.2	<i>GMC Stability Analysis</i>	36
3.4	FURTHER GMC THEORETICAL DEVELOPMENTS.....	37
3.4.1	<i>Relative Degree Considerations of GMC</i>	39
3.4.2	<i>Nonminimum Phase Systems</i>	41
3.5	FURTHER GMC IMPLEMENTATION ISSUES.....	42
3.6	COMPARING GMC TO OTHER CONTROLLERS	44
CHAPTER 4 PREDICTIVE GENERIC MODEL CONTROL.....		47
4.1	GMC CONTROL VARIATION.....	47
4.1.1	<i>Introduction</i>	47
4.1.2	<i>Offline Optimizer</i>	48
4.2	INTRODUCTION TO PGMC.....	52
4.3	ANALYZING PGMC	54
4.4	EXAMPLES.....	60
CHAPTER 5 PARAMETER INTERVAL ADAPTIVE PGMC.....		68
5.1	GMC PARAMETER INTERVAL SYSTEM MODEL	68
5.2	IDENTIFICATION OF PARAMETER INTERVALS.....	68
5.3	PGMC WITH INTERVAL FACTOR	69
5.4	EXPERIMENTAL RESULTS	71
CHAPTER 6 ROBUST PARAMETER INTERVAL PGMC		77
6.1	INTRODUCTION	77

6.2	INTERVAL MATHEMATICS [72, 73]	77
6.3	THE SISO CONTROL ALGORITHM	78
6.3.1	<i>Finding the Optimum Control and Stability Interval</i>	78
6.3.2	<i>Inducing Control Smoothness</i>	88
6.4	SISO EXPERIMENTAL RESULTS	91
6.4.1	<i>Linear Scalar Example with Implementation</i>	91
6.4.2	<i>Nonlinear Scalar Example with Implementation</i>	97
6.5	MULTIVARIABLE PARAMETER INTERVAL GMC	102
6.5.1	<i>Linear Multivariable Case</i>	105
6.5.2	<i>Linear Multivariable Simulation</i>	106
6.5.3	<i>Nonlinear Multivariable Simulation</i>	119
6.6	DISCUSSION	124
	CHAPTER 7 CONCLUSION	125
7.1	FUTURE WORK	126
	APPENDICES	127
	LINEAR EXAMPLE	127
	LINEAR MULTIVARIABLE EXAMPLE	131
	NONLINEAR MULTIVARIABLE EXAMPLE	139
	REFERENCES	146
	VITA	155

LIST OF FIGURES

Figure 3-1 GMC Profile Specification	26
Figure 3-2 GMC Block Diagram for Linear Systems	27
Figure 3-3 Output and Control.....	30
Figure 4-1 3D Optimization.....	50
Figure 4-2 Top-View of Optimization.....	51
Figure 4-3 Pole & Zeros of GMC Transfer Function from Yellow Optimization Line ..	51
Figure 4-4 First Order Approximation	56
Figure 4-5 Prediction Level Improving Control Oscillations	58
Figure 4-6 Closed Loop of GMC and PGM C	61
Figure 4-7 Linear System. 2: Closed Loop Output and Control for GMC and PGM C...	62
Figure 4-8 Nonlinear System: Closed Loop Output and Control for Interval and PGM C	64
Figure 4-9 PGM C Control and Output for MIMO System.....	65
Figure 4-10 PGM C Control and Output for MIMO Uncertain System	65
Figure 4-11 PGM C Control and Output for MIMO Uncertain System	66
Figure 5-1 Depiction of Parameter Interval Factor	70
Figure 5-2 Description of Keyhole-Plasma Welding Process.....	72
Figure 5-3 Experiment 1: Closed Loop Results	73
Figure 5-4 Experiment 2: Closed Loop Results	74
Figure 5-5 Weld Image of Keyhole Plasma Process	75
Figure 5-6 Weld Image 2 of Keyhole Plasma Process	76
Figure 6-1 Open Loop Step Responses for Y_{min} , Y_{max} , Y_{mean}	92

Figure 6-2	Control Signal of Linear Simulation	93
Figure 6-3	Output Signal of Linear Simulation	93
Figure 6-4	First Set of Linear Experiments	94
Figure 6-5	Soft Plasma Experiments 1-3	95
Figure 6-6	Second Set of Linear Experiments	96
Figure 6-7	Keyhole Plasma Process (Experiment 1).....	97
Figure 6-8	Keyhole Plasma Process (Experiment 2).....	98
Figure 6-9	Keyhole Plasma Process (Experiment 3).....	99
Figure 6-10	Keyhole Plasma Process with New Interval Control	101
Figure 6-11	Minimum Open Loop Step Response.....	107
Figure 6-12	Maximum Open Loop Step Response	107
Figure 6-13	Case 1: Change of F thru Time	108
Figure 6-14	Case 1: GMC Control	109
Figure 6-15	Case 1: GMC Output	109
Figure 6-16	Case 1: Interval PGMC Control	110
Figure 6-17	Case 1: Interval PGMC Output	110
Figure 6-18	Case 2: GMC Control	112
Figure 6-19	Case 2: GMC Output	112
Figure 6-20	Case 2: Interval PGMC Control	113
Figure 6-21	Case 2: Interval PGMC Output	113
Figure 6-22	Case 3: Change of Parameters thru Time	114
Figure 6-23	Case 3: GMC Control	115
Figure 6-24	Case 3: GMC Output	115

Figure 6-25 Case 3: Interval PGMC Control.....	116
Figure 6-26 Case 3: Interval PGMC Output.....	116
Figure 6-27 Case 4: Interval PGMC Control.....	118
Figure 6-28 Case 4: Interval PGMC Output.....	118
Figure 6-29 Open Loop Step Response.....	120
Figure 6-30 GMC Control of Nonlinear Multivariable System.....	121
Figure 6-31 GMC Output of Nonlinear Multivariable System.....	121
Figure 6-32 Interval Control of Nonlinear Multivariable System.....	122
Figure 6-33 Interval Output of Nonlinear Multivariable System.....	122
Figure 6-34 Interval Control with $Y_{\max} = 1000$	123
Figure 6-35 Interval Output with $Y_{\max} = 1000$	123

LIST OF FILES

Istre.pdf..... 2.96 MB

CHAPTER 1

INTRODUCTION

1.1 Objectives

The objectives of this research will be to develop a multi-input multi-output (MIMO), nonlinear control that can handle parameter uncertainties, and then implement the algorithm by controlling an arc welding manufacturing process. Generic model control (GMC) has been selected to improve because the control law has the ability to incorporate a nonlinear model, the controller design parameters are intuitive and have a definable affect on the closed loop output, and the controller is already suitable for MIMO systems. A disadvantage with GMC is that the control signal for certain systems can frequently oscillate, which may make it impossible to implement in real control hardware. It is this aspect of GMC that will be improved first. Then improvements will be made on the ability of GMC to handle process uncertainties. The application of the developed controller will be in controlling arc-welding processes.

Therefore, the GMC controller will be studied and modified to develop a MIMO, nonlinear, robust controller. The initial, primary modification will be to reduce the control signal variation associated with a GMC controller by averaging control predictions. GMC control is already MIMO, and therefore extension to a new type of MIMO controller should be straightforward. The robustness of GMC has been studied and will be altered to handle parameter interval systems by changing the control law to

incorporate parameter intervals. After modifying GMC, the new controller design will be used to control an arc welding process.

1.2 Proposed Nonlinear Control Applications

The proposed research applications involve arc welding processes. Many arc welding processes have been shown to be nonlinear. Furthermore, the benefits of controlled arc welding in comparison with the cost of equipment have also been established, making it a worthwhile research project. These benefits include the guaranteed full penetration or fusion, higher energy density and efficiency, deep penetration, improved mechanical properties of the weld, improved arc stability, higher welding speed, and fewer requirements for joint preparation. Thus, by improving the control of the nonlinear processes involved in arc welding by incorporating a newly developed nonlinear control algorithm, the project not only becomes suitable for doctoral research but also has the potential to make a rewarding and marketable product.

1.3 Outline of Dissertation

Chapter two reviews the previous and current methods of nonlinear modeling and control. Chapter three reviews in depth the theoretical background of GMC and associated recent development as well as implementation issues encountered in research. Chapter four begins the control contribution of the author by analyzing control smoothing in GMC by the use of control predictions. Then the contribution is furthered in Chapter five and six by first studying the use of parameter interval systems within the GMC

control technique and then developing the final version of the proposed control algorithm and extending its use to multivariable systems.

CHAPTER 2

REVIEW OF NONLINEAR MODELING & CONTROL

There is now a greater need for high performance control systems in industry. This is due to economic pressures requiring increased throughput, a rise in the demand for higher quality products that are produced more consistently and a demand for greater material and energy utilization. Moreover, environmental and safety issues have increased the performance requirements of modern control systems. Studies have shown that nonlinear high performance controllers are not only possible to implement but also affordable, [1].

The mathematical reason that the modelling and control of nonlinear, high performance systems exists is the inability of linear systems to sufficiently characterize all of the various processes. Nonlinear systems are predominant in the processes and systems in the manufacturing setting. Moreover, even when the system behaves linearly at a certain time, the system dynamics are continuously changing so that there are uncertainties in the linear mathematical model developed to characterize the particular manufacturing process to be controlled. Often the processes are assumed to be linear in order to simplify the analysis and design of the controller, and this is frequently sufficient. However, to capture the behavior in the large or the local subtleties of the nonlinear system dynamics and control them with greater accuracy and/or some other performance criteria, a controller designed to incorporate a nonlinear model is required. However, some digital controllers for nonlinear systems, in the past, have often had the

disadvantage that they were difficult to implement in real systems because of the number of calculations performed to determine the optimum process input. Therefore, a MIMO, robust controller for nonlinear systems that can be easily designed by engineers, can be used for real uncertain systems, and can be implemented on-line is not only an interesting topic of research but also could have an extensive application in the manufacturing setting.

A review of nonlinear modelling and control is necessary in evaluating the performance of any developed nonlinear control technique. There are several nonlinear modelling and control techniques that have been created. In comparing the different methods, the primary considerations are model accuracy, complexity, method of system identification, and the ability to be implemented on-line.

2.1 Nonlinear Modelling

Nonlinear modeling is an interesting research topic. There are several different types of modeling with varying complexity. This section briefly overviews the most extensively used techniques, beginning with the most general of the dynamic models through the more specific and ending with other modeling types such as neural network, fuzzy, and others.

The nonlinear autoregressive moving average with exogenous input (NARMAX), [2], model is the nonlinear version of its linear counterpart ARMAX and is generally described by function f as in Equation 2-1.

$$y(k) = f(y[k-1], \dots, y[k-n_y], u[k-d], \dots, u[k-d-n_u]) + e(k)$$

Equation 2-1

For ARMAX models, prior to parameterization, the dominant system time-constants are used to select the sampling time, and then the model order can be selected. The moving average part of the model is selected based upon the assumed structure of the disturbance.

For NARMAX models, system identification is more complex unless the structure of f is known. Because the function, f , is usually unknown, the system must be estimated as \hat{f} . However, because there are numerous possibilities of the number of model terms and combination of terms, it is best to use a model that is a set of general basis functions. These basis functions can simply be a combination of ARMAX models or for greater accuracy a NARMAX model with a model class that provides a suitable set of regressors to fit the data generally over the domain of interest. Generally, the number of estimated parameters required to accurately characterize the nonlinear dynamics can be exceedingly large. However, recent studies have shown that the use of wavelets as the basis function have resulted in high accuracy with limited parameters, [3].

A subclass of NARMAX models is the Volterra series. The Volterra series [4] describes a large class of nonlinear systems and can be characterized by Equation 2-2 or in discrete time by Equation 2-3.

$$y(t) = h_0 + \sum_{n=1}^{\infty} \int_{R^n} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t - \tau_i) d\tau_1, \dots, d\tau_n$$

Equation 2-2

$$y(k) = h_0 + \sum_{n=1}^N \left[\sum_{k_1=0}^{K_n} \cdots \sum_{k_n=0}^{K_n} h_n(k_1, \dots, k_n) \prod_{i=1}^n x(k - k_i) \right]$$

Equation 2-3

The disadvantage with a Volterra series is also that there are a large number of parameters required in describing a system. Constraining the architecture of the series to reduce its complexity simplifies the model but also reduces its generality. Moreover because of the complexity of the Volterra series model, the control may not be online calculable and if the structure is not known, the parameter estimation is not straightforward [5,6].

The Wiener and Hammerstein models are special cases of a Volterra series [7,8]. The discrete Hammerstein model is described as a memoryless, nonlinear system as is given by Equation 2-4 followed by a linear system described by Equation 2-5.

$$u_{nonlinear} = \sum_{i=1}^m a_i u_{input}^i$$

Equation 2-4

$$G(z) = \frac{\beta_1 z^{-1} + \beta_2 z^{-2} + \dots + \beta_n z^{-n}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} - \dots - \alpha_n z^{-n}}$$

Equation 2-5

The Wiener model is described as the reverse of the Hammerstein model such that it is a linear system followed by memoryless, nonlinear system. Also, the Hammerstein and Wiener models can be cascaded to combine either a Hammerstein-Wiener model or Wiener-Hammerstein model. There are a number of ways of model identification for each of these types [9-11].

Other nonlinear models that have had extensive use are neural networks, which belong to a class of statistical models, and fuzzy systems, [12-19]. Neural networks are universal approximators and offer reduced ideal approximation error for only a linear increase in the number of parameters. Fuzzy systems have the distinct advantage of

incorporating heuristics in the design of the model's nonlinearities. Moreover, for each of these models the parameter identification is well established, and they provide the capability of on-line implementation. However for these models, the particular mechanistic reasons for the nonlinear system's dynamics are not as apparent as they are with some of the previous models.

2.2 Nonlinear Control

2.2.1 Overview

Although the computational demands for nonlinear control implementation are significantly greater than for a linear control system, advances in control system software and hardware have improved the applicability of nonlinear control. There is a wide range of techniques used to control nonlinear processes.

Previously the common approach in controlling nonlinear processes was to design the controller with a series of linear models that were valid for certain operating conditions. It was a type of gain scheduling approach whereby a certain linear controller was selected based upon the specific operating point that the process was running in at that time. However, recent interest in the control of nonlinear processes using nonlinear models has stimulated greater theoretical and practical developments.

Some issues that are encountered in nonlinear control are that of state estimation and disturbance modelling. In linear systems, the state estimation has, in general, been solved so that certainty equivalence between estimation and reality can be assumed, [20]. However, for nonlinear systems state estimation is not as straightforward and requires enhancements or learning mechanisms that can predict the states from the control and

output signal. Moreover unlike linear systems, process stability can not even be guaranteed by ensuring that the state estimators and state feedback control laws are stable. Then the modelling of disturbances is more complex, in that the nature and point of injection of the disturbance must be considered because the superposition principle is not valid in nonlinear systems.

The typical type of nonlinearities that occur in practice can be categorized into smooth and non-smooth groups, [20]. Smooth nonlinearities occur frequently and include products, exponentials or power type functions, and nonlinear continuous functions such as trigonometric functions. Non-smooth functions are also common in practice and these include things like specific mechanical properties such as hardstops or gear backlash and also more general process related things such as process limits or constraints. An important difference between non-smooth nonlinearities and smooth nonlinearities is that nonsmooth nonlinearities do not have definable inverse functions. This difference causes a distinct division in the way that these two process groups can be controlled. For example, there is no inverse function for a process hard stop that can be used to manipulate the control to cancel the effect of the hard stop. The smooth nonlinear functions do have inverse functions, and thus by the use of this inverse function, the nonlinearity can, in effect, be cancelled. However, because the non-smooth nonlinearities cannot be cancelled, the control action must be manipulated by addressing them in another way in the control calculation. Three primary ways have been proposed of manipulating the control action to address these non-smooth nonlinearities. First, it may be possible to lower the process performance requirements so that the non-smooth nonlinearities are completely avoided. Second, the controller could be enhanced with

types of process dependent embellishments that manage the control action based upon the known nonlinearities. This is commonly done in controllers called anti-windup controllers. Lastly, the controller type itself could be designed with the capability of including the nonlinearities within the control law.

Anti-windup controllers are commonly used for situations involving actuator constraints or other similar limitations, [21]. The anti-windup schemes involve mechanisms for notifying the controller when it is operating within a region that has certain constraints or limitations, and then the controller makes the predetermined modification to the control action.

In the following sections specific types of nonlinear control will be reviewed starting with feedback linearization, which addresses smooth nonlinearities by canceling them via a definable inverse. Then Model Predictive control will be reviewed, which can handle nonsmooth nonlinearities and uses non-linear programming to calculate the control action. Also, mentioned will be adaptive backstepping, which has recently been proposed. Lastly, H infinity nonlinear control and others will be discussed.

2.2.2 Feedback Linearization [20]

Feedback Linearization is a conceptually simple technique for the control of nonlinear systems with smooth nonlinearities and stable process inversion. However, if the process has unstable zero dynamics (i.e. nonminimum phase), Feedback Linearization fails to ensure stability. Thus, when using Feedback Linearization there are two ways of dealing with processes with unstable zero dynamics. First, it is possible to extend Feedback Linearization to nonminimum phase systems by selecting another output of the

system with respect to which the system has minimum phase characteristics. Second, it is might be possible to just construct a minimum phase approximation of the original model using an inner-outer factorization.

A brief review of Feedback Linearization is presented below. Consider the single-input single-output nonlinear state space system

$$\begin{aligned}\dot{x}(t) &= f(x) + g(x)u(t) \\ y(t) &= h(x)\end{aligned}$$

Equation 2-6

Assume that $x = 0$ is an equilibrium point of Equation 2-18 i.e. $f(0) = 0$, and that the nonlinear system has relative degree r defined in a certain neighborhood U of $x = 0$. Next, consider a stable linear differential operator $p(\rho)$ of degree r

$$p(\rho) = p_r \rho^r + p_{r-1} \rho^{r-1} + \dots + 1$$

Equation 2-7

Then $p(\rho)$, applied to the system output $y(t)$ can be written as

$$p(\rho)y(t) = b(x) + a(x)u(t)$$

Equation 2-8

where $b(x)$ and $a(x)$ are suitable nonlinear functions of the system states. Also, it is given that $a(x) \neq 0 \forall x \in U$, since the nonlinear system has relative degree r in U . From Equation 2-20 it is clear that applying the control law,

$$u(t) = \frac{y^*(t) - b(x)}{a(x)}$$

Equation 2-9

then the original nonlinear system can be transformed into a linear system of the form

$$p(\rho)y(t) = y^*(t)$$

Equation 2-10

where $y^*(t)$ can be any external signal.

The control law defined in Equation 2-9 is known as Input-Output Feedback Linearization. One advantage of Feedback Linearization is that it is simple and it allows a straightforward design of the differential operator $p(\rho)$, since the roots of $p(\rho)$ determine the dynamic behavior of the output of the closed-loop system.

Another control method related to Feedback Linearization that retains its desirable simplicity but without some of its restrictions is Generalized Feedback Linearization. However, these improvements have the undesirable tradeoff that the closed-loop properties lose their linear behavior.

Previously, the order of the linear differential operator $p(\rho)$ was r i.e., the relative degree of the nonlinear system.

Allowing the degree of $p(\rho)$ to be $n_p \geq r$ and if the input is made to satisfy a linear dynamic model of the form

$$l(\rho)u(t) = u^*(t)$$

Equation 2-11

where $l(\rho)$ is a differential operator of degree $n_l = n_p - r$

$$l(\rho) = l_{n_l} \rho^{n_l} + l_{n_l-1} \rho^{n_l-1} + \dots + 1$$

Equation 2-12

And $u^*(t)$ is the steady state input signal which makes the steady state value of the output $y(t)$ to be equal to $y^*(t)$

Combining of Equations 2-10 and 2-11, yielding, with $\lambda \in [0,1]$.

$$(1 - \lambda)[p(\rho)y(t) = y^*(t)] + \lambda[l(\rho)u(t) = u^*(t)] = 0$$

Equation 2-13

constitutes the Genralized Feedback Linearization (GFL) control law, which can also be written as

$$z(t) = p'(\rho)y(t) + l(\rho)u(t) = \bar{z}(t)$$

Equation 2-14

where

$$p'(\rho) = \frac{1 - \lambda}{\lambda} p(\rho)$$

$$\bar{z}(t) = \frac{1 - \lambda}{\lambda} y^*(t) + u^*(t)$$

Equation 2-15

Notice that Equation 2-15 implicitly defines an improper linear control law which becomes a nonlinear proper control law when the state space model is used to evaluate $p'(\rho)y(t)$. This strategy reverts to the usual Feedback Linearization strategy; by taking $\lambda=0$ in Equation 2-13. The strategy can handle all stable systems, whether or not they are stably invertible, by taking $\lambda=1$. By continuity, various combinations of stable and stably invertible dynamics will also be able to be stabilized by this class of control law, depending upon the design of the differential operators $p(\rho)$ and $l(\rho)$, as well as the value of the parameter λ .

To develop the control law implicitly defined in Equation 2-26 we introduce a dummy variable $\bar{u}(t)$ is introduced as follows:

$$\bar{u}(t) = l(\rho)u(t)$$

Equation 2-16

Substituting Equations 2-15 and 2-16 into expression 2-14, which defines the GFL control strategy, the following nonlinear control law is finally obtained:

$$\bar{u}(t) = \frac{\bar{z}(t) - b(\xi)}{1 + a(\xi)}$$

Equation 2-17

In generalized feedback linearization the linear form is used to define a variable z . The key point in GFL is that p and l' are designed such that the zero dynamics associated with z are locally stable.

2.2.3 Model Predictive Control [22-24]

A well-known class of nonlinear controllers that directly uses the nonlinear model is model predictive controllers (MPC), [22]. Linear MPC is a discrete time controller that calculates the present control, at each sampling time, by predicting over a horizon p the process response to changes in control. The change in control that is within specified constraints and that gave the most desirable process response is then implemented.

Nonlinear MPC (NMPC) is similar and is constructed as solving on-line finite horizon open-loop optimal control problem at each sampling time using the system model and predetermined state, input, and output constraints. Based on measurement samples obtained at time t , the controller predicts the future dynamic behavior of the system over a prediction horizon T_p and determines (over a control horizon $T_c \leq T_p$) the input such that a predetermined open-loop performance objective function is optimized.

Because there are disturbances and the model of the process is likely not identical to the process, the process behavior will be different from the predicted behavior. Thus, only the first step of the system inputs calculated by the controller will be implemented, and after the measurement at the next sampling time, the controller's online prediction/optimization calculation will be completed again. However, if the model of the process was completely accurate and if the optimization problem could be solved for the duration of the experiment, then one could apply the controller's predicted inputs open loop for the duration of the experiment. However, this is not generally possible.

Consider the stabilization problem for a class of systems described by the following nonlinear set of differential equations.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0$$

Equation 2-18

Subject to input and state constraints of the form:

$$\mathbf{u}(t) \in U, \forall t \geq 0 \quad \mathbf{x}(t) \in X, \forall t \geq 0,$$

Equation 2-19

$$U := \{\mathbf{u} \in \mathcal{R}^m \mid \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}\},$$

$$X := \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}\}$$

Equation 2-20

Assumption 1: $U \subset \mathfrak{R}^p$ is compact, $X \subseteq \mathfrak{R}^n$ is connected and $(0,0) \in X \times U$.

Assumption 2: The vector field $\mathbf{f}: \mathfrak{R} \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ is continuous and satisfies $\mathbf{f}(\mathbf{0}, \mathbf{0}) = 0$. In addition, it is locally Lipschitz continuous in \mathbf{x} .

Assumption 3: The system (Equation 2-18) has an unique continuous solution for any initial condition in the region of interest and any piecewise continuous and right continuous input function $\mathbf{u}(\cdot): [0, T_p] \rightarrow U$.

Usually, the finite horizon open-loop optimal control problem described above is mathematically formulated as follows: (internal controller variables are denoted by a bar)

Find

$$\min_{\bar{\mathbf{u}}(\cdot)} J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c T_p)$$

Equation 2-21

with

$$J(\mathbf{x}(t), \bar{\mathbf{u}}(\cdot); T_c T_p) := \int_t^{t+T_p} F(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)) d\tau$$

Equation 2-22

subject to:

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(\tau) &= \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)), \quad \bar{\mathbf{x}}(t) = \mathbf{x}(t) \\ \bar{\mathbf{u}}(\tau) &\in U, \quad \forall \tau \in [t, t+T_c] \\ \bar{\mathbf{u}}(\tau) &= \bar{\mathbf{u}}(t+T_c), \quad \forall \tau \in [t+T_c, t+T_p] \\ \bar{\mathbf{x}}(\tau) &\in X, \quad \forall \tau \in [t, t+T_p] \end{aligned}$$

Equation 2-23

where T_p and T_c are the prediction and control horizon with $T_c \leq T_p$ and internal controller variables are denoted by a bar.

The function F in Equation 2-24, called stage cost, specifies the desired control performance that can arise, for example, from economical and ecological considerations. The standard quadratic form is the simplest and most often used one:

$$F(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_s)^T Q(\mathbf{x} - \mathbf{x}_s) + (\mathbf{u} - \mathbf{u}_s)^T R(\mathbf{u} - \mathbf{u}_s),$$

Equation 2-24

Where \mathbf{x}_s and \mathbf{u}_s denote given setpoints: Q and R denote positive definite, symmetric weighting matrices.

The *closed-loop* control is defined by the optimal solution of Equation 2-20 at the sampling instants:

$$\mathbf{u}^*(\tau) := \bar{\mathbf{u}}^*(\tau; x(t), T_p, T_c), \tau \in [t, \delta]$$

Equation 2-25

The optimal value of the NMPC open-loop optimal control problem as a function of the state will be denoted in the following as value function:

$$V(\mathbf{x}; T_p, T_c) = J(\mathbf{x}, \bar{\mathbf{u}}^*(\cdot; \mathbf{x}(t)); T_p, T_c).$$

Equation 2-26

The value function plays an important role in the proof of the stability of various NMPC schemes, as it serves as a Lyapunov function candidate.

The disadvantages of nonlinear MPC are primarily due to the finite horizon optimal control problem being non-convex. Non-convexity introduces the questions of how long will the optimization take, whether it will terminate, and is a suboptimal solution acceptable. The finite horizon optimal control problem associated with nonlinear MPC is not guaranteed to be convex and it is difficult to obtain the global optimal solution. Therefore, because of the non-convexity NMPC formulations need to be derived that guarantee solution feasibility, robustness, and performance despite the solution being sub-optimal. Moreover, for further development of NMPC algorithms, faster optimization solvers need to exploit the inherent structure of the process. For it is possible that in

solving the finite horizon optimal control problem one can exploit the specific system dynamics, e.g. Lipschitz continuous, static nonlinearity, input-affine, bilinear, hybrid, piecewise affine, non-holonomic or homogeneous.

2.2.4 Backstepping [25]

Backstepping is a method that can be used on nonlinear systems of special structure to find an output having a passivity property, including the relative degree one and stable zero dynamics. Finding an “output” z having stable zero dynamics plays a key role in this design. A recently developed control technique called adaptive backstepping, [25], does not require stable zero dynamics; however, it does require that the unknown parameters of the system are constant.

Consider the class of pure-feedback systems with unknown parameters is well represented by the third order system

$$\begin{aligned}\dot{x}_1 &= x_2 + \phi_1^T(x_1, x_2)\theta \\ \dot{x}_2 &= x_3 + \phi_2^T(x_1, x_2, x_3)\theta \\ \dot{x}_3 &= u + \phi_3^T(x_1, x_2, x_3)\theta\end{aligned}$$

Equation 2-27

where the $p \times 1$ vector θ is constant and unknown.

When the parameter vector θ is known, the pure-feedback system given in Equation 2-27 under backstepping control can be formed to essentially mimic feedback linearization. However, feedback linearization linearizes the output and thus may cancel useful nonlinearities. Adaptive backstepping designs are more flexible and do not force the designed system to appear linear. In fact, they can not only avoid the cancellations of

useful nonlinearities but can often introduce additional nonlinear terms to improve transient performance.

“The idea of adaptive backstepping is to design a controller for Equation 2-27 recursively by considering some of the state variables as “virtual controls” and designing for them intermediate control laws. In Equation 2-27 the first virtual control is x_2 . It is used to stabilize the first equation as a separate system. Since θ is unknown, this task is solved with an adaptive controller consisting of the control law $\alpha_1(x_1)$ and the update law $\hat{\theta} = \tau(x_1)$, as in the Lyapunov-based design.”

“Adaptive backstepping treats the parameter θ in the second equation of Equation 2-27 as a new parameter and assigns to it a new estimate with a new update law. As a result, there are several estimates for the same parameter. This overparameterization is avoided by considering that in the first step $\hat{\theta} = \tau(x_1)$ is not an update law but only a tuning function $\tau(x_1)$. This “tuning function” is used in subsequent recursive steps and the discrepancy $\hat{\theta} - \tau(x_1)$ is compensated with additional terms in the controller. Both adaptive backstepping and tuning functions achieve the goals of stabilization and tracking.”

“The tuning functions’ approach is an advanced form of adaptive backstepping. It has the advantage that the dynamic order of the adaptive controller is minimal. The dimension of the set to which the states and parameter estimates converge is also minimal,” [25].

“Certain drawbacks of tuning functions are that they do not offer freedom of choice of parameter update laws and for systems with many unknown parameters the dynamic order of its overparameterized controller is high. The order of the tuning

functions controller is minimal, but for high-order systems its nonlinear expressions become increasingly complex due to the built-in interaction between the identifier and the control law.”

“There is a new controller that was developed with strong parametric robustness properties: It achieves boundedness without adaptation. Furthermore, the new controller, called the ISS-controller, guarantees boundedness not only in the presence of constant parameter errors, but also in the presence of time-varying parameter estimates. These input-to-state stability (ISS) properties make the ISS-controller suitable for modular adaptive nonlinear designs,” [25].

“Backstepping design of output-feedback controllers is performed on systems enlarged by filters, and the filter states are used for feedback. The output-feedback modular designs result in separation of three design modules: the control law, the identifier, and the state estimator.”

Transient performance bounds of the closed loop systems designed using adaptive backstepping techniques can be computed and therefore the controller can be designed to meet certain transient requirements.

Adaptive backstepping and its use in nonlinear control theory is new and is still focused on the development of the basic schemes. Its robust analysis is yet to be developed. [25]

2.2.5 H_∞ Nonlinear Control [26]

An approach that originates back to the beginning of the eighties is called H_∞ optimization. This method can be viewed as a worst-case design methodology in the

frequency domain. In fact H_∞ stands for the space of complex functions which are bounded and analytic in the closed right half of the complex plane, [26].

“The H_∞ norm from the exogenous disturbance inputs to the to-be-controlled variables in the frequency domain, used to describe the control objectives, is equal to the L_2 -induced norm for the time-domain versions, under the constraint of internal stability. The property of finite L_2 -induced norm of a stable system, also called finite L_2 -gain, can be characterized as the dissipativity of the system with respect to a certain supply rate.”

“The suboptimal H_∞ problem can also be formulated as a two player, zero sum linear quadratic differential game, where the disturbances are considered as the maximizing player whose goal it is to maximize a certain cost criterium, while the controls denote the minimizing player whose goal it is to minimize the same cost criterion,” [26].

These H_∞ solutions for linear systems have been extended for nonlinear systems by using Hamilton-Jacobi equations which extend the Riccati equations used in the linear theory. “The solution of the nonlinear state feedback H_∞ problem was described using a Hamilton-Jacobi inequality. The nonlinear measurement feedback H_∞ problem is up to now not completely understood. Nevertheless, sufficient conditions for the existence of controllers of a specific form solving the regular measurement feedback H_∞ problem have been derived. Also, necessary conditions for solvability of the problem have been given,” [26].

“Most of the results are concerned with the regular H_∞ problem for non-linear systems that are affine in the inputs and the disturbances. Recently an extension of some

of these results to general nonlinear systems have been made. The regularity of the H_∞ problem is concerned with certain rank conditions on the feed through matrices. The singular H_∞ problem occurs when these regularity assumptions are violated. Singular H_∞ problems naturally arise when considering certain robustness problems such as parameter uncertainty and multiplicative uncertainty.” [26]

2.2.6 Other Nonlinear Controllers

Neural networks and fuzzy controllers are suitably designed to control nonlinear systems [27-33]. Neural networks are parameterized nonlinear functions. Their parameters are, for instance, the weights and biases of the network. Adjustment of these parameters results in different shaped nonlinearities. The neural network control is trained by adjusting the neural network parameters as a function of the error between the network output and a series of training data. Fuzzy controllers, which have the unique advantage of including the designer’s heuristics, can sometimes lead to better convergence properties for the actual input-output map. Fuzzy controllers are simply nonlinear functions that are parameterized by, for example, the membership function and consequence parameters. For both neural network and fuzzy controller models, adaptive schemes have been studied and developed.

Other nonlinear control examples include combined linear or constrained nonlinear controllers. Combinations of polynomial linear ARMA models have been used to develop a predictive control strategy for nonlinear systems [34]. Sliding-mode techniques operated by switching between two nonlinear feedback control rules [35]. A nonlinear controller based on normal form theory was designed to drive simple nonlinear

systems with polynomial vector fields derived around singularities [36]. The min-max optimization controllers were developed to find the optimal control for discrete, nonlinear processes [37]. The procedure found the best control for the worst-case plant, based on parameter bounds. Another unique nonlinear controller based on game theories was developed as a robust control to achieve the best bound on the worst close-loop performance [38]. Much work has also been accomplished on many MIMO nonlinear systems, especially a class of nonlinear systems called bilinear systems [39-41].

Nonlinear adaptive controllers have seen extensive use to reduce the effects of plant parameter variations by adjusting the controller online [42]. In nonlinear control applications, the compensation of deadtimes is important and has been studied [43]. In addition, the handling of the control and output constraints is also relevant to nonlinear control implementation and has been studied [44].

In 1988, Lee and Sullivan [45], presented a nonlinear control structure that they called generic model control (GMC) that permits the direct use of the nonlinear multivariable process model. There is not only the usual requirement that each of the controlled variables reach a defined setpoint but also the rate of approach to the setpoint is specified. Similar differential geometric approaches were also developed, [46,47] and are each based upon an earlier work, [48]. However, they differ in the performance definition.

CHAPTER 3

GENERIC MODEL CONTROL

The essential idea of Generic Model Control (GMC) is to find the values of the manipulated variables that force a model of the system to follow a desired reference trajectory. It is related to the subset of mathematical knowledge known as differential geometry that involves linearization of nonlinear mappings between the input and output variables.

3.1 A Review of Generic Model Control

Consider the process model that can be described by a set of differential equations as:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{d}, t, \theta)$$

Equation 3-1

where \mathbf{y} is a vector of process outputs of dimension m , \mathbf{u} is a vector of process inputs of dimension m , \mathbf{d} is a vector of process disturbances of dimension l , t is time and θ is a vector of model parameters of dimension q . In general \mathbf{f} is a vector of nonlinear known functional relationships. In determining the control at a particular step, the GMC algorithm specifies a rate of change of the output variables as:

$$\dot{\mathbf{y}}_{Desired} = \mathbf{K}_1(\mathbf{y}_{ref} - \mathbf{y}) + \mathbf{K}_2 \int_0^t (\mathbf{y}_{ref} - \mathbf{y}) dt$$

Equation 3-2

In Equation 3-2, two process desires are expressed. First, when the system is at a greater distance from the setpoint, then the system should travel towards the setpoint more quickly. Moreover, the longer that the system has remained offset from the setpoint, then the system should also travel towards the setpoint more quickly. The values of \mathbf{K}_1 and \mathbf{K}_2 are what determine the speeds. Lastly, to solve for the control, the actual output rate is set equal to the desired output rate, $\dot{\mathbf{y}}_{Desired} = \dot{\mathbf{y}}$, in other words setting Equation 3-1 equal to Equation 3-2, giving the following equation from which the control, \mathbf{u} , can be solved:

$$\mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{d}, t, \theta) = \mathbf{K}_1(\mathbf{y}_{ref} - \mathbf{y}) + \mathbf{K}_2 \int_0^t (\mathbf{y}_{ref} - \mathbf{y}) dt$$

Equation 3-3

Therefore, GMC is a type of process control for linear or nonlinear systems that uses proportional and integral error terms, similar to PI control, to adjust the control input to achieve a desired closed loop output trajectory. For GMC, the specified closed loop output trajectory is directly determined using Equation 3-2 and then the control to achieve that trajectory is indirectly solved for by using an inverse of the process model (linear or nonlinear), [45].

The system in Equation 3-2 is the key element in the GMC control law. The parameters \mathbf{K}_1 and \mathbf{K}_2 are diagonal matrices with elements chosen independently for each controlled output. These choices should be made reasonably and with an understanding of the natural system's response characteristics. For the simple single-input single-output system, assuming $\dot{\mathbf{y}}_{Desired} = \dot{\mathbf{y}}$, the Laplace transformation of the closed loop output in Equation 3-2 gives:

$$\frac{y}{y_{ref}} = \frac{2\tau\xi s + 1}{\tau^2 s^2 + 2\tau\xi s + 1}$$

Equation 3-4

where

$$\tau = \frac{1}{\sqrt{k_2}}, \xi = \frac{k_1}{2\sqrt{k_2}}$$

Equation 3-5

The transfer function given in Equation 3-4 does not behave identically to a classic second-order system due to presence of the zero in the transfer function. However, similar plots to the second-order system can be made with the plot of a normalized response of the system $\frac{y}{y_{ref}}$ vs. normalized time $\frac{t}{\tau}$ where ξ is a parameter as is shown in Figure 3-1.

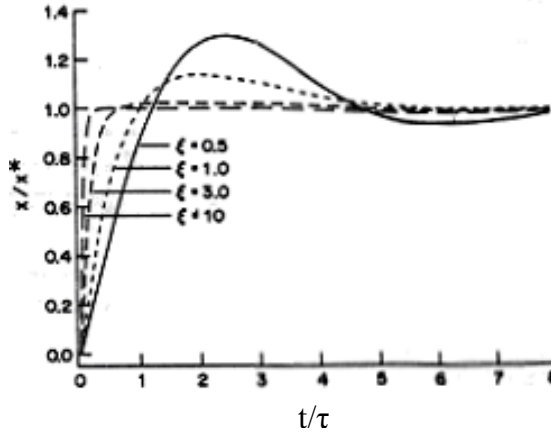


Figure 3-1 GMC Profile Specification

As ξ is increased the process has less overshoot and eventually after $\xi \geq 4$, the process due to pole-zero cancellation resembles a first order response. The settling time is predominantly determined by the value of τ , and increases directly with τ . Thus, the design of the reference can be accomplished by the first determining the shape of the desired response by selecting ξ and then specifying the timing of the response in

appropriate selection of τ in relation to the natural process's response time.

Alternatively, the designer can also choose the closed loop reference system by setting the poles of the transfer function given in Equation 3-4.

Insight can be gained into the nature of the GMC control technique by examining the GMC control of linear systems. Assuming the process can be modeled as:

$$Y = G \cdot U + d'$$

Equation 3-6

and if \hat{G} is the approximate model of the process, then the GMC control is given as:

$$U_{GMC} = (\hat{G}^{-1}) \cdot \left(\frac{1}{s^2 + sK_1 + K_2} \right) \cdot ((sK_1 + K_2) \cdot (y_{ref} - d) - sEd)$$

Equation 3-7

Furthermore, Equation 3-6 and 3-7 can be used to construct the closed loop block diagram for linear systems given in Figure 3-2.

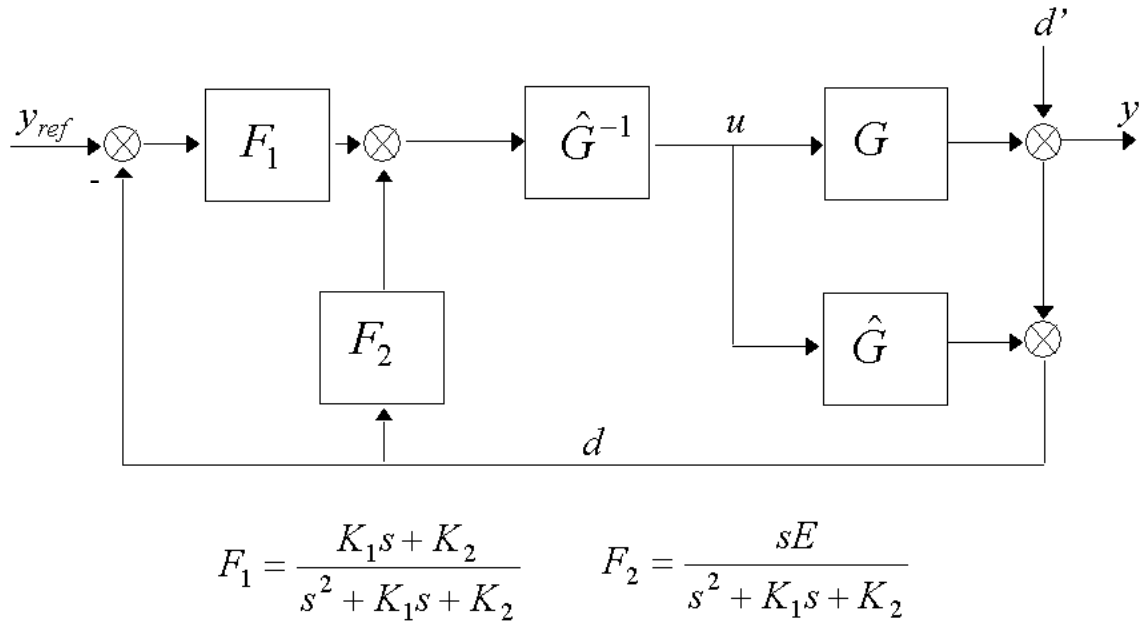


Figure 3-2 GMC Block Diagram for Linear Systems

The control parameter, E , is a diagonal matrix that is used in rejecting disturbances, and should be properly set according to the assumed nature of the disturbance.

Alternatively if the process can be modeled as a nonlinear system given as:

$$\dot{x} = f(x, u, d, t)$$

$$y = g(x)$$

Equation 3-8

where x, d, t , and y are the state, disturbance, time, and output, respectively. The GMC control input, u , can then be solved for by calculating:

$$\dot{y} = \frac{\partial g}{\partial x} f(x, u, d, t) = K_1(x_{ref} - x) + K_2 \int (x_{ref} - x) dt$$

Equation 3-9

to obtain u as a function of a number of variables as given below.

$$u = h(x, u, d, t, K_1(x_{ref} - x), K_2 \int (x_{ref} - x) dt)$$

Equation 3-10

Furthermore, if there are constraints on the input or process variables, then the control u should be chosen to minimize the instantaneous value of Equation 3-11 (in other words $\dot{y} - \dot{y}_{Desired}$) such that all constraints are satisfied. This minimization may be solved for algebraically but will likely be accomplished via nonlinear programming optimization routines.

$$J(x, u, d, t) = \frac{dg}{dx} f(x, u, d, t) - K_1(x_{ref} - x) - K_2 \int (x_{ref} - x) dt$$

Equation 3-11

Another GMC closed loop control example is given as follows, if the system can be described by the state space linear system:

$$f(x, u, d, t) = \dot{x} = Ax + Bu + Dd$$

$$y = x$$

Equation 3-12

then the GMC control without any variable constraints would be given by

$$u_{GMC} = B^{-1}K_1(y_{ref} - y) + B^{-1}K_2 \int (y_{ref} - y)dt - B^{-1}Dd - B^{-1}Ax$$

Equation 3-13

GMC can also be used for discrete systems. Using discrete approximations for the time derivative and time integration operators, the following digital GMC desired closed loop trajectory is given as:

$$\dot{y}_{Desired,k} = \frac{y_{k+1} - y_k}{T_s} = K_1 * (y_{ref,k} - y_k) + K_2 * \sum_{i=1}^k \frac{T_s}{2} * ((y_{ref,i} - y_i) + (y_{ref,i-1} - y_{i-1}))$$

Equation 3-14

The digital control can be solved for in the same way as the continuous control so that:

$$u_k = h_k \begin{pmatrix} x_k, \dots, x_1, d_k, \dots, d_1, t_k, \\ K_1(y_{ref,k} - y_k), \\ K_2 \sum_{i=1}^k T_s \cdot (y_{ref,i} - y_i) \end{pmatrix}$$

Equation 3-15

Despite the control precision achieved by including the model of the process in the derivation of the control, a disadvantage with the GMC method is that there is no limitation on control signal variation. The implementation of a maximum control variation, Δu_{max} , can be imposed upon the control; however, this constraint degrades the closed loop performance, [45]. An example of the control oscillations required by the

GMC method is shown in Figure 3-3, for the linear system given in Equation 3-16 and using $\xi=6$, $\tau=0.5$.

$$G(s) = \frac{1.5}{s^2 + 14s + 40.02} \quad \text{or}$$

$$G(z) = \frac{.00015}{1.144z^2 - 2.14z + 1} \quad \text{for } T_s = 0.01 \text{ sec.}$$

Equation 3-16

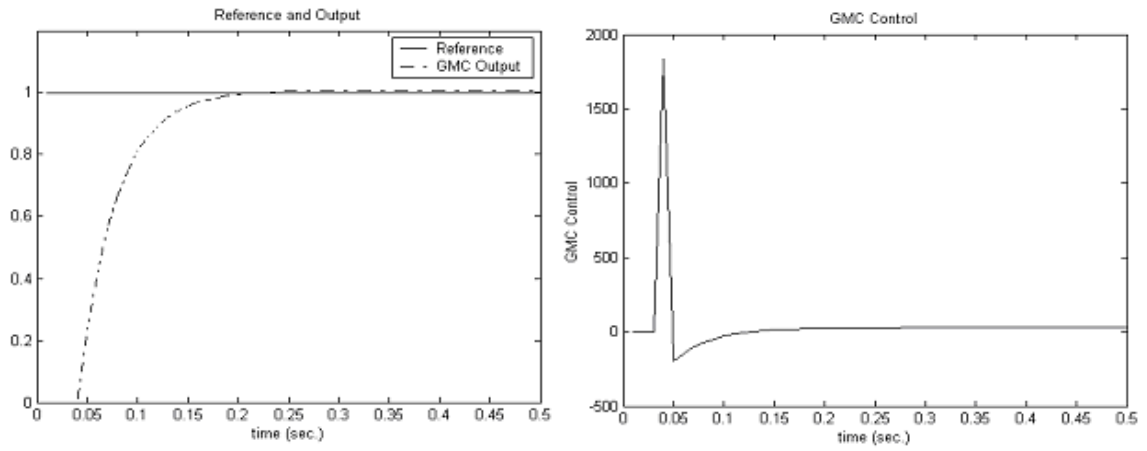


Figure 3-3 Output and Control

3.1.1 A Nonlinear Modeling Approach for GMC

The core of the GMC algorithm that has been just previously discussed has also been extended to cope with other important aspects of nonlinear control. These include the use of steady-state models and systems with dead time. For the GMC algorithm, the nonlinear dynamic model as given in Equation 3-1 is assumed to be available. However, steady-state models of the form given in Equation 3-17 are much more readily available.

$$\mathbf{f}_{ss}(\mathbf{y}, \mathbf{u}, \mathbf{d}, \theta) = 0$$

Equation 3-17

Thus, if a dynamic nonlinear model of the form in Equation 3-1 is not available, it has been shown that an appropriate approximation of it, for GMC control, is to use the nonlinear steady-state model of Equation 3-17 in tandem with a linear model that models the process transients. The approximate transient modelling is accomplished by forming a first or second order linear model by estimating the primary time constants of the process, which are available through step response tests. Although these estimates of the transient behavior are going to be inaccurate at different operating conditions, the total result of the linear transient model coupled with the nonlinear steady-state model are usually sufficiently accurate for control performance, [1].

An example of the transient first order linear modelling could be given as:

$$\dot{\mathbf{y}} \approx \mathbf{T}^{-1}(\mathbf{y}_u - \mathbf{y})$$

Equation 3-18

where \mathbf{T} is a diagonal matrix of the estimated open-loop time constants of a step change in all input variables and \mathbf{y}_u is the steady-state values of the outputs with no further control action. Therefore, if control designer has the nonlinear steady-state model as given in Equation 3-17, and if the transient behavior of the model can be approximated by Equation 3-18, then a control action can be determined through the use of the combination of the two. This is done by solving the nonlinear steady-state model for the control action that achieves the output \mathbf{y}_u given via the GMC derived requirement in Equation 3-19.

$$\mathbf{y}_u = \mathbf{y} + \mathbf{T}(\mathbf{K}_1(\mathbf{y}_{ref} - \mathbf{y}) + \mathbf{K}_2 \int (\mathbf{y}_{ref} - \mathbf{y}) dt)$$

Equation 3-19

3.2 A Numerical Method in Applying GMC

In application of the GMC control law, it is not always suitable to algebraically manipulate all the equations in obtaining an explicit expression for the control. The solution to this problem is to solve them numerically, and this will also allow for there to be process and input constraints, [1]. Thus, there are two things to consider. First, whether the GMC algorithm together with the process model can be solved for to obtain the control u , and secondly how to minimize the difference between the reference system and the process outputs when there are process constraints. Constrained optimization approaches have been formed to answer both of these questions for several nonlinear chemical processes, [49]. The control optimization uses slack variables that define a process variable's or constraint variable's departure from the chosen reference trajectory. The selection of the weighting factor for each of these slack variables provides customization to emphasize certain constraints or certain tracking of the output references.

The constraints for the optimization problem can be written using a nonlinear function, Ψ_i , as:

$$C_{Li} \leq C_i = \Psi_i(\mathbf{y}, \mathbf{u}, \mathbf{d}, t, \theta) \leq C_{Ui}; \quad i = 1 \dots q$$

Equation 3-20

Moreover, it is sensible that the trajectory of the violated constraints follow a reference using the GMC rule and slack variables λ such that:

$$\begin{aligned}\frac{d\mathbf{C}_i}{dt} - \lambda_{ci}^+ &\leq \mathbf{K}_{1Ci}(\mathbf{C}_{Ui} - \mathbf{C}_i); \quad i = 1 \dots q \\ \frac{d\mathbf{C}_i}{dt} + \lambda_{ci}^- &\leq \mathbf{K}_{2Ci}(\mathbf{C}_i - \mathbf{C}_{Li}); \quad i = 1 \dots q\end{aligned}$$

Equation 3-21

Similarly to the simplifying approach taken in Section 3.1.2 of modeling the nonlinear dynamic model by using a nonlinear steady-state model and a linear transient model, the solution to Equation 3-21 can either use the nonlinear dynamic models, \mathbf{C}_i , or can make use of the following approximate models:

$$\begin{aligned}\frac{d\mathbf{C}_i}{dt} &= \mathbf{T}_{Ci}^{-1}(\mathbf{C}_i^{\text{AIM}} - \mathbf{C}_i); \quad i = 1 \dots q \\ \mathbf{C}_i^{\text{AIM}} &= \psi_i(\mathbf{y}^{\text{AIM}}, \mathbf{u}, \mathbf{d}, t, \theta); \quad i = 1 \dots q \\ \mathbf{f}_{ss}(\mathbf{y}^{\text{AIM}}, \mathbf{u}, \mathbf{d}, t, \theta) &= 0\end{aligned}$$

Equation 3-22

Finally, the solution of the multivariable constrained control problem can be solved as a nonlinear optimization problem, which minimizes a function of the slack variables. The formulation of this optimization is as follows:

Choose:

$$\mathbf{u}, \lambda_{pi}^-, \lambda_{pi}^+, \lambda_{cj}^-, \lambda_{cj}^+; \quad i = 1 \dots q$$

Equation 3-23

To minimize:

$$J = \sum_{i=1}^m \omega_{pi}^- (\lambda_{pi}^-)^2 + \omega_{pi}^+ (\lambda_{pi}^+)^2 + \omega_{cj}^- (\lambda_{cj}^-)^2 + \omega_{cj}^+ (\lambda_{cj}^+)^2 + \omega_{\Delta u}^+ (\Delta u_i)^2 + \sum_{j=1}^q \omega_{cj}^- (\lambda_{cj}^-)^2 + \omega_{cj}^+ (\lambda_{cj}^+)^2$$

Equation 3-24

where ω_i and ω_j are weights such that $\omega_i \geq 0$ and $\omega_j \geq 0$.

Subject to:

$$\begin{aligned}
\frac{\partial \mathbf{g}_i}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}, t) + \lambda_{pi}^+ - \lambda_{pi}^- &= \mathbf{K}_{1i}(\mathbf{y}_{ref,i} - \mathbf{y}_i) + \mathbf{K}_{2i} \int_{t_0}^t (\mathbf{y}_{ref,i} - \mathbf{y}_i) dt; \quad i = 1 \dots m \\
\frac{d\mathbf{C}_j}{dt} - \lambda_{cj}^+ &\leq \mathbf{K}_{1Cj}(\mathbf{C}_{uj} - \mathbf{C}_j); \quad j = 1 \dots q \\
\frac{d\mathbf{C}_j}{dt} + \lambda_{cj}^- &\leq \mathbf{K}_{2Cj}(\mathbf{C}_j - \mathbf{C}_{Lj}); \quad j = 1 \dots q \\
\mathbf{C}_j &= \boldsymbol{\Psi}_j(\mathbf{y}, \mathbf{u}, \mathbf{d}, t, \theta); \quad j = 1 \dots q \\
\mathbf{u}_{Li} &\leq \mathbf{u}_i \leq \mathbf{u}_{Ui}; \quad i = 1 \dots m \\
\Delta \mathbf{u}_{Li} &\leq \mathbf{u}(t + \Delta t)_i - \mathbf{u}(t)_i \leq \Delta \mathbf{u}_{ui}; \quad i = 1 \dots m \\
\lambda_{pi}^+ &\geq 0, \lambda_{pi}^- \geq 0, \lambda_{cj}^+ \geq 0, \lambda_{cj}^- \geq 0; \quad i = 1 \dots m, j = 1 \dots q
\end{aligned}$$

Equation 3-25

The variables λ_p^+ , λ_p^- , λ_c^+ , λ_c^- are the upper and lower output performance slack variables and the upper and lower constraint slack variables, respectively. The gains \mathbf{K}_1 , \mathbf{K}_2 are the diagonal matrices that have the PI gains setting the closed loop output reference, which are selected using Figure 3-1. The soft process constraints are described in $\mathbf{C} - \mathbf{C}_u$ for the upper bound and $\mathbf{C} - \mathbf{C}_L$ for the lower bound. The gains \mathbf{K}_{1C} , \mathbf{K}_{2C} specify the speed the constrained process variables approach their bounds. Higher gains avoid constraint violation but sacrifice closed loop output performance. The values of the matrices are also selected using the method in Figure 3-1. Finally, there are also hard constraints as those given for the maximum and minimum control $\mathbf{u}_L \leq \mathbf{u} \leq \mathbf{u}_u$ and maximum and minimum change in control $\Delta \mathbf{u}_L \leq \mathbf{u}(t + \Delta t) - \mathbf{u}(t) \leq \Delta \mathbf{u}_u$. The nonlinear optimization that arises is well structured since a slack variable is added to each control law equation to ensure that a solution exists. The solution of the nonlinear programming (NLP) optimization is reasonably fast since the current control settings and slack variables provide a good estimate for the solution vector. For Ansari and Tade's

processes using NLP techniques such as sequential quadratic programming (SQP) usually converged in 3 to 4 iterations, [49].

They also found that for the dynamic nonlinear model a multivariable first order model of the type in Equation 3-26, provided sufficiently accurate control.

$$\begin{matrix} Y_1 \\ Y_2 \end{matrix} \begin{bmatrix} \begin{matrix} U_1 \\ k_{11} \frac{e^{-\theta_{11}s}}{\tau_{11}s + 1} \end{matrix} & \begin{matrix} U_2 \\ k_{12} \frac{e^{-\theta_{12}s}}{\tau_{12}s + 1} \end{matrix} \\ \begin{matrix} e^{-\theta_{21}s} \\ k_{21} \frac{e^{-\theta_{21}s}}{\tau_{21}s + 1} \end{matrix} & \begin{matrix} e^{-\theta_{22}s} \\ k_{22} \frac{e^{-\theta_{22}s}}{\tau_{22}s + 1} \end{matrix} \end{bmatrix}$$

Equation 3-26

3.3 GMC Robustness and Stability

3.3.1 GMC Robustness Properties

A different approach to the Generic Model Control (GMC) design for uncertain nonlinear processes is robust GMC. The approach is developed by integrating H_∞ -loop shaping controller within the GMC framework. This new design framework provides not only the ability to optimally tune the GMC parameters, but also to enjoy the guaranteed robust stability and performance properties in terms of the v -gap. [50]

The design methods are derived from the application of the v -gap metric and the H_∞ -loop shaping method within the GMC design framework. It should be noted that the robust GMC (RGMC) concept is not limited to a class of nonlinear processes with relative degree one. Simple and iterative approaches to robust Generic Model Control (RGMC) design have been developed extending Samyudia and Lee's method. A simple procedure to design a robust GMC controller has been derived by analyzing the bounds

on the loop shapes at different frequency ranges. To achieve an improved closed-loop performance of the simplified robust GMC controller, a data-driven optimizer has been proposed to iteratively adjust the β -parameterized robust GMC controller using a set of closed-loop data, [51].

3.3.2 GMC Stability Analysis

Considering the closed loop system using GMC given by the equation:

$$\begin{aligned}\dot{\mathbf{y}} &= \hat{\mathbf{f}}(\mathbf{y}, \hat{\mathbf{f}}^{-1}(\mathbf{y}, \mathbf{K}_1(-\mathbf{y}) + \mathbf{K}_2(-\mathbf{z}), \mathbf{d}, t), \mathbf{d}, t) = \mathbf{K}_1(-\mathbf{y}) + \mathbf{K}_2(-\mathbf{z}) \\ \dot{\mathbf{z}} &= \mathbf{y} \\ \mathbf{y}(0) &= \mathbf{y}_0, \quad \mathbf{z}(0) = \mathbf{0}\end{aligned}$$

Equation 3-27

It has been shown that under the GMC explicit control law, if the model is perfect, there are no unmeasured disturbances, the sampling time of the controller is insignificant, and no process constraints exist, then the closed-loop system is always asymptotically stable for any choice of positive ξ_i and τ_i .

For any initial condition \mathbf{y}_0 of system Equation 3-27, there exist $M > 0$ and $\delta > 0$ such that if:

$$\|\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}, t) - \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \mathbf{d}, t)\| < \delta$$

Equation 3-28

for all \mathbf{x} and \mathbf{u} , all possible disturbances \mathbf{d} and all $t > 0$, then the solution of Equation 3-27 satisfies:

$$\|\mathbf{y}(t)\| < M, \quad \|\mathbf{z}(t)\| < M \quad \forall t \geq 0$$

Equation 3-29

Moreover, for any $\varepsilon > 0$ there exist positive numbers $\delta(\varepsilon)$ and $t_1(\varepsilon, \delta)$ such that if Equation 3-28 is true for all \mathbf{x} and \mathbf{u} , all possible disturbances \mathbf{d} and all $t > 0$, then the solution of Equation 3-27 satisfies Equation 3-29 and:

$$\|\mathbf{y}(t)\| < \varepsilon, \quad \|\mathbf{z}(t)\| < \varepsilon \quad \forall t \geq t_1(\varepsilon, \delta)$$

Equation 3-30

Thus, if the process/model mismatch is appropriately limited, the process outputs of the closed-loop system are bounded, which means that the closed-loop system is stable. The unmeasured disturbances can be considered as a special kind of process/model mismatch.

3.4 Further GMC Theoretical Developments

In this section, several theoretical developments will be briefly reviewed that have extended the basic GMC technique.

Lee and Zhou, [53], designed a new multivariable dead-time compensator under the GMC framework by decomposing an $n \times n$ MIMO dead-time problem into a series of n SISO feedback controllers, each with a feedforward compensation term.

Dunia et al., [54], implemented the Sliding Mode Control (SMC) for the GMC reference trajectory, and showed that SMC allows one to incorporate the effect of the uncertainty bounds in the controller structure, making GMC robust to processes with bounded uncertainties.

A new approach to Adaptive Generic Model Control (AGMC) was proposed order to further improve the performance of GMAC, [52]. Two AGMC schemes were developed.

The first scheme is based on estimation of time-varying parameters on-line in the embedded nonlinear model of GMC. The use of a new filter was proposed, called a Strong Tracking Filter (STF) to estimate time-varying parameters on-line. The negative influence of time-varying parameters on the control performance is effectively overcome, by updating them in every control period. This constitutes a parameter-estimation-based AGMC scheme.

The second scheme is based on the implementation of a new concept of Input Equivalent Disturbance (IED). When there are structural process/model mismatches, GMC cannot reject the influence of such disturbances quickly. A new concept of Input Equivalent Disturbance (IED) is introduced and the process/model mismatches are lumped into IED. This constitutes an IED-estimations-based AGMC scheme.

3.4.1 Relative Degree Considerations of GMC

It has been previously stated that GMC is only useful for systems where the relative order of all outputs is one. This is because the time derivatives of the outputs \dot{Y} must be directly dependent on the inputs U . In high relative order systems, this condition is not satisfied. The relative degree of a system is measured by the Lie derivative.

The Lie Derivative of a scalar function $g_i(X)$ in the direction of a vector function $f(X, U, D)$ is defined as:

$$L_f g_i(X, U, D) = \frac{\partial g_i}{\partial X^T} f$$

Equation 3-31

Note that this is a scalar function of X , U , and D . The Lie derivative is a linear derivative operator and its extended use is denoted by:

$$L_f^k g_i = \frac{\partial (L_f^{k-1} g_i)}{\partial X^T} f$$
$$L_f^0 g_i = g_i$$

Equation 3-32

The order of a dynamic system is generally the highest order of time derivative, which appears in the differential description of the system. Conventionally, the system order is the highest exponent of s appearing in the denominator of the transfer function. The relative order, however, is the order of the output, y_i , relative to the input vector, U . In mathematical terms, the output y_i has relative order r_i if:

$$\begin{aligned}\frac{\partial(L_f^k g_i)}{\partial U^T} &= [0 \dots 0], \quad k = 0, \dots, r_i - 1 \\ \frac{\partial(L_f^{r_i} g_i)}{\partial U^T} &\neq [0 \dots 0]\end{aligned}$$

Equation 3-33

For single-input single-output (SISO) systems in the Laplace domain, the relative order is the difference between the order of the denominator and the order of the numerator. In well-defined problems, every output y_i has a finite relative order. If this condition is not met, then the output is not dependent on the inputs and the system is not output-controllable.

Effectively, GMC solves the following equation:

$$(\dot{Y})^* = K_1(Y^* - Y) + K_2 \int (Y^* - Y) dt = \begin{bmatrix} L_f g_1 \\ \vdots \\ L_f g_m \end{bmatrix}$$

Equation 3-34

It is easy to see that if the relative order of any output y_i is greater than 1, the $L_f g_i$ will not be dependent on U and the above equation has no solution.

High relative order systems are common in the process industries and there are two modelling techniques for such systems so that, under a GMC framework, the closed

loop control is both stable and of satisfactory performance. The first group of techniques focuses on incorporating the detailed high relative order model into the GMC framework, [56]. This involves “difficulties” from an implementation point of view. The other group of techniques focuses on modelling high relative order systems as relative order one, and then applying them in the standard GMC framework, [57]. There are “imperfections” in both groups of techniques. Consequently, perfect control of high relative order systems via any of these methods is not possible. However, perfect control has never been possible via imperfect models.

Also, Robust Generic Model Control has been developed which expands the utility of GMC to higher relative order $RD > 1$ systems, [51].

3.4.2 Nonminimum Phase Systems

GMC can also be extended to address nonminimum phase systems. A method for the design of input-output linearizing feedback laws for nonlinear systems with unstable zero-dynamics has been proposed [58]. The approach is based on a suitable stable-antistable factorization of the zero-dynamics and guarantees internal stability of the compensated system.

“The method of approximated I/O-linearization of nonminimum phase nonlinear systems crucially depends on the existence and computability of suitable coordinates for the driven zero-dynamics. The requirement for (approximate) I/O-linearizability (of order k) is that the driven zero-dynamics can be transformed into a form, where the antistable part of the driven zero-dynamics (when expanded in a Taylor series up to order k) is firstly linear and secondly not influenced by the stable part of the driven zero-dynamics.

A systematic and general method for the transformation of the driven zero-dynamics into various forms was given. The method described is motivated by corresponding results from normal form theory of dynamical systems and contains the transformation into (approximately) antistable decoupled and (approximately) antistable linear form (of order k) is a special case.” [58]

Also, a decoupling strategy has been integrated into the Generic Model Control (GMC) framework to compensate for dynamic mismatch. The Multi-Model Decoupler (MMD) approach is based upon using a separate model inverse for the calculation of each manipulated variable. The use of separate model inverses allows the extra degrees-of-freedom required for the dynamic compensation. MMD-GMC is shown to offer significant control performance advantages over conventional GMC and PI control.

“From an overall point of view, MMD-GMC significantly improves the performance of GMC applied using steady-state models for distributed parameter processes in which there exists significant dynamic mismatch in the process. Compared to linear controllers, such as model predictive control, MMD-GMC has the advantage that it can provide nonlinear decoupling and nonlinear feed-forward compensation for measured disturbances.” [59]

3.5 Further GMC Implementation Issues

Another GMC implementation concern is integrator windup. It has been shown that integrator windup can be mitigated by restricting the absolute value of the integral

error to a user-set maximum value. However, the integral term is not required when the process model is adapted in order to eliminate any process / model mismatch, [60].

“Also, there are difficulties with the standard GMC structure in that it forces the same closed-loop time constant on the controlled variable, no matter what the current process dynamics are. If the closed loop time constant is given by Equation 3-35, then a way to overcome this problem is by modifying the normal GMC control law to Equation 3-36.

$$(y_{sp} - y) / \tau_c = \frac{dy}{dt}$$

Equation 3-35

$$f(y, x, u, t, d) = \frac{\tau_p}{\tau_c} (y_{sp} - y) + y$$

Equation 3-36

Equation 3-36 shows that the control moves made by the modified GMC controller are dependent on the ratio of τ_p to τ_c . The closed-loop time constant, τ_c , in Equation 3-36 was replaced with:

$$\tau_c = \alpha * \tau_p$$

Equation 3-37

Here, α is a tuning constant relating the closed-loop time constant to the open-loop time constant.” [60]

Also, process model selection is a key issue. One must strike a balance between detail and accuracy in the model and the quality of instrumentation that will provide the

inputs into the model. In order to capture important effects in the process, the instrumentation measuring the effects must be good.

Improvements can be made in collecting any process / model mismatch in an adjustable model parameter rather than letting the error accumulate in a controller integral term. Adapting a model parameter permits constant monitoring of the model performance, and can suggest areas in which the model may be improved, [61].

“GMC system identification is also an important issue. System identification methods build mathematical models of dynamical systems based on observed data. The intended use of the model should always be reflected in the methods and techniques used for identification. An identification scheme was derived for the case where the model was going to be used for GMC controller design. The aim of GMC control is to make the output approach a setpoint along a given desired trajectory. The identification emphasizes the output trajectories of the model. GMC control was improved by using an identification scheme based on a trajectory oriented predictor.” [62]

3.6 Comparing GMC to Other Controllers

GMC has been theoretically and experimentally compared with other linear and nonlinear controllers, [63,64].

For linear systems the GMC controller is equivalent to the reduced Internal Model Control (IMC) with the addition of two second order filters, [63].

Linear Multivariable Regulator (LMR) can be equivalent to the GMC approach control law with the a certain selection of \mathbf{K}_1 and \mathbf{K}_2 that are abnormal for the GMC

approach because they are not diagonal. However, these selections make the approach identical to the LMR.

The GMC approach is claimed to be much simpler than the Global Linearizing Control (GLC) structure although the two methods are essentially equivalent for relative degree one systems, [65]. Unlike most differential geometric techniques, GMC has been modified to accommodate time delays.

Babu, [66], found that GMC is well suited for a system when there is not a large rate of change of the setpoint and GLC for the system when the setpoint changes exponentially. In the former case there might be a chance of reaction temperature overshoot if the controller constants are changed to track.

Henson and Seborg, [67], proposed a unique unifying differential geometric approach (UDGA). “A new static state feedback control law that linearizes and decouples the closed-loop input-output response was developed. Since the approach is applicable to MIMO systems of any finite relative order in which the control variables enter linearly or nonlinearly, it unifies the GLC and GMC/Internal Decoupling methods. Additionally, the unified approach addresses controller synthesis for processes with measurable disturbances and implicit output equations.”[67]

“The GMC approach is better than the UDGA structure for the regulatory control problem with a non-zero constant setpoint. However, for relative degree one systems, the two methods are basically equivalent in the sense that a PI controller is applied for the derivative of the output, the control action is solved from the nonlinear model, and the measured disturbances can be compensated by the feedforward control action through the model.”

Linear Model Predictive Control is a discrete control method based on the convolution model. The GMC control law is based on the present and historic output measurements. The control movement ($\Delta \mathbf{u}_k$) based on these measurements and the process model provide the necessary control action to achieve the predefined GMC reference trajectories of the outputs.

The linear MPC control law is based on the present output measurements and the predicted future outputs. The predictions are made through the convolution model using the historic control action. The control movement provides the necessary control action to achieve the MPC reference trajectories of the outputs.

Assuming that the convolution model is perfect and there is no change in the disturbances, then it can be shown that the GMC approach control law is equivalent to the linear MPC control law with a specific selection of \mathbf{K}_1 and \mathbf{K}_2 .

In conclusion, there are several developed methods for nonlinear control each with their own advantages and disadvantages. Generic model control has been selected to be improved, and the creation of those improvements begins in the next chapter.

CHAPTER 4

PREDICTIVE GENERIC MODEL CONTROL

For this research project, the GMC controller will be studied and modified to develop a MIMO, nonlinear, robust controller. The initial, primary modification will be to reduce the control signal variation associated with a GMC controller. GMC control is already MIMO, and therefore extension to a new type of MIMO controller should be straightforward. After modifying GMC, the new controller design will be used to control an arc welding process.

4.1 GMC Control Variation

4.1.1 Introduction

Generic model control is a nonlinear control that incorporates the model of the process into the control law. However, the direct use of GMC can produce a control signal that frequently oscillates, which in a real system may not be implementable. A process constraint upon the change in the control could be given, but the calculation of the GMC control with process constraints becomes much more complex requiring nonlinear programming optimization, [49].

For example, an online optimization program could be run at each step k , so that the optimum K_1 and K_2 of the GMC control law would be determined that satisfies:

$$\min(\lambda_1(\dot{y} - \dot{y}_{GMC})^2 + \lambda_2(u_k - u_{k-1})^2)$$

Equation 4-1

where λ_1 and λ_2 are relative weights to emphasize either the reference trajectory accuracy or control variation reduction. However, this research is not focused on nonlinear optimization, and therefore other strategies for control variation reduction will be pursued.

In this chapter then, an approach is developed to smooth the control signal by the use of control predictions. The Predictive GMC (PGMC) method that is developed in this chapter reduces this control variation by influencing the control at a particular time with predicted future control inputs. Moreover, a method is presented for the selection of the controller parameters and the number of control predictions to be made. Simulations are then performed on linear and nonlinear systems, and compared with the original GMC method.

4.1.2 Offline Optimizer

Before choosing to smooth the control by use of control predictions, it was thought that an offline optimizer could be employed to select a particular K_1 and K_2 of the GMC control law to find the particular desired closed loop trajectory that would minimize the control oscillation. This optimizer was created and employed genetic algorithms to determine the optimum K_1 and K_2 . Genetic algorithms have shown to be useful in complicated function minimization and identification. The optimization routine was constructed as follows.

A response to a typical step input for the process could be simulated for a given K_1 and K_2 , and then the resulting response was used by the optimizer to adjust K_1 and K_2 to search for better results. The optimization is made up of a fitness function that you

would like to minimize by varying a design variable, in this case K_1 and K_2 . The fitness function in this case is given below:

$$Fitness = \frac{|Y_{CL}^{\infty} - Y_{OL}^{\infty}|}{OLG} - \sum |\Delta U| + K ,$$

Equation 4-2

where the output trajectory is subject to the following limitations:

$$Fitness = 0, \quad \text{if} \quad \begin{cases} \left| \frac{Y_{CL}^{ss} - Y_{CL}^{\infty}}{Y_{CL}^{end}} \right| \geq 2\% \\ \left| \frac{Y_{CL}^{ss} - Y_{ref}}{Y_{ref}} \right| \geq 2\% \end{cases} ,$$

Equation 4-3

where

$\sum |\Delta U|$ is the sum of the absolute value of the changes in the GMC from one step to another.

OLG is the open loop gain of the process.

Y_{CL}^{ss} is the closed loop system's output upon entering the steady-state region.

Y_{CL}^{∞} is the closed loop system's eventual output, if the last U_{GMC} remains constant.

Y_{OL}^{∞} is the open loop system's eventual output, if there is zero input.

For a genetic optimization, the design variable is controlled by the genetic algorithm using the following steps, [68]:

1. Take the two most “fit” numbers from a population of numbers, where fitness is defined by the function evaluation of the defined cost function.

2. Convert these numbers to binary format.
3. “Mate” the two best fit numbers by swapping their bits several times to create various “children” numbers.
4. Mutate the “children” numbers by flipping one or more of their bits.
5. Now a new population has been created and go back to step 1, unless the maximum number of generations has been reached.
6. Output the “most fit” number as the optimum.

The offline, genetic optimization was run on two different linear models and one nonlinear model. Insights for a method of directly selecting an optimum K_1 and K_2 were gained from the comparison of each of these optimizations.

The optimization of the nonlinear model in Equation 4-4 that is taken from a plasma welding process resulted in the optimization shown in Figure 4-1.

$$y_k = a_a + a_1 \cdot u_{k-1} + a_2 \cdot u_{k-2} \cdot y_{k-1} + a_3 \cdot u_{k-3} \cdot y_{k-2}$$

Equation 4-4

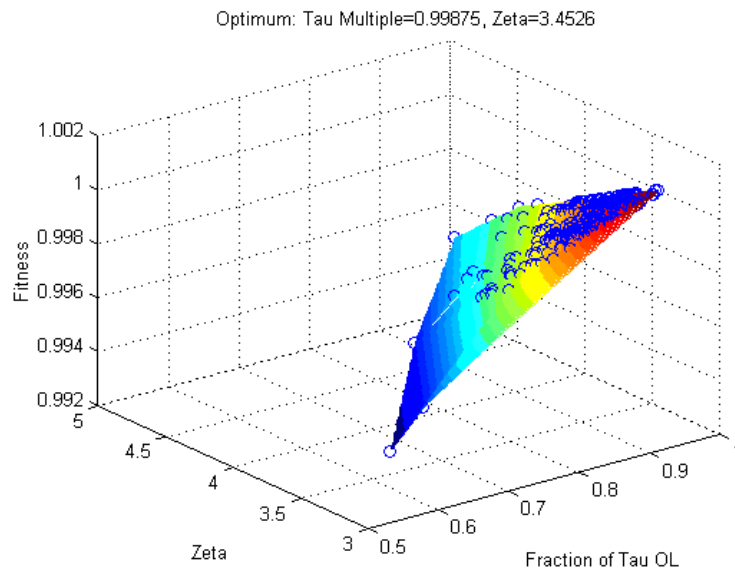


Figure 4-1 3D Optimization

The top-view of the optimization given in Figure 4-2 shows that an optimized region exists for $\xi \approx 4$ and $\tau = \tau_{\max}$. Moreover, there are constant optimization lines within $\{\xi, \tau\}$ space. If the points on one of these lines, for example the yellow region, can be converted into a diagram of the resulting poles and zeros of the GMC reference transfer function (Equation 3-4), the Figure 4-3 is created.

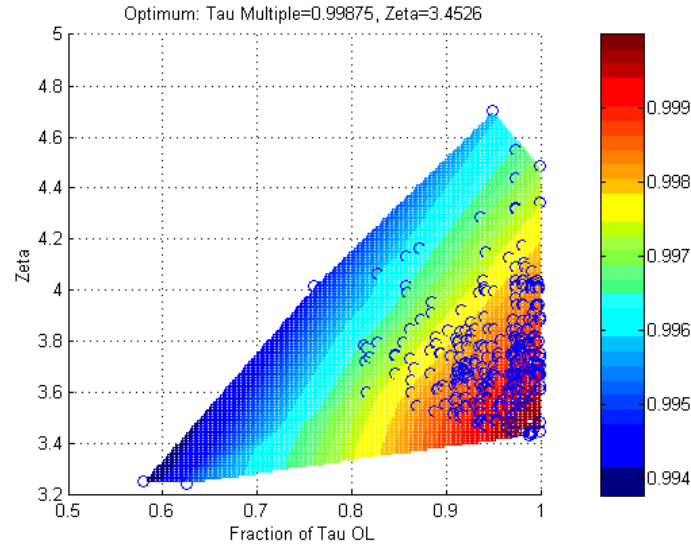


Figure 4-2 Top-View of Optimization

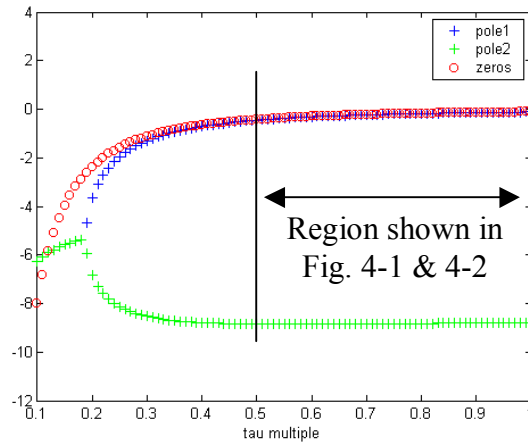


Figure 4-3 Pole & Zeros of GMC Transfer Function from Yellow Optimization Line

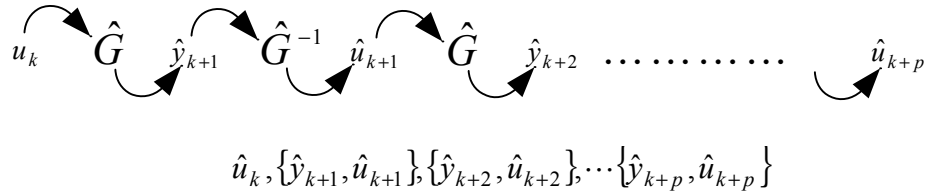
Figure 4-3 shows that for $\xi \geq 4$, there is pole-zero cancellation and that the constant optimization lines converge to this cancellation and a remaining constant pole. Therefore,

what seems to be a whole region in $\{\xi, \tau\}$ space is only a small variation of the GMC reference transfer function. It is thought then that this nonlinear example and other linear examples with similar results could be used to make the case that a *likely* optimum K_1 and K_2 can be selected using $\xi = 4$ and $\tau = \tau_{\max}$ for all processes. There is, however, no guarantee of this, and so to further ensure the control smoothness control predictions will be made.

4.2 Introduction to PGMC

Predictive GMC (PGMC) reduces the control signal variation associated with the original GMC by averaging predicted future controls that are each calculated using the original GMC method.

To illustrate, PGMC uses the original GMC method and the nonlinear process model in a “for” loop to calculate, p , predicted controls and outputs as:



Equation 4-5

The control and output predictions are made using the digital model process, process inverse, and GMC control law with the past inputs and outputs as shown in Equation 4-6.

$$\hat{u}_{k+n} = h \begin{pmatrix} \hat{x}_{k+n}, \dots, x_1, \hat{d}_{k+n}, \dots, d_1, t_{k+n}, \\ K_1(x_{ref,k+n} - \hat{x}_{k+n}), \\ K_2 \sum_{i=1}^{k+n} \frac{Ts}{2} \left(\begin{matrix} (x_{ref,i} - \hat{x}_i) \\ + (x_{ref,i-1} - \hat{x}_{i-1}) \end{matrix} \right) \end{pmatrix} (n = 0, 1, \dots, p)$$

$$y_{k+n} = \frac{dg}{dx} f \begin{pmatrix} \hat{x}_{k+n}, \dots, x_1, \\ \hat{u}_{k+n}, \dots, u_1, \\ \hat{d}_{k+n}, \dots, d_1, \\ \hat{t}_{k+n} \end{pmatrix} (n = 1, 2, \dots, p)$$

Equation 4-6

The actual control implemented, u_k , is set equal to a weighted average of the predicted controls:

$$u_k = \alpha_0 \hat{u}_k + \alpha_1 \hat{u}_{k+1} + \dots + \alpha_p \hat{u}_{k+p}$$

$$= \sum_{i=k}^{k+p} \alpha_{i-k} \hat{u}_i$$

Equation 4-7

where

$$\alpha_j = \frac{p-j}{(1+2+\dots+p)}, j = \{0, 1, \dots, p-1\}$$

Equation 4-8

The choice of α 's at this point in the control design is subjective and was simply selected to weight the control predictions the heaviest that are nearest to the present time.

PGMC offers two improvements over the original GMC method. First, the control signal variation associated with the original method is decreased. Moreover, because the predicted future control inputs exert influence on the present control, the present control steers the system's response to the desired future reference. Therefore, the PGMC method

is more applicable to a real system due to the control stability, and the control decision at a particular time is “smarter” because it considers the future reference.

Moreover, the original GMC control is an instantaneous prediction-based control for continuous systems or a one-step prediction-based control for discrete-time systems, and thus, can produce control variations. Its “nearsighted” nature is not suitable for the control of non-minimum phase systems. The PGMCM method proposed in this paper is a long-range prediction-based control. It reduces the control variation and thus, can better control non-minimum phase systems.

4.3 Analyzing PGMCM

In the original GMC method, the closed loop performance was determined by selecting the parameters ξ and τ . For PGMCM, not only do ξ and τ need to be selected but the prediction level, p , should be selected as well.

Consider the GMC closed loop transfer function where x is the system state:

$$\frac{x}{x_{ref}} = \frac{2\tau\xi s + 1}{\tau^2 s^2 + 2\tau\xi s + 1}$$

Equation 4-9

The closed loop poles and zero are given by

$$\begin{aligned}
pole\ 1 &= \frac{-\xi + \sqrt{\xi^2 - 1}}{\tau} \\
pole\ 2 &= \frac{-\xi - \sqrt{\xi^2 - 1}}{\tau} \\
zero &= \frac{-1}{2\xi\tau}
\end{aligned}$$

Equation 4-10

However, for $\xi \geq 4$, pole 1 \cong zero, and there is approximate pole/zero cancellation, which reduces the GMC closed loop transfer function to a first order response with a pole equal to *pole 2*. The settling time, using $\tau = 1$, for the GMC transfer function and the first order response are plotted versus ξ , in Figure 4-4, where the first order response's settling time is given by

$$T_{sett} = \frac{4}{pole\ 2}$$

Equation 4-11

The plot not only validates that the first order approximation is accurate for $\xi \geq 4$, but also reveals the specific pairs of ξ and τ , that will produce a particular desired closed loop settling time. By selecting a particular $\xi \geq 4$, then:

$$T_{sett, desired} = C \cdot \tau, \quad C = \text{constant}$$

Equation 4-12

The selection of the ξ, τ pair was the smallest *pole 2* in order to minimize the strength of the closed loop response and, subsequently, the control signal variation.

Therefore, for PGMC, ξ and τ are given by the following equations:

$$\left\{ \begin{array}{l} \xi = 4 \\ \tau = T_{sett, desired} / 0.43 \end{array} \right\}$$

Equation 4-13

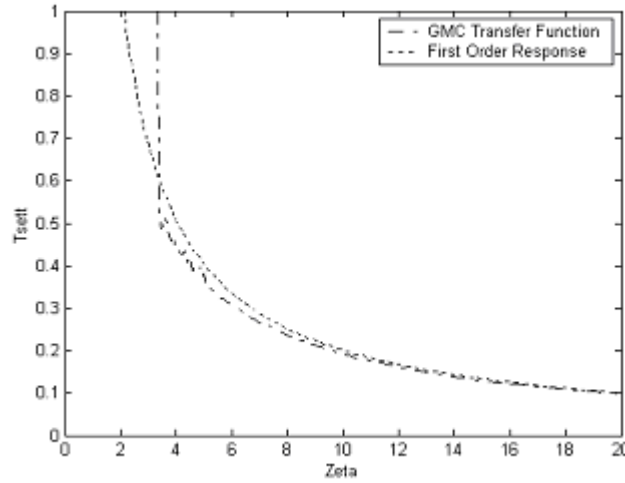


Figure 4-4 First Order Approximation

The ability of the PGMC method to minimize the control variation primarily stems from the averaging of the predicted future controls. The optimum number of steps that the control is predicted, p , is related to both the GMC closed loop performance curve, Figure 3-1, and the dynamics of the process. Although a quantitative solution for p has not been developed, the following qualitative analysis has revealed a sensible method for its selection.

For PGMC, the greatest advantage in influencing the present control with the predicted future controls, occurs when the change in the future control is large. The change in the future control decreases as closed-loop system approaches steady state. Thus, as the number of control predictions is increased towards the closed loop system's steady state, the rate of reduction in the control variation achieved by the predictions also decreases. Moreover, the number of control predictions can also affect the closed loop

settling time. For example, because the GMC method uses the process model to calculate the control action, if the process model has overshoot then the control predictions made are actually forcing the control energy to overcompensate and the settling time is reduced. Moreover, if enough control predictions are made during the overshoot period, then the control energy can be overly compensated such that the closed loop system overshoots the reference and the settling time is increased. These concepts are illustrated in Figure 4-5 for an underdamped system. The control variation multiple (CVM) was calculated using Equation 4-14.

$$CVM = OLG \frac{\sum |\Delta U_{PGMC}|}{|Y_{CL}^{\infty} - Y_{OL}^{\infty}|}$$

Equation 4-14

where

$\sum |\Delta U_{PGMC}|$ is the sum of the absolute value of the changes in the PGMC from one step to another.

OLG is the open loop gain of the process.

Y_{CL}^{∞} is the closed loop system's eventual output, if the last U_{PGMC} remains constant.

Y_{OL}^{∞} is the open loop system's eventual output, if there is zero input.

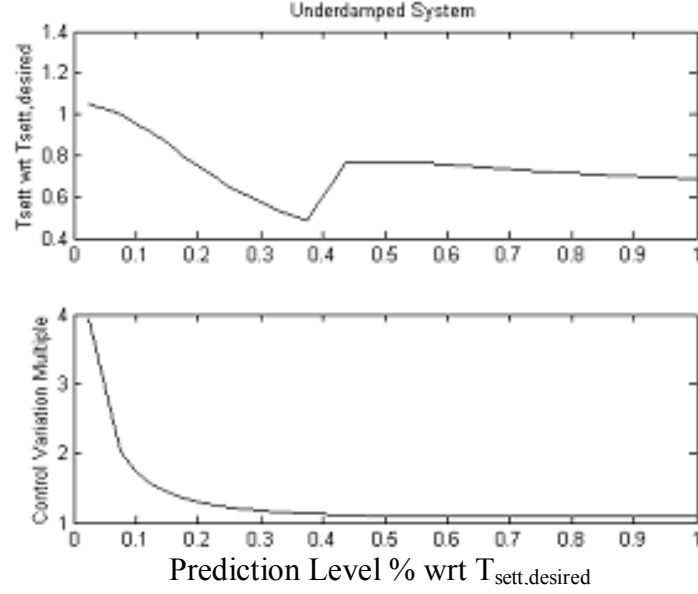


Figure 4-5 Prediction Level Improving Control Oscillations

From Figure 4-5, it can be seen that an optimum prediction level, p , could be described as the prediction level where most of the reduction in the control variation has occurred and yet the settling time is not significantly altered. In other words, the optimum p should be less than the closed loop rise time but as large as possible. Because the PGMC control predictions are based upon the GMC closed loop response, a sensible choice for the optimum p would be the dominant time constant of the GMC closed loop transfer function. Using the values of ξ and τ from Equation 4-13 to force pole/zero cancellation, the dominant time constant is equal to $1/pole\ 2$, Equation 4-11, and the optimum prediction level can be written as

$$p_{optimum} \approx round\left(\frac{0.65 \cdot T_{sett,desired}}{T_{sample}}\right)$$

Equation 4-15

Depending upon the dynamics of the open loop system slight adjustments to $p_{optimum}$ may improve the closed loop results. For example, if the open loop system is overdamped, then a slightly higher p would further reduce the control variation without affecting the settling time. This is evident when the GMC control law is organized for linear, minimum phase, zero disturbance, and no process/model mismatch as is assumed in predicting control values for linear systems. Using Equation 4-9 and 4-13, the control simplifies to

$$u = \left(\frac{18.61 \cdot T_{sett,desired} s + 1}{s^2} \right) G(s)^{-1} (y_{ref} - y)$$

Equation 4-16

The control signal behaves different for different plants due to $G(s)^{-1}$; however, for all plants the change in the signal is a function of the error $(y_{ref} - y)$. This is the reason that the dominant time constant of the closed loop output can be used to determine a proper amount of the control predictions to calculate.

Another consideration in selecting the prediction level, p , is that the number of computations per sampling time is limited. Therefore, there may be a practical constraint to the number of control predictions that can be made. Moreover, it should be noted for the controller designer that because control predictions are used to calculate the present control, the closed loop output will lead the reference. Also, as the number of control predictions is increased, the lead amount will also increase.

4.4 Examples

To demonstrate the advantages associated with PGMC, an underdamped linear system, an overdamped linear system, and a nonlinear system are selected to simulate the closed loop performance of both the original GMC and the developed PGMC. Moreover, to more fully compare the PGMC control, a nonlinear control algorithm called Interval Control, [69], is simulated for the nonlinear system.

The first linear system, which is an underdamped system, is given by Equation 4-17.

$$G(s) = \frac{200}{s^2 + 9s + 200} \text{ or}$$
$$G(z) = \frac{.02}{1.11z^2 - 2.09z + 1} \text{ for } T_s = 0.01 \text{ sec.}$$

Equation 4-17

For this system the open loop settling time is equal to 0.8 seconds and the open loop gain is equal to 1. It is desired to reduce the settling time by $\frac{1}{2}$. Thus, using Equation 4-13 and Equation 4-15 the controller parameters are:

$$\left\{ \begin{array}{l} \xi = 4 \\ \tau = .4 / 0.43 \\ p = 14 \end{array} \right\}$$

Equation 4-18

The closed loop output and control for both GMC and PGMC are plotted in Figure 4-6.

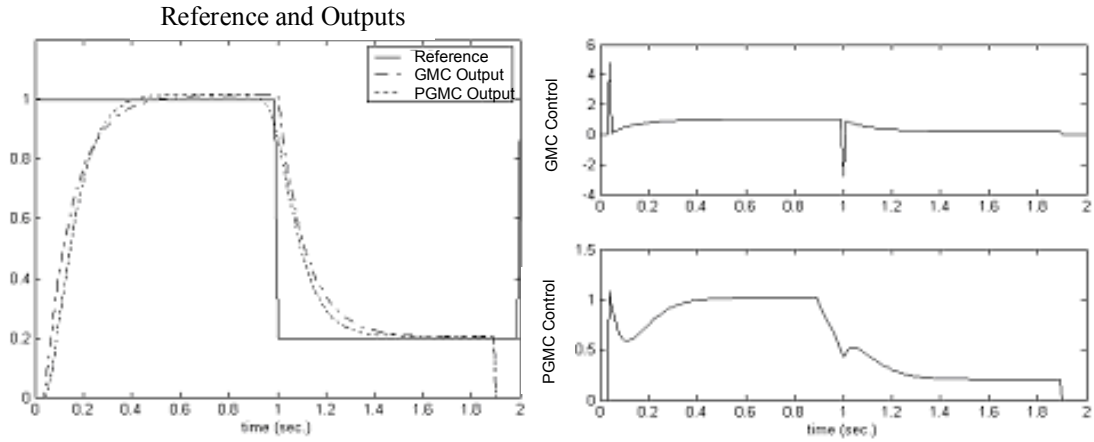


Figure 4-6 Closed Loop of GMC and PGMC

The PGMC achieved a 75% reduction in the control variation with respect to the GMC control. However, the actual PGMC settling time of the closed loop system equals 0.33, which is slightly faster than designed. This is because the open loop system is underdamped. If the p is reduced to 11, to obtain a slower settling time, the PGMC settling time decreases to 0.36, but the control variation reduction also decreases to 70%.

The second linear system, which is an overdamped system, is given by Equation 4-19.

$$G(s) = \frac{1.5}{s^2 + 14s + 40.02}$$

$$G(z) = \frac{.00015}{1.144z^2 - 2.14z + 1} \text{ for } Ts = 0.01$$

Equation 4-19

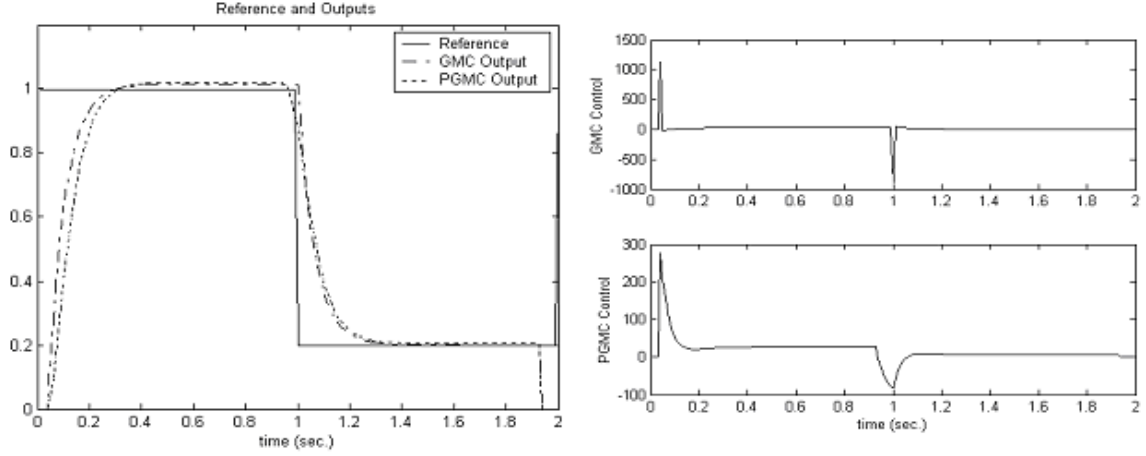


Figure 4-7 Linear System. 2: Closed Loop Output and Control for GMC and PGMC

For this system the open loop settling time is equal to 1.16 seconds and the open loop gain is equal to 0.0375. It is desired to reduce the settling time by $4/5$. Thus, using Equation 4-13 and Equation 4-15 the controller parameters are:

$$\left\{ \begin{array}{l} \xi = 4 \\ \tau = .232 / 0.43 \\ p = 7 \end{array} \right\}$$

Equation 4-20

The closed loop output and control for both GMC and PGMC are plotted in Figure 4-7. The PGMC achieved a 69% reduction in the control variation with respect to the GMC control. The actual PGMC settling time of the closed loop system equals 0.275 sec., which is slightly slower than designed. This is because the open loop system is overdamped. If the p is increased to 11, to obtain a faster settling time, the PGMC settling time decreases to 0.26 sec, but the control variation reduction also decreases to 60%.

In order to judge Predictive GMC's nonlinear control capability, it has also been compared with another nonlinear control method called Interval Control, [69]. Interval

control uses the model of the process to solve for the control at each sampling time such that if the control were kept constant for a number of steps, n , the closed loop output at that time n , would equal the reference. The nonlinear system, which is used in a plasma arc welding process developed at the University of Kentucky, is given by Equation 4-21.

$$y_k = a_0 + a_1 u_{k-1} + a_2 u_{k-2} y_{k-1} + a_3 u_{k-3} y_{k-2}$$

Equation 4-21

where $a_0 = 1949$, $a_1 = -0.9$, $a_2 = -9.9 \cdot 10^{-4}$, $a_3 = -5 \cdot 10^{-6}$

For this system the open loop settling time is equal to 0.005 seconds. It is desired to maintain the settling time and to track a reference trajectory. Thus, using Equation 4-13 and Equation 4-15 the controller parameters are:

$$\left\{ \begin{array}{l} \xi = 4 \\ \tau = 0.005 / 0.43 \\ p = 3 \end{array} \right\}$$

Equation 4-22

The closed loop output and control for both GMC and PGMC are plotted in Figure 4-8.

For this system the original GMC, not displayed, resulted in an unstable closed-loop response. However, the PGMC closed-loop output was not only stable, but the control also had less control variation than the interval model control. This demonstrates the advantage of PGMC over the original GMC method, and that PGMC control can be compared with other long range based prediction controllers.

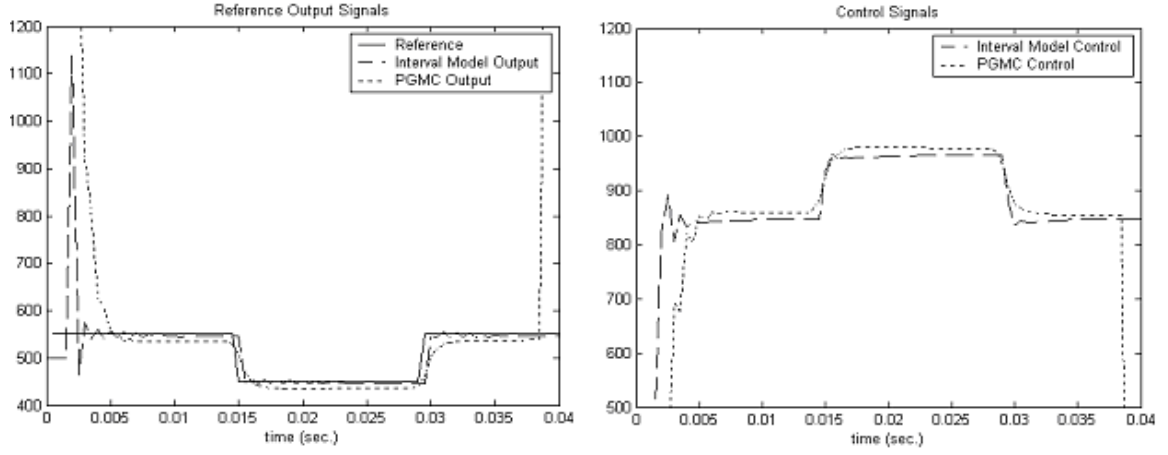


Figure 4-8 Nonlinear System: Closed Loop Output and Control for Interval and PGMC

To further test capabilities of the PGMC technique, a MIMO system is selected to control. The MIMO system is linear with the following parameters:

$$\mathbf{A} = \begin{bmatrix} -1 & -0.2 & -0.1 \\ -0.1 & -0.2 & -0.3 \\ -0.1 & -0.2 & -0.5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 1 & 1 \\ 3 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix}, \quad \mathbf{C} = \mathbf{I}_{3 \times 3}, \quad \mathbf{D} = [0]$$

Equation 4-23

The system is fully controllable with eigen values, $\lambda = -0.9967, -0.4372, 1.1338$. The solution for the GMC control signal is given by Equation 3-13. The desired settling time for each output was selected to be $T_{sett,desired} = 1.5$. The GMC control law, and therefore PGMC as well, already contains some degree of robustness due to the inclusion of the integral gain.

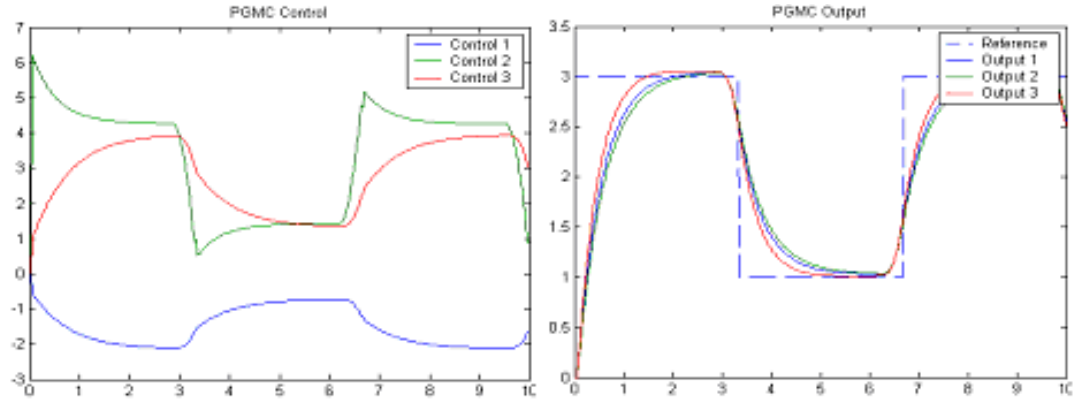


Figure 4-9 PGMC Control and Output for MIMO System

For example, if the previous MIMO system was incorrectly modeled so that the parameters of A were given as in Equation 4-24 then the resulting control and output is shown in Figure 4-9. Also, if the parameters of A were reduced or enlarged as in Equation 4-25 by 50%, the resulting control and output would be given by Figure 4-10. Over a longer period of time, the integral action to reduce the modelling error can be observed.

$$\hat{A} = \begin{bmatrix} -1 & .2 & -.1 \\ -.1 & .2 & -.3 \\ -.1 & -.2 & .5 \end{bmatrix}, \quad \hat{B} = B, \quad C = \hat{C}, \quad D = [0]$$

Equation 4-24

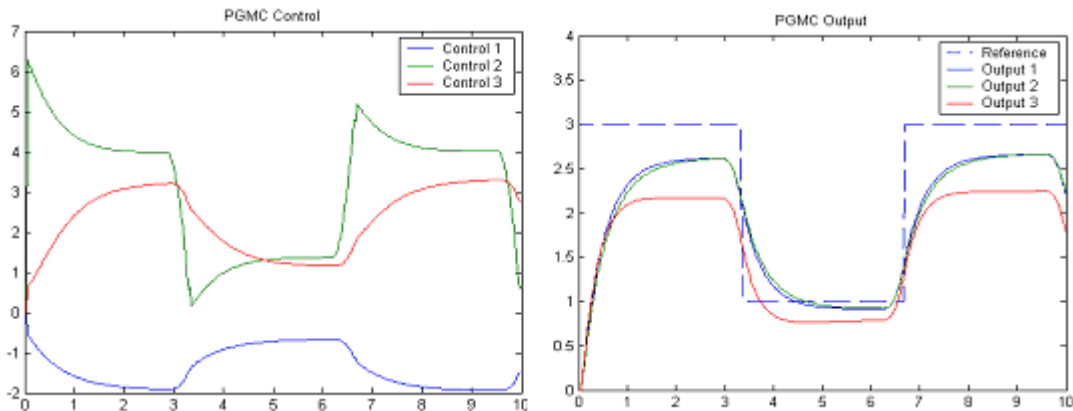


Figure 4-10 PGMC Control and Output for MIMO Uncertain System

$$\hat{\mathbf{A}} = \begin{bmatrix} -.5 & .4 & -.05 \\ -.2 & .1 & -.6 \\ -.05 & -1 & .25 \end{bmatrix}, \quad \hat{\mathbf{B}} = \mathbf{B}, \quad \mathbf{C} = \hat{\mathbf{C}}, \quad \mathbf{D} = [0]$$

Equation 4-25

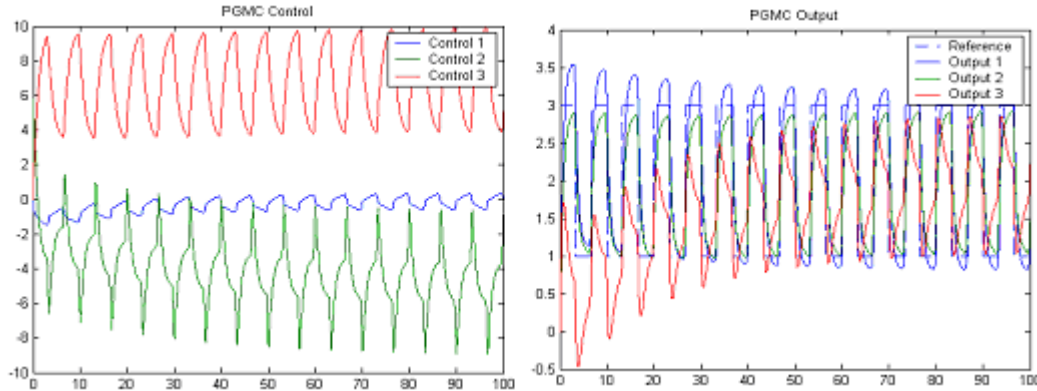


Figure 4-11 PGM Control and Output for MIMO Uncertain System

The PGM technique has proven valuable in efficiently controlling both linear and nonlinear systems. The important parameters the control designer specifies in using Predictive GMC are the desired closed loop settling time and the prediction level. However, under a certain amount of parameter or model structure mismatch the PGM control does not perform entirely as desired. Therefore, further testing needs to be completed to form a methodology in achieving a more robust PGM control algorithm. Moreover, for MIMO systems the proper application and use of the prediction level is not fully understood. If, for example, different states are required to reach their settling points at different times, because the PGM controller prediction level is based upon the desired settling time, then the actual proper prediction level to implement is not known.

Therefore, further work also needs to be accomplished in generalizing the algorithm for MIMO systems.

A likely possibility in improving the controller robustness will be a PGMC controller that uses parameter intervals instead of the parameters themselves in solving for the particular process input. Then a more intelligent control action could be taken in consideration of the process states and process parameter intervals. For example, if altering Equation 3-10 so that the control calculation can include the parameter interval, $\Delta\theta$, then the control would be

$$u = h(x, u, d, t, K_1(x_{ref} - x), K_2 \int (x_{ref} - x)dt, \Delta\theta)$$

Equation 4-26

Then a control could be calculated in consideration of the parameter interval that would minimize the output error. This could also be used to guarantee the stability of the closed loop process. This approach is what is accomplished in the following chapters.

CHAPTER 5

PARAMETER INTERVAL ADAPTIVE PGMC

In this chapter, the robustness of GMC will be improved by changing the control law to incorporate parameter intervals and adapting the process model used in the controller to move smoothly within those intervals.

5.1 GMC Parameter Interval System Model

A parameter interval system model assumes a model structure for a process and then embeds the uncertainty of the process into the parameter intervals. It may not even be completely necessary to fully have the correct model structure for during system identification, the inaccuracies of the model structure can simply be absorbed into the parameter intervals. The parameter intervals can be used on structural components involving all observable and unobservable states that have state estimation. This includes disturbance model components, which are a type of process state variable.

5.2 Identification of Parameter Intervals

The identification of the parameter intervals can be accomplished either heuristically or by experimentation. If done by experimentation, a series of experiments can be used in conjunction with a least-squares parameter identification routine to inform the designer of the nature of the parameter intervals. Experimental data can also be used

with routines that determine the parameter intervals directly according to a selected probable certainty condition, [71]. Once the parameter intervals are identified, the following designed method can make use of the intervals in closed loop control.

5.3 PGMC with Interval Factor

If the real process is given by:

$$y_{k+1} = f(x_k, u_k, d_k, t_k, \theta)$$

Equation 5-1

And the estimation of the process is given by:

$$\hat{y}_{k+1} = f(x_k, u_k, d_k, t_k, \hat{\theta}_k)$$

Equation 5-2

If all states are known and all variables in Equation 5-2 are measured such that they are positive, then a maximum and minimum prediction of the output can be made as:

$$\begin{aligned}\hat{y}_{\min, k+1} &= f(x_k, u_k, d_k, t_k, \theta_{\min}) \\ \hat{y}_{\max, k+1} &= f(x_k, u_k, d_k, t_k, \theta_{\max})\end{aligned}$$

where

$$\theta_{\min} \leq \hat{\theta} \leq \theta_{\max}$$

Equation 5-3

If the real output, y , moves smoothly with respect to the sampling time between y_{\max} and y_{\min} , then an estimation of y can be made by estimating the real parameters, θ_k , as:

$$\hat{\theta}_k = \theta_{\min} + F_k \cdot (\theta_{\max} - \theta_{\min}); \quad F_k = \frac{y_{k-1} - \hat{y}_{\min, k-1}}{\hat{y}_{\max, k-1} - \hat{y}_{\min, k-1}}$$

Equation 5-4

The assumption embedded in Equation 5-4 can be graphically pictured in Figure 5-1. If the control signal is smooth, then the output and output predictions should be relatively smooth in comparison to the sampling time. More importantly however is that the real output travels smoothly between y_{max} and y_{min} and thus an estimation of the position of y in relation to y_{max} and y_{min} could be based upon the previous sample.

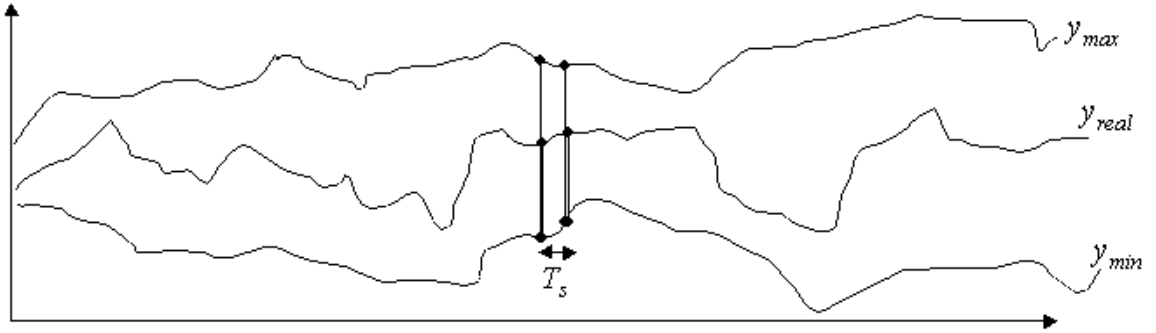


Figure 5-1 Depiction of Parameter Interval Factor

It is, however, pivotal to the soundness of this assumption that the control is smooth because the interval factor parameter, F_k , in Equation 5-4 is obviously influenced by the control as is shown in the following equation:

$$\frac{\partial F_k}{\partial t} = \frac{\partial}{\partial t} \cdot \frac{(y_{k-1} - \hat{y}_{\min,k-1})}{(\hat{y}_{\max,k-1} - \hat{y}_{\min,k-1})} = \frac{\partial}{\partial t} \cdot \frac{f(x_k, u_k, d_k, t_k, \theta - \theta_{\min})}{f(x_k, u_k, d_k, t_k, \theta_{\max} - \theta_{\min})}$$

Equation 5-5

The Equation 5-4 is used in conjunction with the PGMCM method of control so that given a F_{k-1} , a series of outputs and controls are predicted using the same F_{k-1} . Then the actual control that is implemented is equal to a weighted average of the predicted controls. The control smoothness is then improved by averaging the predicted controls, which consequently should also improve the validity of the assumption that the parameter interval factor changes only slightly between sampling times.

The conditions for which the above assumption is valid and how much the variation of F_k between sample points affects the output and control are extremely important issues. However, from experimental results, with an additional constraint on ΔF_k between sampling times, the closed loop results were promising. Moreover, the control algorithm is further modified, in Chapter 6, to improve the closed loop stability properties and control smoothness. The constraint upon ΔF_k was implemented by filtering F_k through the use of a moving average of previous F_k 's. The reasoning behind this constraint is that it is assumed that there is noise in the output signal, which if not recognized, actually changes F_k at a different rate than the actual process is changing F_k .

5.4 Experimental Results

For the plasma keyhole welding process, developed at the University of Kentucky, the control algorithm discussed in the previous section was implemented. The plasma keyhole process is illustrated in Figure 5-2.

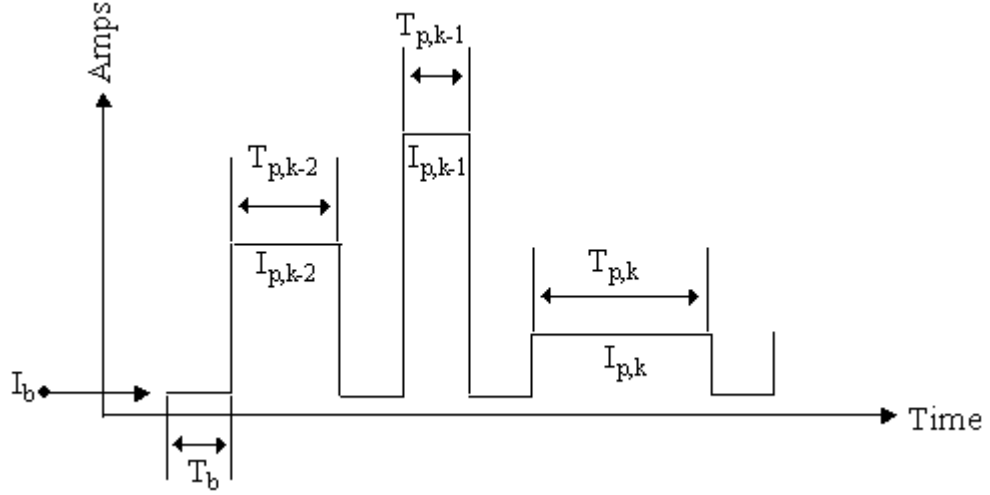


Figure 5-2 Description of Keyhole-Plasma Welding Process

The phenomenological model structure that was developed by consideration of the energy input into the system is given by:

$$T_{p,k} = a_0 + a_1 I_{p,k-1} + a_2 I_{p,k-2} T_{p,k-1} + a_3 I_{p,k-3} T_{p,k-2}$$

Equation 5-6

System identification was completed using random inputs within the likely process operating range, and a least squares algorithm was employed for determining the process parameters. This was completed four times to enable the construction of the probable max and min of each of the four model parameters. The range of each parameter is given as follows:

$$\begin{aligned} a_{0,\max} &= 734.66, & a_{1,\max} &= -4.96, & a_{2,\max} &= 0.7 \cdot 10^{-3}, & a_{3,\max} &= -14 \cdot 10^{-4} \\ a_{0,\min} &= 707.15, & a_{1,\min} &= -4.53, & a_{2,\max} &= -1.3 \cdot 10^{-3}, & a_{3,\max} &= -4 \cdot 10^{-4} \end{aligned}$$

Equation 5-7

After system identification, closed loop control was possible. In using the PGM control methodology and the parameter interval factor, the filter and the tunable control parameters were selected as follows:

$$T_s = 1, \quad T_{sett,desired} = 3, \quad p = 2, \quad F_{Filter} = \frac{4}{z^3 + z^2 + z + 1}$$

Equation 5-8

The results of two different closed loop control experiments are illustrated in Figure 5-3,4.

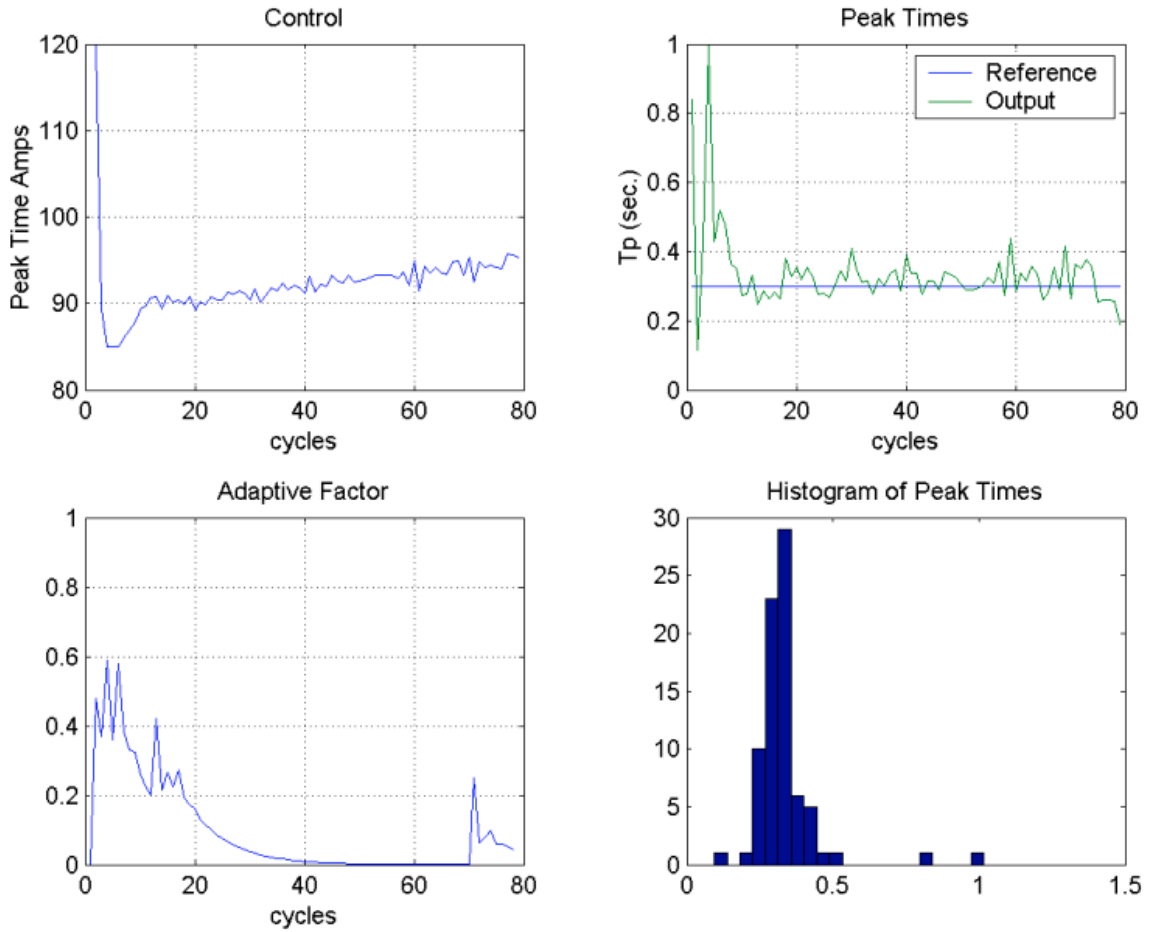


Figure 5-3 Experiment 1: Closed Loop Results

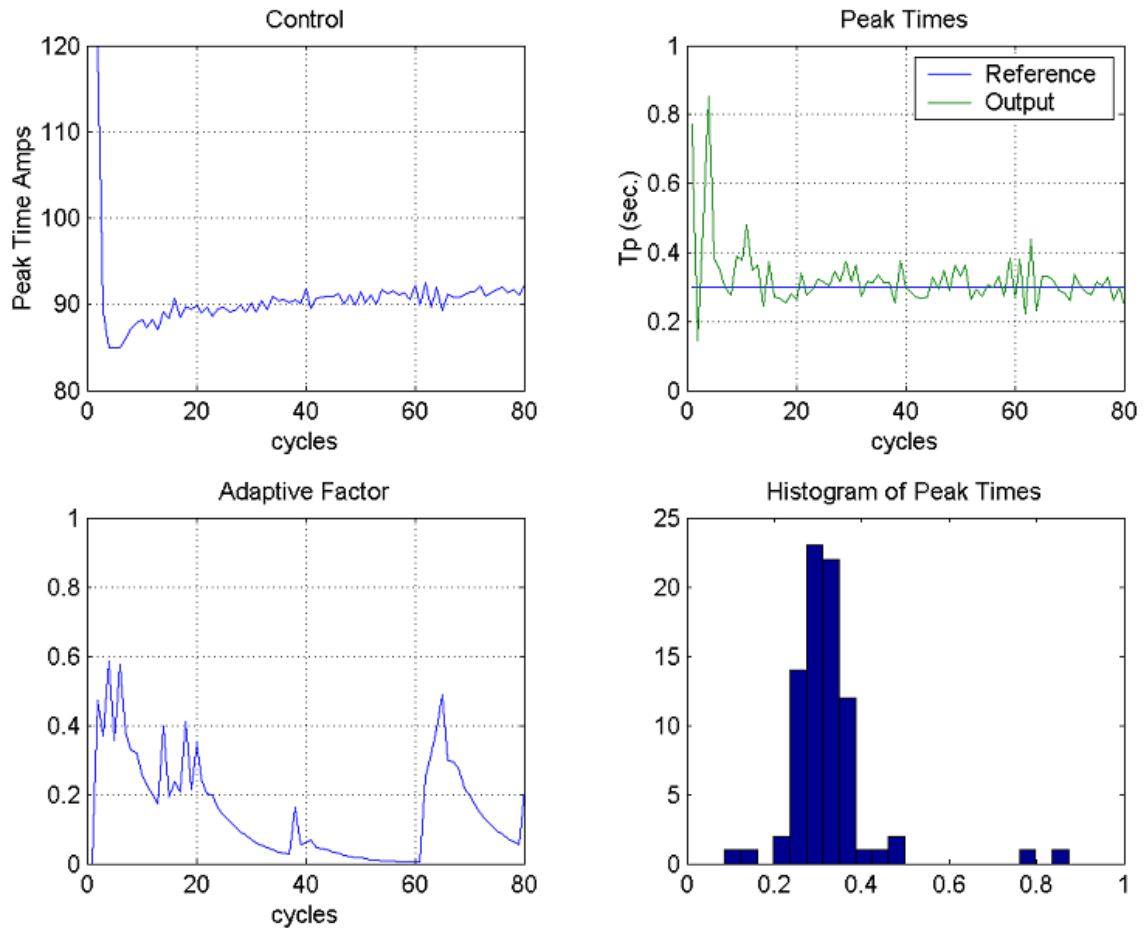


Figure 5-4 Experiment 2: Closed Loop Results

The goal of the keyhole plasma process is to obtain slightly overlapping keyhole penetration spots. This can be particularly seen in the zoomed backside view of the weld in the images given in Figure 5-5 and Figure 5-6. The goal of these overlapping penetration spots is to minimize the energy input that will achieve a quality weld.

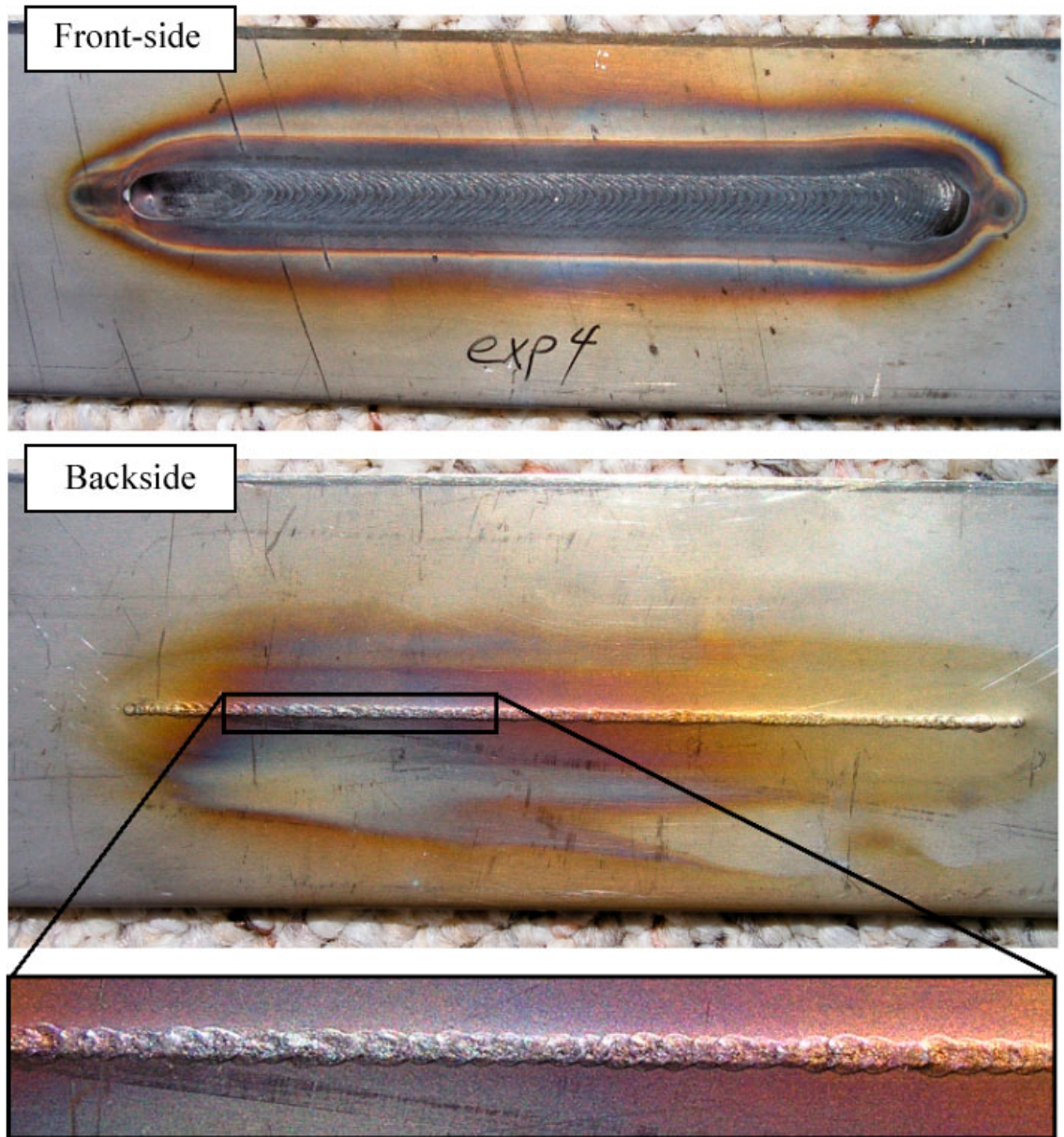


Figure 5-5 Weld Image of Keyhole Plasma Process

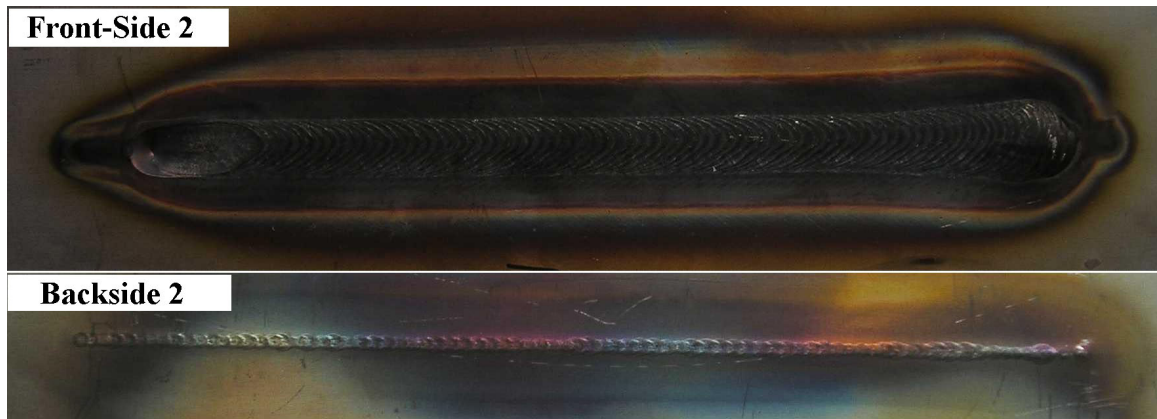


Figure 5-6 Weld Image 2 of Keyhole Plasma Process

The illustrations in Figures 5-3 and 5-4, show that after 5 to 10 cycles the output has reached the reference peak time. Although the parameter interval factor has varied widely throughout the entire process, variation between cycles is relatively small. The control signal is also relatively smooth because of the Factor filter. However, it is thought that an improvement on the control algorithm could better restrict the oscillation of the output about its reference. Also, because there is no methodology on the design of the Factor filter, it is thought a further improvement to the control algorithm could be accomplished to strive for control smoothness while maintaining closed loop stability. These are the goals that are attempted in the controller design in the next chapter.

CHAPTER 6

ROBUST PARAMETER INTERVAL PGM

6.1 Introduction

The previous chapter made use of the parameter interval models by simply assuming the real model moved smoothly relative to the sampling time between the maximum and minimum models. This section attempts to determine an optimum solution using the parameter intervals as uncertainty without the restricting assumption that the model move smoothly between the maximum and minimum. The algorithm makes use of interval mathematics, which is reviewed below.

6.2 Interval Mathematics [72, 73]

An interval $[x] \in \mathfrak{R}$ or real interval is a closed, connected, and bounded subset of \mathfrak{R} , such that

$$[x] = [x^-, x^+] = \{x \mid x^- \leq x \leq x^+\}$$

Equation 6-1

Interval arithmetic generalizes addition, subtraction, multiplication, and division to intervals. If, for instance, $x^- \leq x \leq x^+$, $y^- \leq y \leq y^+$ and $z = x + y$, then $x^- + y^- \leq z \leq x^+ + y^+$ so that the addition of two intervals is defined as

$$[x] + [y] = \{x + y \mid x \in [x] \text{ and } y \in [y]\} = [x^- + y^-, x^+ + y^+]$$

Equation 6-2

Similarly,

$$\begin{aligned}
-[x] &= \{-x \mid x \in [x]\} = [-x^+, -x^-] \\
[x] - [y] &= \{x - y \mid x \in [x] \text{ and } y \in [y]\} = [x^- - y^+, x^+ - y^-] \\
\text{If } 0 \notin [x], \text{ then } 1/[x] &= \{1/x \mid x \in [x]\} = [1/x^+, 1/x^-] \\
[x] * [y] &= [\min(x^-y^-, x^-y^+, x^+y^-, x^+y^+), \max(x^-y^-, x^-y^+, x^+y^-, x^+y^+)] \\
[x]^2 &= \{x^2 \mid x \in [x]\}
\end{aligned}$$

Equation 6-3

Using the relations above and the parameter intervals of the model, at each sampling time an optimum control and a control interval that makes the output stable will be calculated.

6.3 The SISO Control Algorithm

6.3.1 Finding the Optimum Control and Stability Interval

Before explaining the entire control algorithm, a few principles must first be reviewed. Given the interval mathematics problem below:

$$[Y^- \ Y^+] = [B^- \ B^+] \times [U^- \ U^+] + [H^- \ H^+]$$

Equation 6-4

Where the intervals $[B]$, $[U]$, and $[H]$ are known, the interval $[Y]$ is found using the Equations 6-1,2,3 to be:

$$\begin{aligned}
Y^- &= \min(B^-U^-, B^-U^+, B^+U^-, B^+U^+) + H^- \\
Y^+ &= \max(B^-U^-, B^-U^+, B^+U^-, B^+U^+) + H^+
\end{aligned}$$

Equation 6-5

If, however, it is known that the entire interval for both $[B]$ and $[U]$ are greater than zero, then the interval $[Y]$ can be found directly using:

$$Y^- = B^-U^- + H^-$$

$$Y^+ = B^+U^+ + H^+$$

Equation 6-6

Moreover, if given a feasible $[Y]$ interval for which it is known there is a corresponding $[U]$ that satisfies $[U] \geq 0$ then the inverse problem can be solved whereby for the given feasible $[Y]$ interval, the $[U]$ interval that satisfies Equation 6-6 is determined to be:

$$U^- = \frac{Y^- - H^-}{B^-}$$

$$U^+ = \frac{Y^+ - H^+}{B^+}$$

Equation 6-7

It is known that in general, the solution to this linear inverse interval problem simply described as $[A] [x] = [b]$ or $[A] [x] \leq [b]$ can only be solved through nonlinear programming methods. This is why there needs to be a restriction on the nature of some of the intervals to construct a closed form solution.

However, if the interval $[U]$ and the interval $[Y]$ have no restriction on their sign (i.e. U, U^+, Y, Y^+ can be above or below zero with constraint $U^+ \geq U, Y^+ \geq Y$), then the solution for the inversion problem to solve for $[U]$ given a $[Y]$ becomes more complicated. Assuming $[B] > 0$, the following equations show all the possibilities.

For $[B] \geq 0$ and $[Y]$ is feasible given $[B]$ and $[H]$

If $[U] \geq 0$

$$Y^+ = B^+U^+ + H^+ \Rightarrow U^+ = (B^+)^{-1} \cdot (Y^+ - H^+)$$

$$Y^- = B^-U^- + H^- \Rightarrow U^- = (B^-)^{-1} \cdot (Y^- - H^-)$$

If $U^- \leq 0, U^+ \geq 0$

$$Y^+ = B^+U^+ + H^+ \Rightarrow U^+ = (B^+)^{-1} \cdot (Y^+ - H^+)$$

$$Y^- = B^+U^- + H^- \Rightarrow U^- = (B^+)^{-1} \cdot (Y^- - H^-)$$

If $[U] \leq 0$

$$Y^+ = B^-U^+ + H^+ \Rightarrow U^+ = (B^-)^{-1} \cdot (Y^+ - H^+)$$

$$Y^- = B^+U^- + H^- \Rightarrow U^- = (B^+)^{-1} \cdot (Y^- - H^-)$$

Equation 6-8

Therefore, the method by which to calculate $[U]$ given $[Y]$ is not as straightforward as that given in Equation 6-7. However, it can be shown that the proper method to calculate $[U]$ given a feasible $[Y]$ is by the following equation.

$$U^- = \max \left(\frac{Y^- - H^-}{B^-}, \frac{Y^- - H^-}{B^+} \right)$$

$$U^+ = \min \left(\frac{Y^+ - H^+}{B^-}, \frac{Y^+ - H^+}{B^+} \right)$$

Equation 6-9

The method given by Equation 6-9 is found using the following analysis. If Y^- is chosen such that $(Y^- - H^-) < 0$, then given Equation 6-8, the correct and incorrect U^- are given by:

$$U_{correct}^- = (B^+)^{-1} (Y^- - H^-)$$

$$U_{incorrect}^- = (B^-)^{-1} (Y^- - H^-)$$

Equation 6-10

However, because $[B] \geq 0$, $U_{incorrect}^- \leq U_{correct}^-$ for all $(Y^- - H^-) \leq 0$. Also, if Y^- is chosen such that $(Y^- - H^-) > 0$, then given Equation 6-8, the correct and incorrect U^- are given by:

$$\begin{aligned} U_{correct}^- &= (B^-)^{-1} (Y^- - H^-) \\ U_{incorrect}^- &= (B^+)^{-1} (Y^- - H^-) \end{aligned}$$

Equation 6-11

However, again because $[B] \geq 0$, $U_{incorrect}^- \leq U_{correct}^-$ for all $(Y^- - H^-) \geq 0$. A similar analysis can be done for U^+ with the resulting generalization given in Equation 6-9.

Moreover, if there are stability constraints on the interval $[Y]$ given by:

$$[Y] \in [Y_{\min} \quad Y_{\max}]$$

Equation 6-12

Then Equation 6-9 can be used to find the stable interval $[U]_{stable}$ to satisfy the $[Y]$ constraints by:

$$[U]_{stable} = \left[\max \left(\frac{Y_{\min} - H^-}{B^-}, \frac{Y_{\min} - H^+}{B^+} \right), \min \left(\frac{Y_{\max} - H^-}{B^-}, \frac{Y_{\max} - H^+}{B^+} \right) \right]$$

Equation 6-13

These developed relations will be used in the derivation of the control algorithm in the following sections. Note that the derivation of Equation 6-13 assumes existence of the solution to the inverse interval problem of Equation 6-7,8 with the constraints of Equation 6-12. If there is no solution, the equation obviously does not apply.

Consider the nonlinear scalar model:

$$Y_k = \varphi_1 \cdot f_1(x_{1_k}) + \varphi_2 \cdot f_2(x_{2_k}) + \dots + \varphi_n \cdot f_n(x_{n_k}) + \beta \cdot g(x_k) \cdot u_{k-1}$$

Equation 6-14

Where $\varphi_i^{\min} \leq \varphi_i \leq \varphi_i^{\max}$ and $\beta^{\min} \leq \beta \leq \beta^{\max}$ which are found by system identification.

Note that this model structure is capable of being controlled by the GMC control technique. Also, although this model structure assumes that the model is control affine, if the real model is not control affine (i.e. U^2), then an in-between temporary calculation can make use of the model in Equation 6-14 (i.e. $U_{temp}=U^2$). Moreover, note that linear systems with state feedback fall within this structure. A case of output feedback is discussed later.

Assuming all of the states of equation 6-14 are known fully at each time k (i.e. $f_1(x_{1k}), f_2(x_{2k}), \dots, f_n(x_{nk}), g(x_k)$ are known), then the model can be simplified to an interval problem, for $[B_k] > 0$, given by:

$$\begin{bmatrix} Y_k^- & Y_k^+ \end{bmatrix} = \begin{bmatrix} H_k^- & H_k^+ \end{bmatrix} + \begin{bmatrix} B_k^- & B_k^+ \end{bmatrix} * \begin{bmatrix} U_k^- & U_k^+ \end{bmatrix}$$

where

$$\begin{aligned} H_k^- &= \sum_{i=1}^n \min(\varphi_i^{\min} \cdot f_i(x_{ik}), \varphi_i^{\max} \cdot f_i(x_{ik})) \\ H_k^+ &= \sum_{i=1}^n \max(\varphi_i^{\min} \cdot f_i(x_{ik}), \varphi_i^{\max} \cdot f_i(x_{ik})) \\ B_k^- &= \beta^- \cdot g(x_k) \\ B_k^+ &= \beta^+ \cdot g(x_k) \end{aligned}$$

Equation 6-15

The utility of Equation 6-14 is that the nonlinear model is reduced to an interval equation of the form given in Equation 6-4. If $[B_k] \geq 0$, then a closed form optimal and stable solution for the inverse interval problem can be found.

If there is a desired scalar output for the system in Equation 6-4,15 called $Y_{ref,k}$, then a sensible choice for the desired interval $[Y_k]$ would be centered upon $Y_{ref,k}$ such that:

$$(Y_k^+ + Y_k^-) / 2 = Y_{ref,k}$$

Equation 6-16

Assuming that the system moves within the parameter intervals with equal likelihood, this choice of the desired interval $[Y_k]$ will maximize the probability of actually achieving $Y_{ref,k}$. Thus, the choice at the center of $[Y_k]$ is known; however, the actual values of Y_k^+ and Y_k^- are not determined except that there is a restriction that $[Y_k] \geq [H_k]$.

The selection of the desired $[Y_k]$ is really a function of the system states and parameter intervals as well as the desired scalar output $Y_{ref,k}$. Thus, let $[Y_k]$ be given as:

$$\begin{aligned} Y_k^- &= Y_{ref,k} - C_k \\ Y_k^+ &= Y_{ref,k} + C_k \end{aligned}$$

Equation 6-17

such that Equation 6-16 is satisfied. Also, choosing the interval restrictions:

$$\begin{aligned} 1. \quad & Y_k^+ \leq Y_{\max} \\ 2. \quad & Y_k^- \geq Y_{\min} \\ 3. \quad & Y_k^+ \geq Y_k^- \\ 4. \quad & U_k^+ \geq U_k^- \\ 5. \quad & U_k^- \geq U_{\min} \\ 6. \quad & U_k^+ \leq U_{\max} \end{aligned}$$

Equation 6-18

Where U_{\min} , U_{\max} , Y_{\min} and Y_{\max} are assigned by the control designer from knowledge of the process. Then, if there exists a desired interval $[Y_k]$ as well as the interval $[U_k]$ to achieve it within the interval restrictions in Equation 6-18, the solution is found using the following method constructed below.

The smallest interval $[U_k]$ to satisfy Equation 6-16 is for the elements of $[U_k]$ to be equal, in other words, not an interval at all. Thus the optimum C_k value can be solved for the cases $[U]>0$ or $[U]<0$ using $U_k^+ = U_k^-$, $Y_k^+ = Y_{ref,k} + C_k$, and $Y_k^- = Y_{ref,k} - C_k$ by solving

$$(B^-)^{-1} \cdot (Y_{ref,k} - C_k - H_k^-) = (B^+)^{-1} \cdot (Y_{ref,k} + C_k - H_k^+)$$

Equation 6-19

To obtain:

$$C_k = \frac{1}{\left(1 + \frac{B^+}{B^-}\right)} \cdot \left\{ (Y_{ref,k} - H_k^-) \left(\frac{B^+}{B^-} \right) + H_k^+ - Y_{ref,k} \right\}$$

Equation 6-20

Then the optimum output interval is equal to:

$$[Y]_{opt} = [Y_{ref,k} + C_k, Y_{ref,k} - C_k]$$

Equation 6-21

And the optimum U_k is given by:

$$U_k = \frac{Y_{ref,k} + C_k - H_k^+}{B^+} = \frac{Y_{ref,k} - C_k - H_k^-}{B^-}$$

Equation 6-22

It will now be shown that if $Y_{ref,k}$ is calculated using generic model control, the optimum U_k given in Equation 6-22 is equal to the control calculated using generic model control, U_{GMC} . If generic model control were used then $Y_{ref,k}$ would be calculated, assuming digital control by:

$$Y_{ref,k} = T_s \left\{ K_1 (Y_{set,k-1} - Y_{k-1}) + K_2 \sum_{i=1}^{k-1} T_s \cdot (Y_{set_i} - Y_i) \right\} + Y_{k-1}$$

Equation 6-23

Then assuming that each parameter of the process model is given by the center of its interval, the generic model control, U_{GMC} , would be calculated by:

$$U_{GMC} = \frac{Y_{ref,k} - \bar{H}_k}{\bar{B}}$$

where

$$\bar{H}_k = (H_k^- + H_k^+)/2 \quad \text{and} \quad \bar{B} = (B^- + B^+)/2$$

Equation 6-24

Because the optimum U_k is not an interval, in other words $U_{opt,k}^- = U_{opt,k}^+$, then only the interval problem for $[U]>0$ or $[U]<0$ need be considered (i.e. if $U_k^- = U_k^+$, then it cannot be that $U^- < 0$ and $U^+ > 0$). Then using Equation 6-17 and the cases from Equation 6-8 for $[U]>0$, $[U]<0$, with $U=U^+$ the following derivation can be made:

If $[U] \geq 0$, then

$$Y_{ref,k} + C_k = B^+ U_k + H_k^+$$

$$Y_{ref,k} - C_k = B^- U_k + H_k^-$$

Equation 6-25

Adding the two equations and multiplying both sides by $1/2$

$$\text{Produces} \Rightarrow Y_{ref,k} = \frac{1}{2} (B^+ + B^-) U_k + \frac{1}{2} (H_k^+ + H_k^-)$$

Equation 6-26

Then solving for U_k

$$\text{Produces} \Rightarrow U_k = \frac{Y_{ref,k} - (H_k^+ + H_k^-)/2}{(B^+ + B^-)/2} = U_{GMC}$$

Equation 6-27

Similarly if $[U] \leq 0$, then

$$\begin{aligned} Y_{ref,k} + C_k &= B^- U_k + H_k^+ \\ Y_{ref,k} - C_k &= B^+ U_k + H_k^- \end{aligned}$$

Equation 6-28

Adding the two equations and multiplying both sides by $\frac{1}{2}$

$$\text{Produces} \Rightarrow Y_{ref,k} = \frac{1}{2}(B^+ + B^-)U_k + \frac{1}{2}(H_k^+ + H_k^-)$$

Equation 6-29

Then solving for U_k

$$\text{Produces} \Rightarrow U_k = \frac{Y_{ref,k} - (H_k^+ + H_k^-)/2}{(B^+ + B^-)/2} = U_{GMC}$$

Equation 6-30

Thus, to calculate the optimum $U_{opt,k}$ of Equation 6-22, one need only to calculate the generic model control, U_{GMC} , with the model parameters set equal to their interval center.

However this, U_{GMC} ,calculated control may violate the constraints given in Equation 6-18. Thus, the optimum input $U_{opt,k}$ must be constrained so that Equation 6-18 is satisfied. For example assuming $[B] \geq 0$, the constraints upon $U_{opt,k}$ from Equation 6-18 upon U_k with $Y_{min}=0$ and $U_{min}=0$ are given as:

$$\begin{aligned}
U_k^-, U_k^+ \geq 0 &\Rightarrow U_k \geq 0 \\
Y_k^- \geq 0 &\Rightarrow U_k \geq -\frac{H_k^-}{B^-} \\
Y_k^+ \geq 0 &\Rightarrow U_k \geq -\frac{H_k^+}{B^+} \\
Y_k^+ \leq Y_{\max} &\Rightarrow U_k \leq \frac{Y_{\max} - H_k^+}{B^+}
\end{aligned}$$

Equation 6-31

Then the control U_{opt_k} that is implemented is given by

$$U_{opt_k} = \min \left(\max \left(-\frac{H_k^-}{B^-}, -\frac{H_k^+}{B^+}, 0, \frac{Y_{ref_k} - \bar{H}_k}{\bar{B}} \right), \frac{Y_{\max} - H_k^+}{B^+} \right)$$

Equation 6-32

or more generally as:

$$U_{opt_k} = \min \left(\max \left(\frac{Y_{\min} - H_k^-}{B^-}, \frac{Y_{\min} - H_k^+}{B^+}, U_{\min}, \frac{Y_{ref_k} - \bar{H}_k}{\bar{B}} \right), \frac{Y_{\max} - H_k^+}{B^+}, U_{\max} \right)$$

Equation 6-33

The control interval at time k that ensure process stability defined by Equation 6-18, called $[U_k]_{stable}$, is given as:

$$[U_k]_{stable} = \left[\max \left(\frac{Y_{\min} - H_k^-}{B^-}, \frac{Y_{\min} - H_k^+}{B^+}, U_{\min} \right), \min \left(\frac{Y_{\max} - H_k^+}{B^+}, U_{\max} \right) \right]$$

Equation 6-34

Therefore at this point in the control algorithm, an optimum control is known and a control interval that makes the process stable is known. However aside from the stability guarantee, the control will not be greatly improved over the normal generic model control. Thus, a further improvement is required.

6.3.2 Inducing Control Smoothness

From the previous chapter, predictive generic model control (PGMC) was found to be helpful in inducing control smoothness. Thus, if PGMC can be combined with the stability guarantees of the previous section, a truly robust/smooth control could be determined. The following analysis derives this profitable combination.

If during the control calculation at time k , the model of the process is simulated p steps ahead such that the following control predictions are determined using Equations 6-33 and 6-34: $[U_k]_{stable}, \hat{U}_{opt\ k}, \hat{U}_{opt\ k+1}, \hat{U}_{opt\ k+2}, \dots, \hat{U}_{opt\ k+p}$, where $\hat{U}_{opt\ i}$ denotes a prediction during the simulation. Then a new control called \tilde{U}_k could actually be implemented, which is a weighted average of $U_{k-1}, \hat{U}_{opt\ k}, \hat{U}_{opt\ k+1}, \hat{U}_{opt\ k+2}, \dots, \hat{U}_{opt\ k+p}$ to induce control smoothness with the following stability constraint:

$$\tilde{U}_k \in [U_k]_{stable}$$

Equation 6-35

The method by which the predictions $\hat{U}_{opt\ k+1}, \dots, \hat{U}_{opt\ k+p}$ are made is by updating the model of Equation 6-14 and the corresponding interval problem (Equation 6-15) and interval problem solution (Equation 6-33) with the new states H_{k+i}^-, H_{k+i}^+ calculated using:

$$U_{k+i} = \hat{U}_{opt\ k+i}$$

$$Y_{k+i} = \hat{Y}_{k+i}^- + F_k \cdot \hat{Y}_{k+i}^+ - \hat{Y}_{k+i}^- \quad \text{for} \quad U_{k+i} = \hat{U}_{opt\ k+i}$$

Equation 6-36

where F_k can become an adaptive factor as given in Section 5.3, and can be either set to a worst case scenario or to its likely value according to previous history.

For example, as in Section 5.3, a prediction can be made for Y_k^{\max}, Y_k^{\min} using the control calculated at time $k-1$. Then a sensible choice for F_k would be equal to:

$$F_k = \frac{Y_k - Y_k^{\min}}{Y_k^{\max} - Y_k^{\min}}$$

Equation 6-37

Moreover, for calculating the predictions U_{k+2}, \dots, U_{k+p} a sensible choice for

F_{k+1}, \dots, F_{k+p} would be:

$$F_{k+1} = F_{k+i-1} + \Delta F_k$$

Equation 6-38

Where ΔF_k can then become a control parameter to either find a worst case design by alternating its sign or by simulating the system's likely change in F . Then again, after the predictions are made, the actual control implemented, \tilde{U}_k , is equal to

$$\min(U_{k_{stable}}^+, \max(U_{k_{stable}}^-, \tilde{U}_k))$$

where

$$\tilde{U}_k = W_0 U_{k-1} + W_1 U_k + W_2 U_{k+1} \dots W_p U_{k+p}$$

Equation 6-39

Then tunable parameters of the control then become: $Y_{max}, Y_{min}, U_{max}, U_{min}, p, \Delta F_k, W$.

The first four parameters change the allowable output and input and can be relaxed or lessened to suit stability and performance criteria. They could be static or they could change with time or alternatively the designer could make them a function of $Y_{ref,k}$. The ΔF_k parameter is used in calculating the predictions at each sampling time to simulate the likely movement of the process throughout its parameter interval and can be used to simulate the worst case scenario or something less stringent. The prediction level, p , and

the control weights, W_i , affect control smoothness. As referenced in the previous chapter, increasing p to thirty percent of the dominant time constant of the GMC reference trajectory will significantly reduce the control oscillation.

To review the final control algorithm, the following steps are stated:

1. Calculate the control interval that ensure process stability, $[U_k]_{stable}$, (Equation 6-34)
2. Calculate interval factor, $F_k = \left\{ \frac{Y_{pred_{k-1}}^{\min} - Y_k}{Y_{pred_{k-1}}^{\max} - Y_{pred_{k-1}}^{\min}} \right\}$ where $Y_{pred_{k-1}}^{\min}$, $Y_{pred_{k-1}}^{\max}$ are calculated in the previous sampling time.
3. In a for loop (i=0...p-1) where p = prediction level:

Using the process model (Equation 6-15) and optimum control Equation 6-33, calculate $[H_{k+i}]$, $U_{k+i\ opt}$.

Then for next iteration's states, $[H_{k+i+i}]$, calculate:

$$Y_{k+i+1} = Y_{opt_{k+i+1}}^- + (F_{k+i} + \Delta F_{k+i+1}) \cdot (Y_{opt_{k+i+1}}^+ - Y_{opt_{k+i+1}}^-)$$

$$U_{k+i+1} = U_{opt_{k+i+1}}$$

4. Then implement $\tilde{U}_k = \sum_{i=k-1}^p W_i \cdot U_i$

with constraint $\{U_{k_{stable}}^- \leq U_k \leq U_{k_{stable}}^+\}$

5. Finally, calculate $Y_{pred_k}^{\min} = \min(B^- \cdot \tilde{U}_k, b^+ \cdot \tilde{U}_k) + H_k^-$
 $Y_{pred_k}^{\max} = \max(B^- \cdot \tilde{U}_k, b^+ \cdot \tilde{U}_k) + H_k^+$ for the next sampling time.

One further comment is that if the parameter interval of a particular parameter of Equation 6-14,15 is thought to be too large, then assuming that the probability density function of the actual process parameter lying within its interval is given by a normal distribution such that:

$$\varphi_i = N(\mu_i, \sigma_i) \quad \text{where} \quad \mu_i = \frac{\varphi_i^{\max} - \varphi_i^{\min}}{2}, \sigma_i = \frac{\varphi_i^{\max} - \varphi_i^{\min}}{4}$$

$$\beta = N(\mu_\beta, \sigma_\beta) \quad \text{where} \quad \mu_\beta = \frac{\beta^{\max} - \beta^{\min}}{2}, \sigma_\beta = \frac{\beta^{\max} - \beta^{\min}}{4}$$

Equation 6-40

then a new interval can be chosen based upon the parameter's σ with a knowledge of the probability that the new interval will contain that the real process parameter. However, if the new interval is smaller than the real process interval then process stability is no longer guaranteed.

6.4 SISO Experimental Results

6.4.1 Linear Scalar Example with Implementation

A first order system describing a soft plasma arc welding process, developed at the University of Kentucky, was controlled using the parameter interval PGMC algorithm. The first order parameter interval model is given by

$$Y_{k+1} = a \cdot Y_k + cU_k$$

where

$$0.5 \leq a \leq 0.75$$

$$1.5E-5 \leq c \leq 3.75E-4$$

Equation 6-41

For this process, the plasma arc oscillates between a high current called a peak current and a low current called a base current. The value of the peak and base current are predetermined, and only the amount of peak current duration is altered. With a longer peak current duration, a greater amount of energy is released into the workpiece and the

weld penetration is increased. The process model describes a relation between the peak current duration, U , and the output of a sensor measuring weld penetration, Y . The first order model was first simulated to obtain reasonable control parameter values for $\tau_{desired}$, p (prediction level), W , and ΔF_k . The open loop step responses are graphed in Figure 6-1 for a constant input of 700 to show the varying time constants and open loop gains of the process.

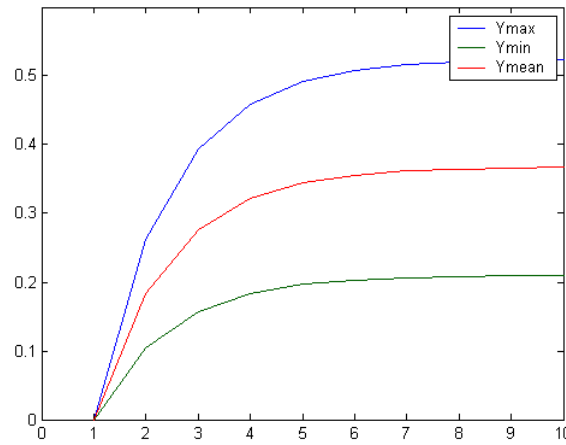


Figure 6-1 Open Loop Step Responses for Ymin, Ymax, Ymean

The closed loop system was simulated for both the normal GMC technique and the parameter interval PGMC using:

$$\tau_{desired} = 4$$

$$p = 3$$

$$W = 1$$

$$\Delta F_k = 0.5$$

$$Y_{max} = 0.9$$

$$Y_{min} = 0$$

The control is shown in Figure 6-2 and the closed loop output is shown in Figure 6-3.

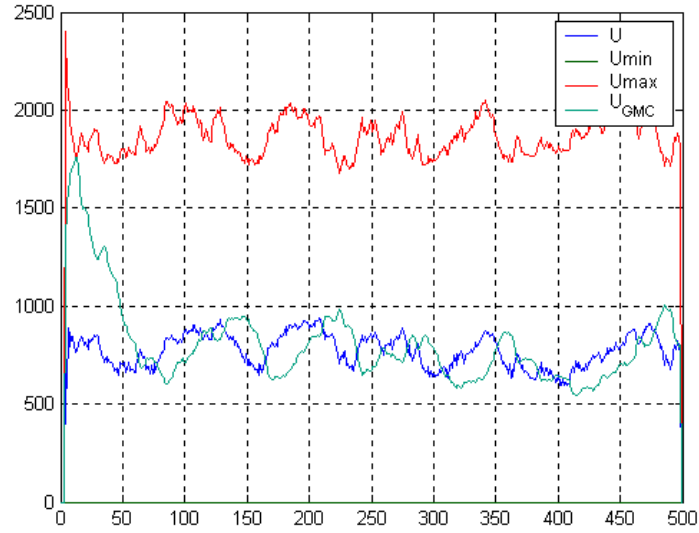


Figure 6-2 Control Signal of Linear Simulation

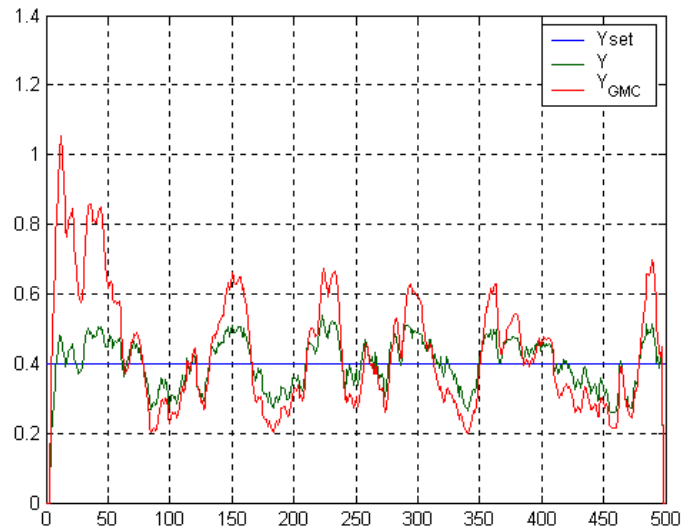


Figure 6-3 Output Signal of Linear Simulation

From Figure 6-3 it is apparent that parameter interval PGMC not only has the benefit of a stability criteria included in the control law, but the closed loop output is more accurate due to the parameter adaptation and the control smoothness, which is generated by the averaging of the control predictions. After determining the region of the control parameter values via simulation, the parameter interval PGMC was implemented

in the real arc welding process. Six experimental data sets are shown in Figure 6-4 and Figure 6-6 with the following set of output setpoint and controller parameters:

	Y_{set}	τ	p	W	ΔF
exp1	0.45	4	3	1	0.3
exp2	0.50	3	2	1	0.3
exp3	0.50	3	2	0	0.5
exp4	0.20	3	2	0	0.2
exp5	0.40	3	1	0	0.1
exp6	0.40	3	1	0	0.1

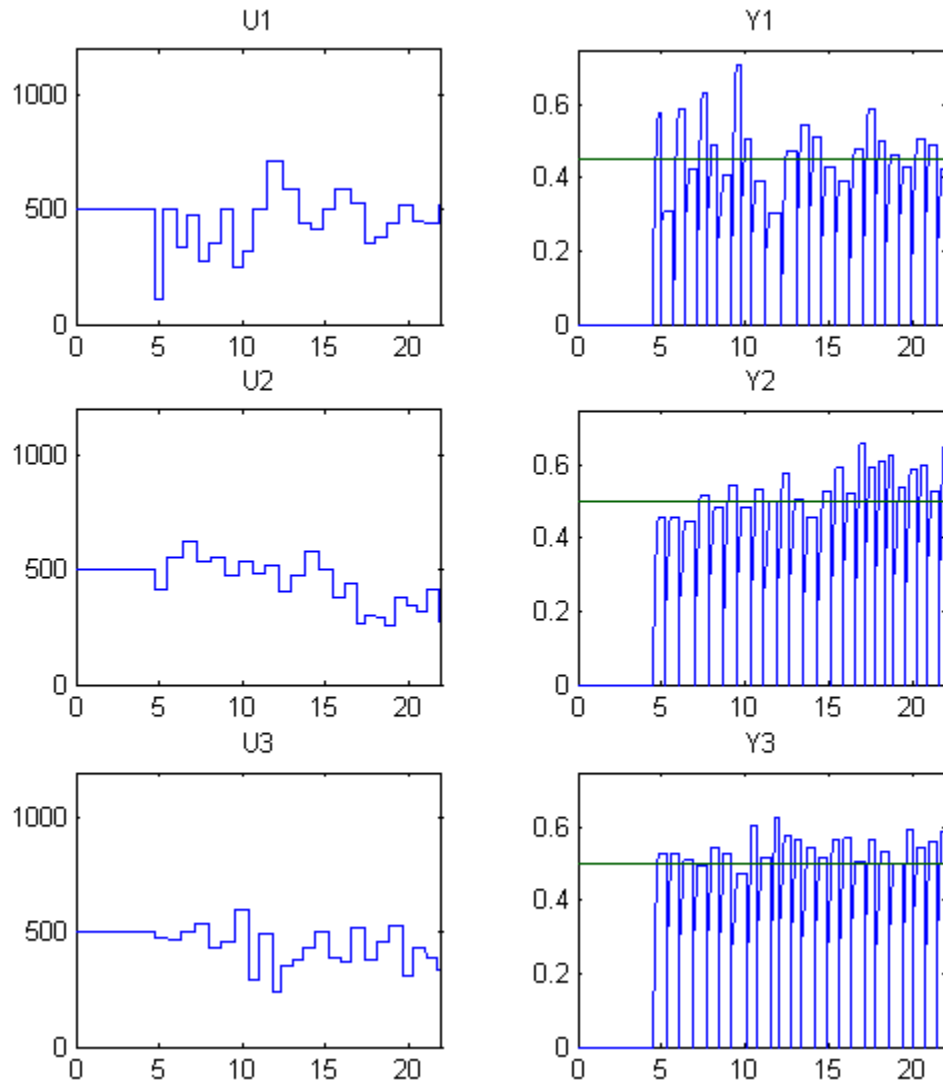


Figure 6-4 First Set of Linear Experiments

The images of the first three experiments are given in Figure 6-5. For these experiments the closed loop output is not filtered and retains the noise in the signal. However, one can see from Figure 6-4 that the control is adjusting to have the output track the setpoint. The linear model structure for this arc welding process is actually only valid for a narrow operating range of peak times and torch speed. The torch speed for the last three experiments was increased and therefore the peak times were much higher to appropriately heat the workpiece. However, the parameter intervals and the linear model assumption of the process become less valid with the higher torch speed. From the results in the last three experiments one can begin to see that the parameter intervals found during system identification are becoming less valid. In the first experiment the closed loop output experiences oscillation because the process has a faster time constant than the controller. The aggressive controllers in experiments 3, 5, and 6 with $W=0$ and $p=2$ or 1 obtain the highest closed loop performance.

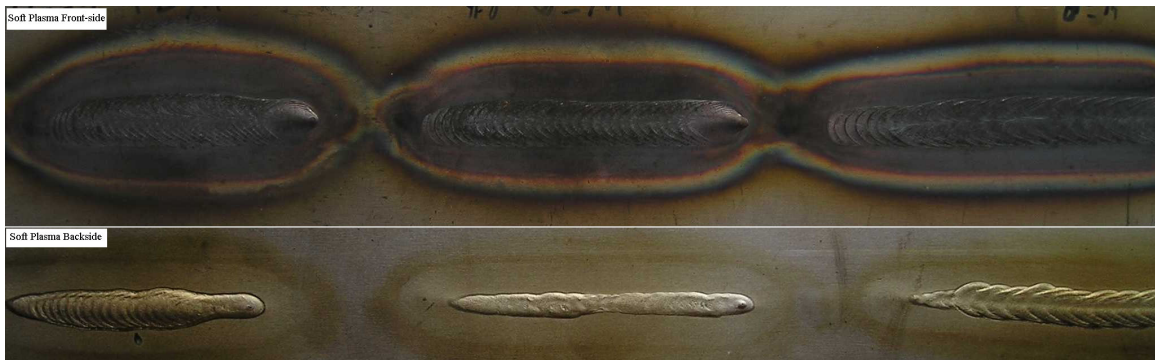


Figure 6-5 Soft Plasma Experiments 1-3

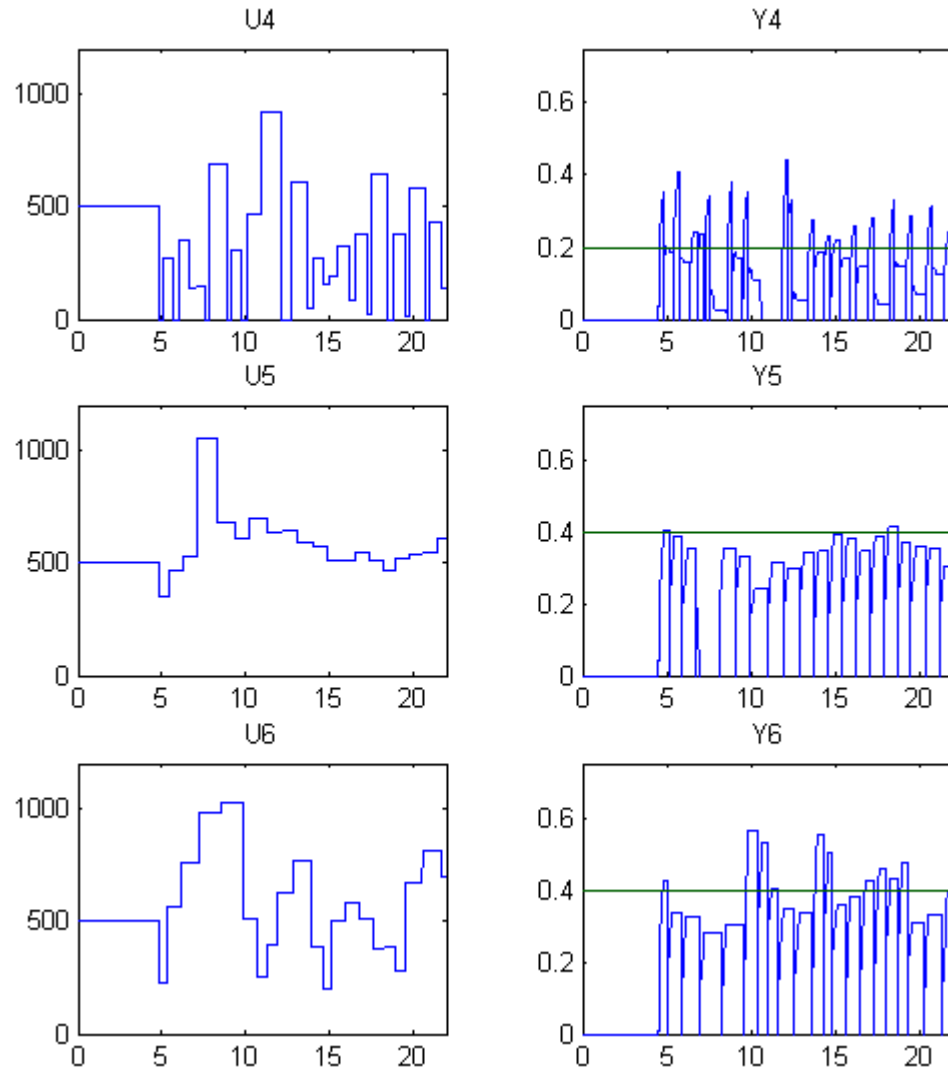


Figure 6-6 Second Set of Linear Experiments

In addition to the linear example above, in the next section an implementation of the proposed algorithm employing the stabilizing control interval was completed for a nonlinear system.

6.4.2 Nonlinear Scalar Example with Implementation

The quasi-keyhole plasma process that was controlled in Section 5.2 was again used for the implementation of the nonlinear control using the stabilizing control interval. The data from the experimental results is given in Figure 6-7,8,9.

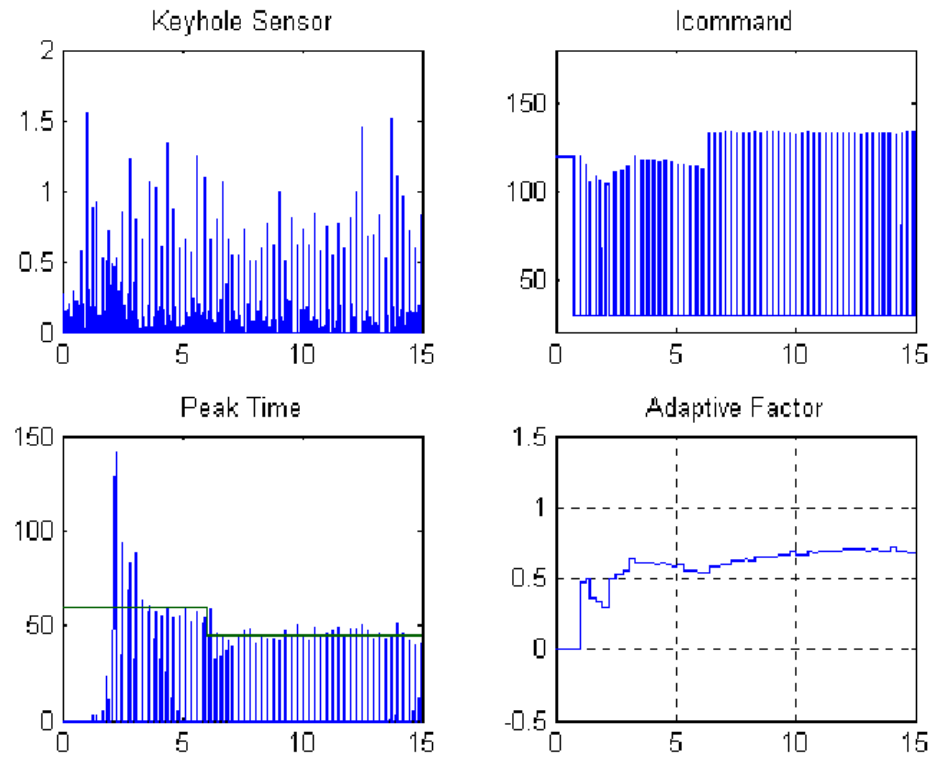


Figure 6-7 Keyhole Plasma Process (Experiment 1)

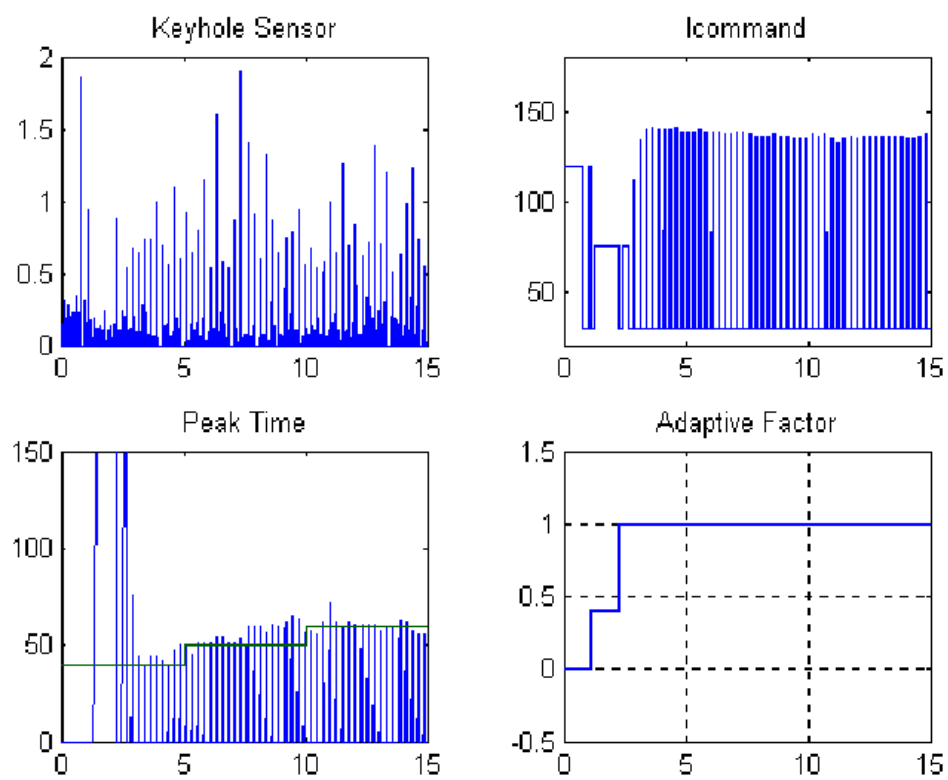


Figure 6-8 Keyhole Plasma Process (Experiment 2)

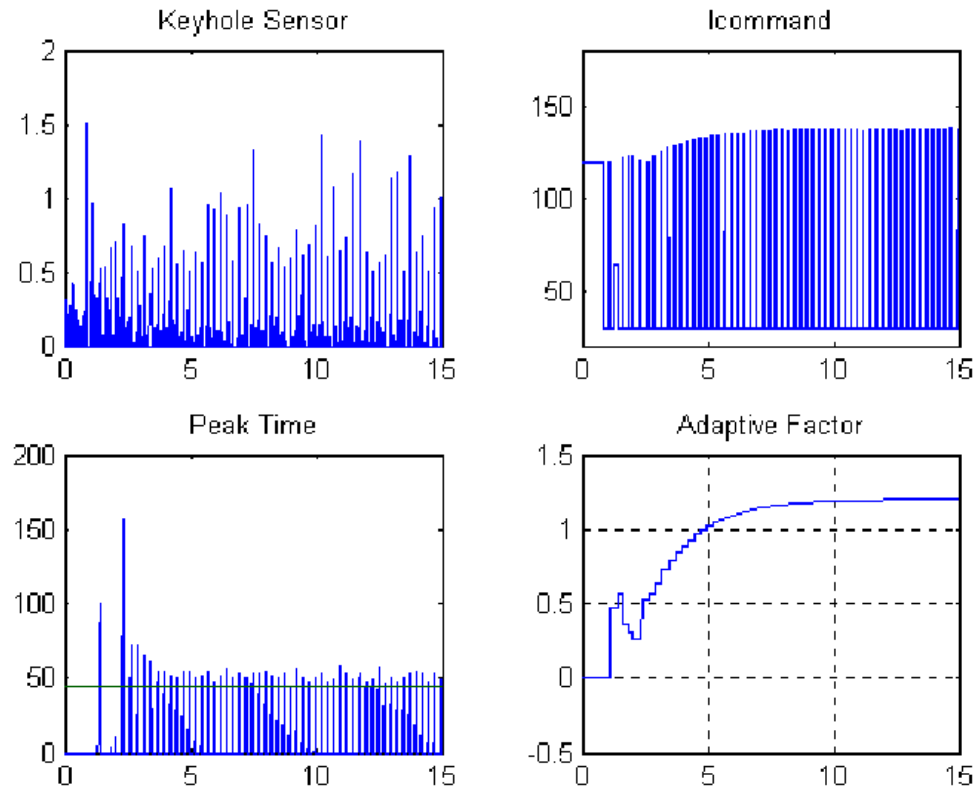


Figure 6-9 Keyhole Plasma Process (Experiment 3)

For the first experiment the parameter intervals that were found from system identification were used in the process control, and the closed loop results achieve what is expected. However, for the second and third set of experiments the parameter intervals were significantly narrowed about each of their mean with the expectation that the process could operate outside of the process intervals. Also, the adaptive factor was constrained to remain between zero and one for experiment 2 and between zero and 1.2 for experiment 3. From the results shown in Figure 6-8 and Figure 6-9, one can see that the process began to operated above the process intervals and because the adaptive factor was saturated, the control could not properly adjust to achieve the desired closed loop reference. An interesting conclusion of these experiments is that movement of the

adaptive factor can tell the control designer something about the accuracy of the parameters intervals. The images each of the welds are given in Figure 6-10.

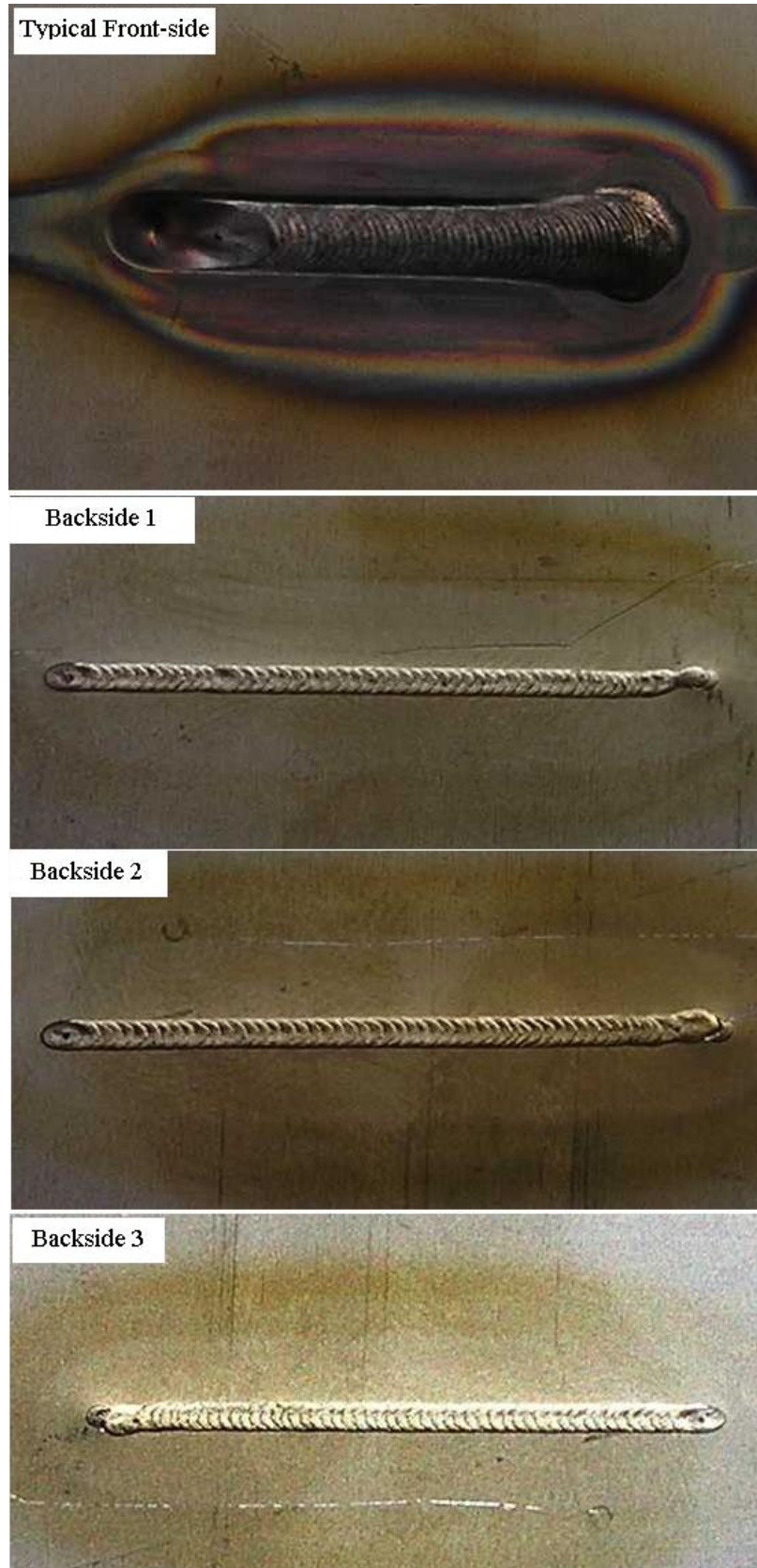


Figure 6-10 Keyhole Plasma Process with New Interval Control

6.5 Multivariable Parameter Interval GMC

The approach for the multivariable case is similar to the scalar development.

Consider a model given by

$$Y_k = \varphi_1 \cdot f_1(x_{1_k}) + \varphi_2 \cdot f_2(x_{2_k}) + \cdots + \varphi_n \cdot f_n(x_{n_k}) + \beta \cdot g(x_k) \cdot U_{k-1}$$

Equation 6-42

where Y_k and $\varphi_i \cdot f_i(x_{ik})$ are nx1 vectors
 $\beta \cdot g(x_k)$ is a nxm matrix
and U_k is a mx1 vector

Because the states $f_i(x_{ik})$ and $g(x_k)$ are known, then if the following intervals are known:

$$\begin{aligned} \varphi_i^{\min} &\leq \varphi_i \leq \varphi_i^{\max} \\ \beta^{\min} &\leq \beta \leq \beta^{\max} \end{aligned}$$

Equation 6-43

then the problem can once again be constructed as an interval problem but with this time the intervals being multivariable.

$$[Y_k] = [H_k] + [B_k][U_k]$$

Equation 6-44

Where $[Y_k]$ is a nx1 interval, $[H_k]$ is a nx1 interval, $[B_k]$ is a nxm, interval, and $[U_k]$ is a mx1 interval.

The solution to the multivariable interval problem is similar in nature to the scalar case, but requires a few extra steps. The reason for the extra steps is that in solving for either U_{min} or U_{max} , the boundary solution that is sought for is actually not the control that causes all the outputs to be equal to their output boundary, but rather the offset from the

generic model control solution that causes at least one output to be set equal to its output boundary. For example, the multivariable solution to:

$$U_{\max} = (B^+)^{-1} (Y_{\max} - H^+)$$

Equation 6-45

solves for the exact control that sets all outputs equal to their maximum allowed.

However, this solution will likely result in some elements of U_{\max} calculated to be much greater than their GMC counterpart and others much less.

Thus, an alternative approach is required which solves the interval inverse problem and also provides useful boundaries for the control variable. The approach proposed uses the following equation:

$$Y = H + B (U_{GMC} + L)$$

Equation 6-46

where L is the control offset from the GMC calculated control which sets at least one element of the output, Y, to its boundary either Y_{\min} for U_{\min} or Y_{\max} for U_{\max} .

Similar to the scalar case, because [Y] or [U] can be either above or below zero, the solution to Equation 6-47 is accomplished in two steps. For example, for determining U_{\max_k} calculate:

$$\begin{aligned} L \max_k 1 &= \min \left((Y_{\max} + H_k^+ - B^+ U_{GMC_k}) ./ R^+ \right) \\ L \max_k 2 &= \min \left((Y_{\max} + H_k^+ - B^- U_{GMC_k}) ./ R^- \right) \end{aligned}$$

Equation 6-47

And then the maximum control boundary is:

$$U_{k\ stable}^+ = \min(U_{GMC_k} + L \max_k 1, U_{GMC_k} + L \max_k 2)$$

Equation 6-48

Where U_{GMC_k} is the control calculated using generic model control, $(./)$ is element-wise division and R^+ and R^- are the gains of B calculated using the equations below:

$$R^- = [B^-] \cdot [1]_{mx1\ vector}$$

$$R^+ = [B^+] \cdot [1]_{mx1\ vector}$$

Equation 6-49

Also for determining U_{\min_k} calculate

$$L \min_k 1 = \max((Y_{\min} + H_k^- - B^- U_{GMC_k}) ./ R^-)$$

$$L \min_k 2 = \max((Y_{\min} + H_k^- - B^+ U_{GMC_k}) ./ R^+)$$

Equation 6-50

And then the minimum control boundary control is given by:

$$U_{k\ stable}^- = \max(U_{GMC_k} + L \min_k 1, U_{GMC_k} + L \min_k 2)$$

Equation 6-51

Finally, then the final control implemented is equal to

$$U_{opt_k} = \max(\min(U_{GMC}, U_{\max}), U_{\min})$$

Equation 6-52

Moreover, similar to the scalar case, predictions can be made for $\hat{U}_{k+1}, \hat{U}_{k+2}, \dots, \hat{U}_{k+p}$ and

then averaged giving U_k with the restriction that

$$U_k^- < \tilde{U}_k < U_k^+$$

Equation 6-53

In addition, also similar to the scalar case, during the control predictions, the modeling of the states can be updated using an adaptive factor based upon the sampled output and maximum and minimum predictions from the previous sampling time.

Therefore, the control is calculated by using the following steps:

1. Calculate F_k from output sample and $Y_{pred\ k}^{\min}$, $Y_{pred\ k}^{\max}$ from the previous iteration.
2. Calculate $U_{GMC} = (\bar{B})^{-1} \cdot (Y_{ref\ k} - \bar{H})$
3. Determine the possible offsets that set the control boundary Lmin1, Lmin2, Lmax1, Lmax2
4. Calculate $[U_k]_{stable}$ using Equations 6-49-53
5. Calculate $U_{opt\ k}$ using Equation 6-54
6. In a for loop predict $U_{opt\ k+1} \dots U_{opt\ k+p}$ using the adaptive factor, F_k , and control parameter ΔF_k to update states.
7. Then implement $\tilde{U}_k = \sum_{i=k}^{k+p} W_i \cdot U_i$ with $U_k \in [U_k]_{stable}$
8. Predict $Y_{pred\ k+1}^{\min}$, $Y_{pred\ k+1}^{\max}$ using \tilde{U}_k and the maximum and minimum models

6.5.1 Linear Multivariable Case

Given a discrete, observable linear system with parameter intervals for A and B only (i.e. not C) so that the states x_k can be solved for:

$$\begin{aligned} x_{k+1} &= A_{sys} x_k + B_{sys} U_k \\ Y_k &= C_{sys} x_k \end{aligned}$$

Equation 6-54

Then the output feedback system can be reorganized to

$$Y_{k+1} = C_{sys} A_{sys} x_k + C_{sys} B_{sys} U_k$$

Equation 6-55

So that in relation to the interval problem:

$$\begin{aligned} H_k^- &= C_{sys} A_{sys}^- x_k, & H_k^+ &= C_{sys} A_{sys}^+ x_k \\ B^- &= C_{sys} B_{sys}^-, & B^+ &= C_{sys} B_{sys}^+ \\ R^- &= C_{sys} B_{sys}^- \cdot [1]_{(mx1 \text{ vector})}, & R^+ &= C_{sys} B_{sys}^+ \cdot [1]_{(mx1 \text{ vector})} \end{aligned}$$

Equation 6-56

See the example in the following section.

6.5.2 Linear Multivariable Simulation

For the coupled (2 input – 2 output) continuous linear system given by:

$$A = \begin{bmatrix} -.5 & 0 & 0 & 0 \\ 0 & -.1 & 0 & 0 \\ 0 & 0 & -.33 & 0 \\ 0 & 0 & 0 & -.083 \end{bmatrix} \quad B = \begin{bmatrix} .5 & 0 \\ .125 & 0 \\ 0 & .125 \\ 0 & .25 \end{bmatrix} \quad C = \begin{bmatrix} .5 & 0 & .16 & 0 \\ 0 & .064 & 0 & .133 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Equation 6-57

where each parameter of A & B of the system varies by $\pm 50\%$ resulting in the models with the following minimum and maximum model step responses in Figures 6-11,12. The system in Equation 6-57 can be discretized and then Equations 6-54,55, and 56 can be used with the PGMC control methodology to form a closed loop system.

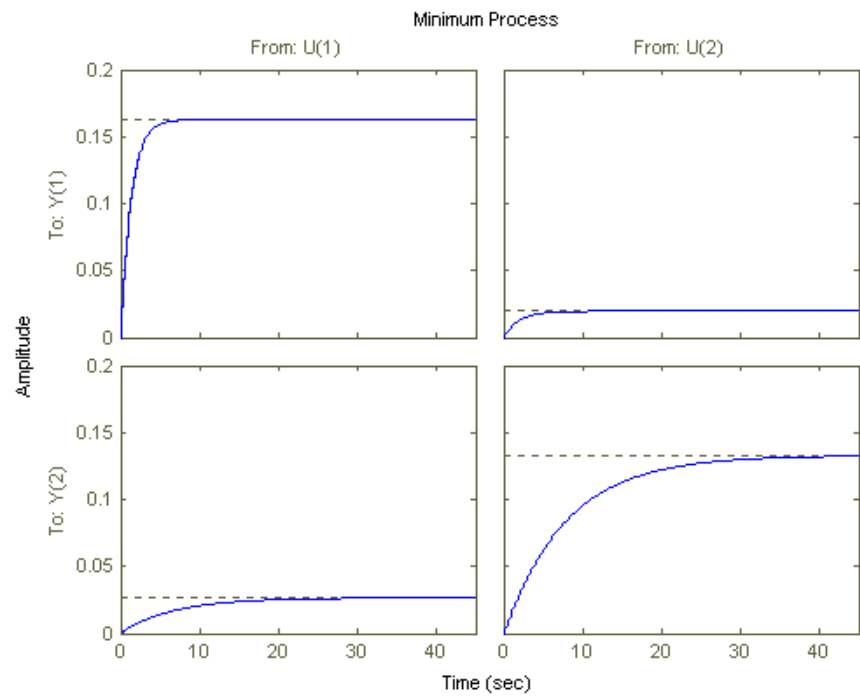


Figure 6-11 Minimum Open Loop Step Response

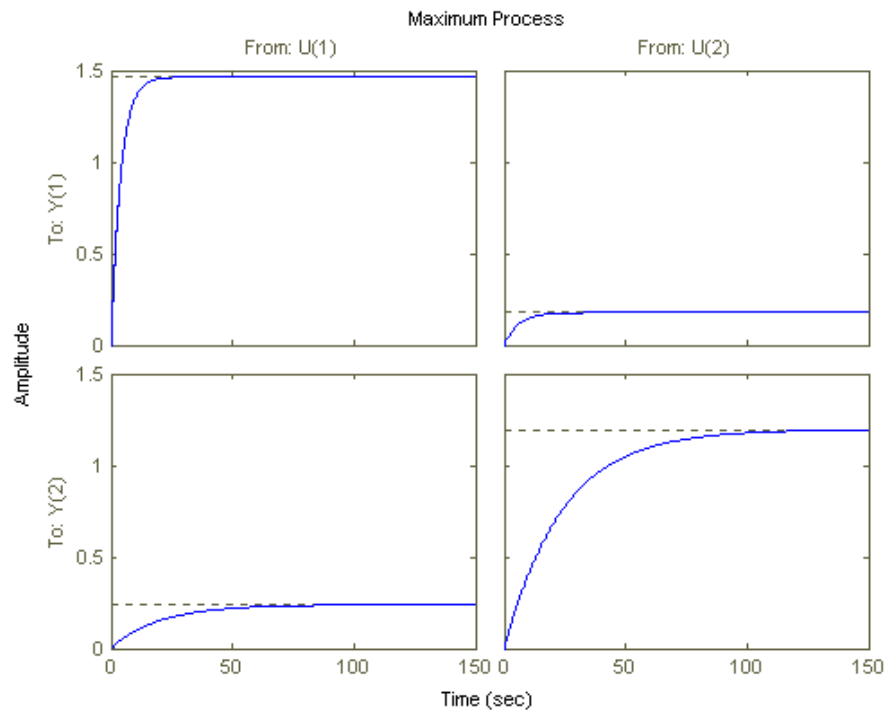


Figure 6-12 Maximum Open Loop Step Response

If the maximum and minimum discrete model that can be formed from the $\pm 50\%$ variation of the parameters of the continuous model in Equation 6-57 are used in the closed loop simulation, the actual process could be simulated as varying between the maximum and minimum according to:

$$Y_{Actual} = Y_{min} + F \cdot (Y_{max} - Y_{min})$$

Equation 6-58

where because there are two outputs, then F is composed of two factors, F_1 and F_2 . If the adaptive factors change smoothly or are approximately constant, the adaptation of the PGMC control performs optimally in comparison with the regular GMC method. If the change in F_1 and F_2 is prescribed as is given in Figure 6-13, then the resulting closed loop GMC control is given in Figures 6-14,15 and the resulting Interval PGMC control is given in Figures 6-16,17.

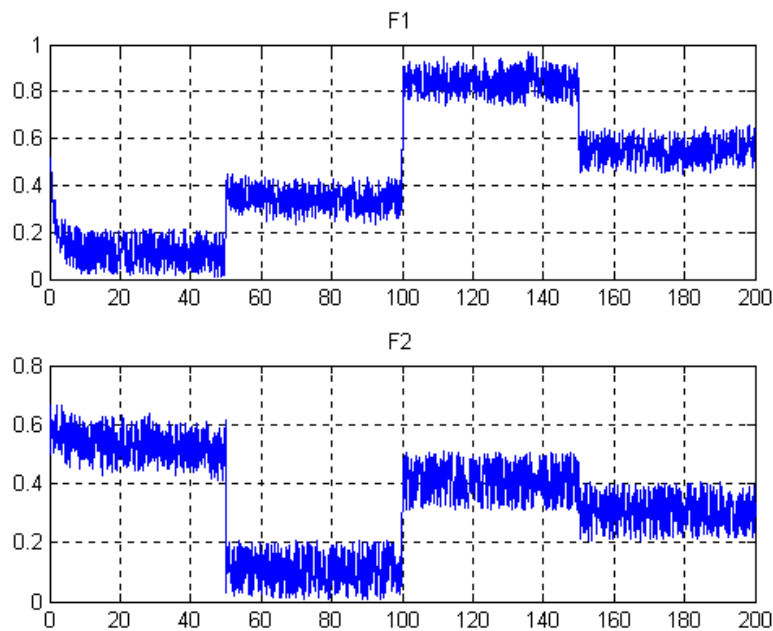


Figure 6-13 Case 1: Change of F thru Time

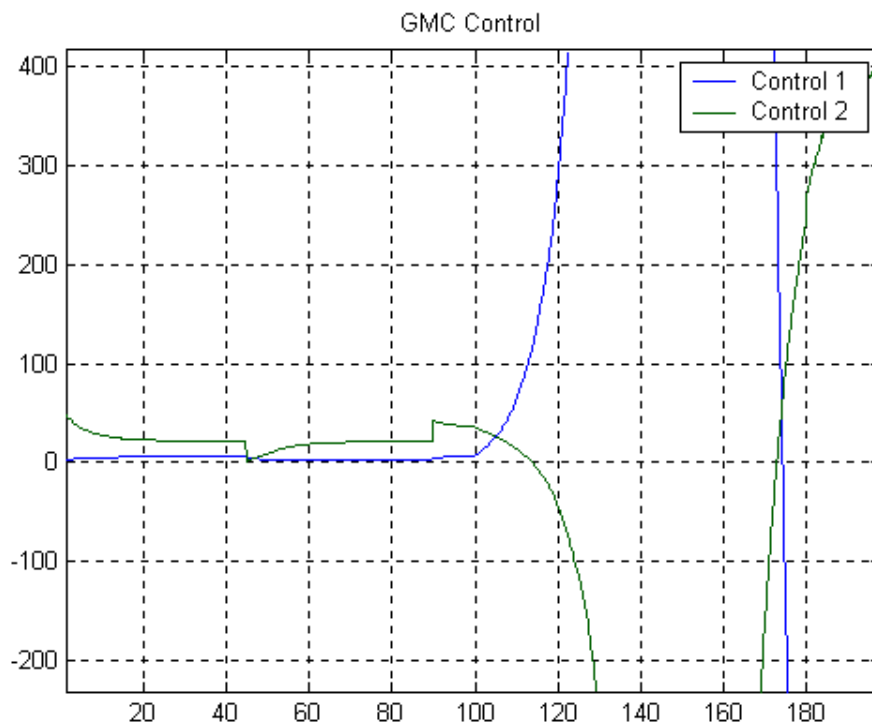


Figure 6-14 Case 1: GMC Control

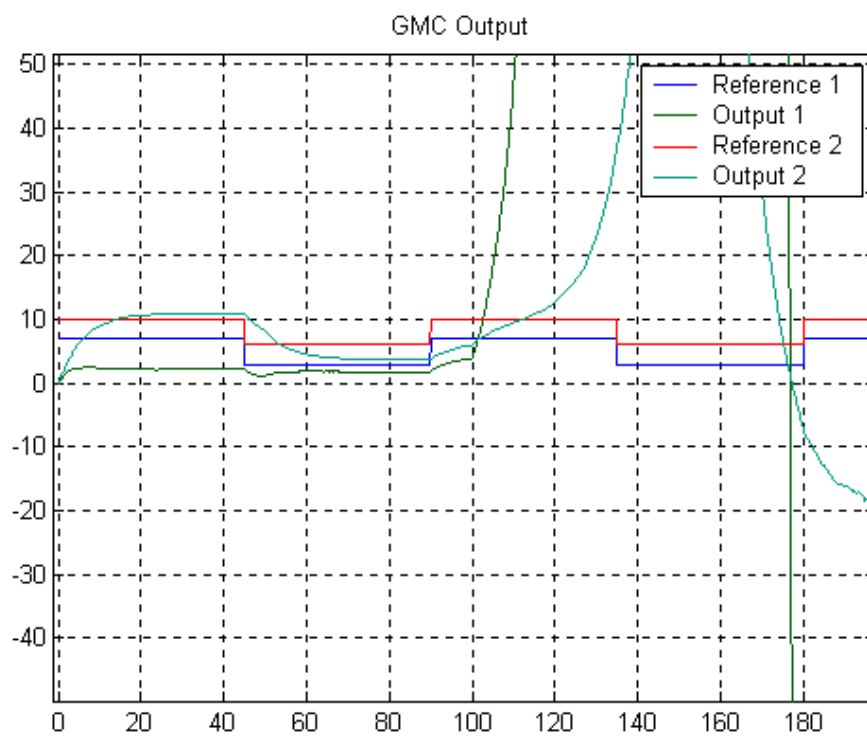


Figure 6-15 Case 1: GMC Output

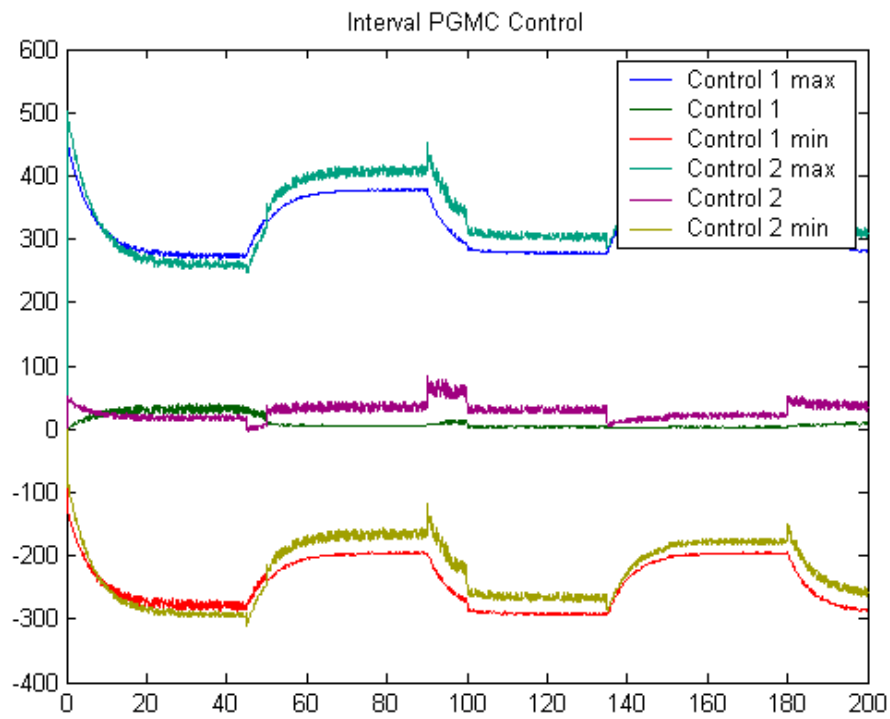


Figure 6-16 Case 1: Interval PGMC Control

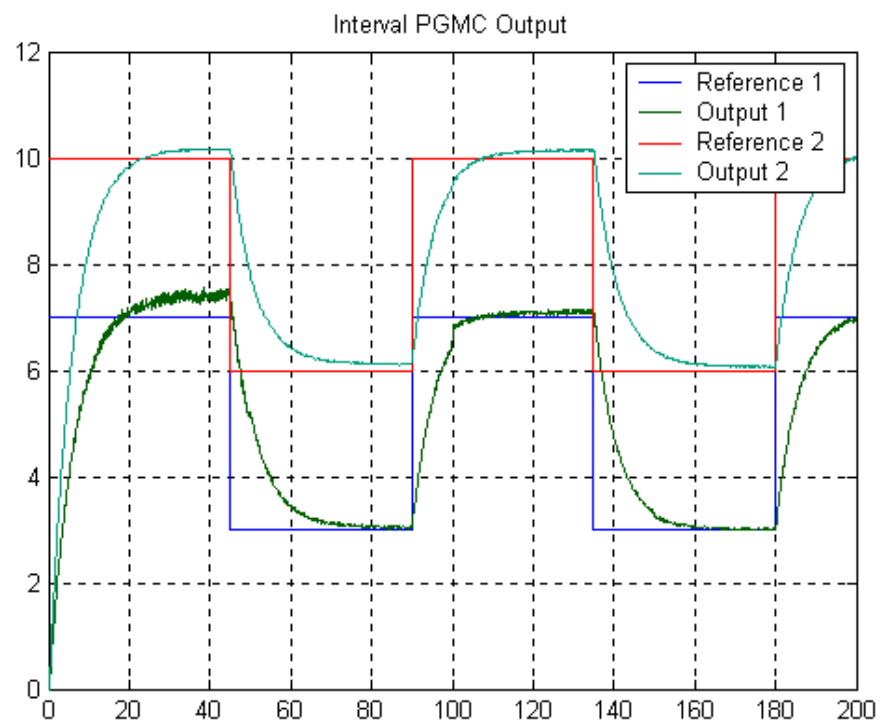


Figure 6-17 Case 1: Interval PGMC Output

From the previous figures, it can be seen that the parameter interval PGMC performance is far superior to the regular GMC method. However, this was under ideal circumstances in that the simulated process varied according Equation 6-58 which is essentially the same equation that is used in the adaptation part of the control algorithm. Thus, the adaptation part of the control algorithm could identify the actual process model. If, however, the actual process varied between its maximum and minimum models such that each individual parameter of Equation 6-57 varies uniformly and randomly between its maximum and minimum, then this would be the worst case scenario. For the process model does not vary smoothly but completely randomly between sampling times from its maximum to minimum model, and thus the adaptive part of the control algorithm does not improve the control. In fact, if the oscillation is about the mean of the parameters, the normal GMC algorithm can perform better than newly designed algorithm. In this case, it would behoove the control designer to restrict the change of F between sampling times by the use of a filter or by directly filtering the output signal. Moreover, the prediction level p should be small if not equal to 1 because valid predictions can only be made if the change in the model between sampling times is smooth.

However, to demonstrate this worst case scenario the following simulation is given for the system where each individual parameter of Equation 6-57 varies uniformly and randomly between its maximum and minimum. Figure 6-18,19 show the closed loop behavior for GMC and Figures 6-20,21 show the closed loop behavior of the Interval PGMC method.

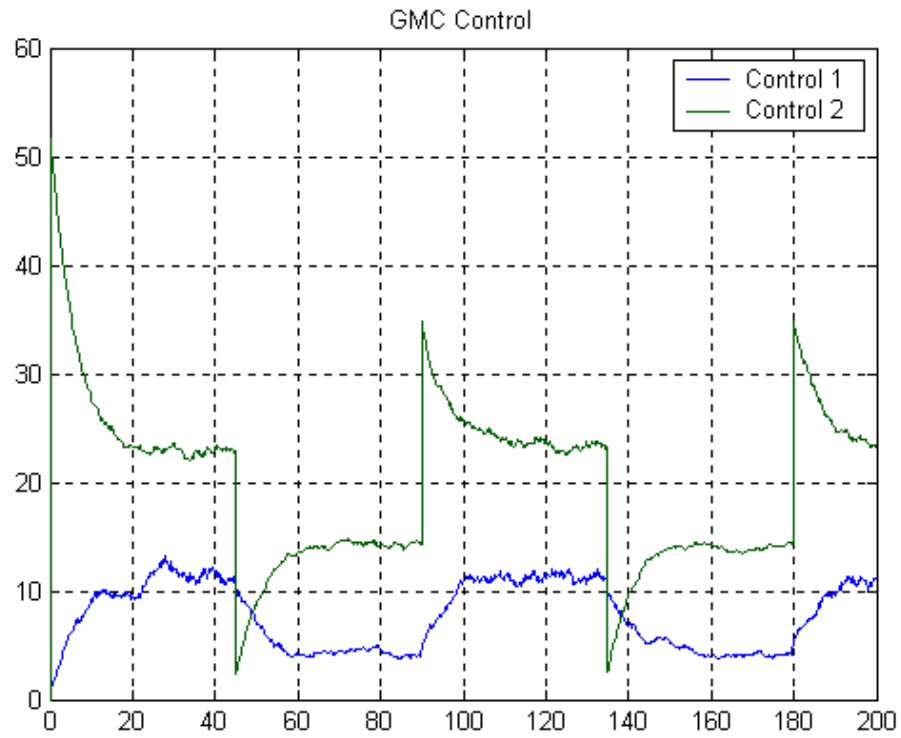


Figure 6-18 Case 2: GMC Control

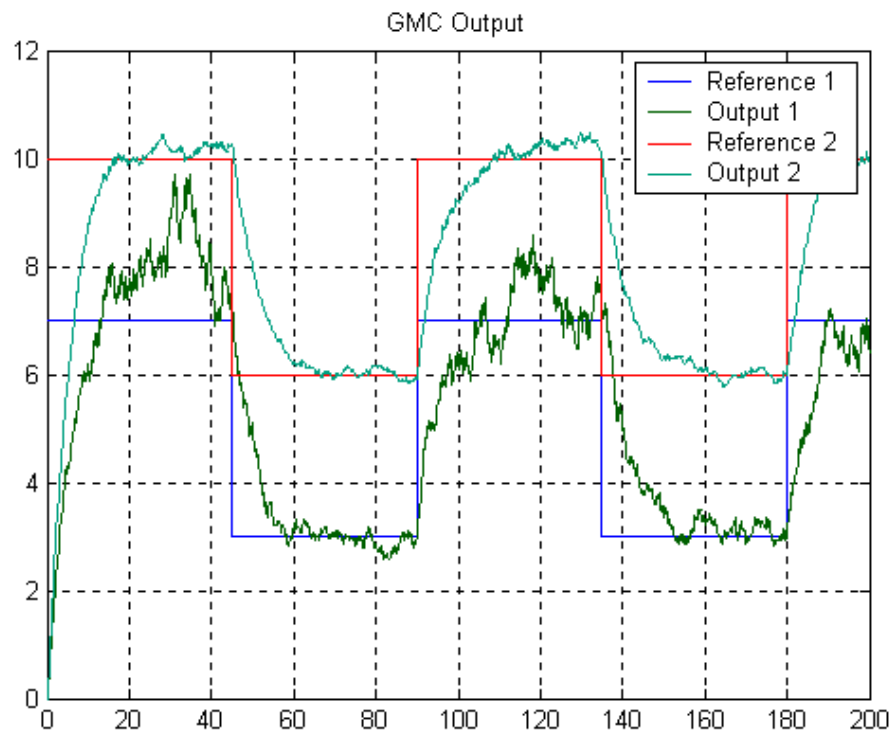


Figure 6-19 Case 2: GMC Output

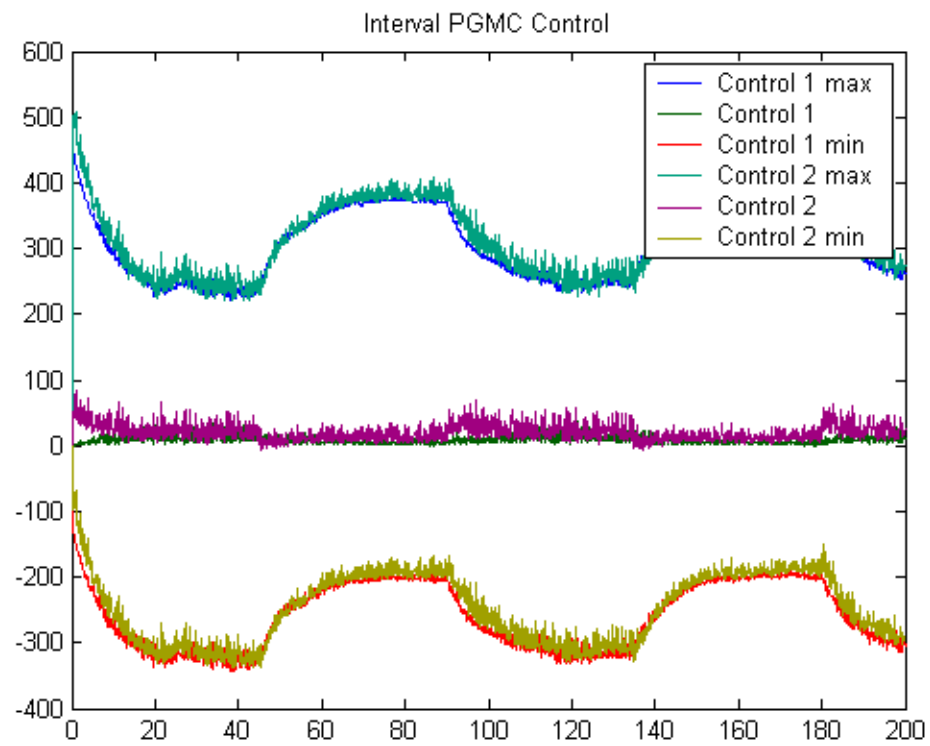


Figure 6-20 Case 2: Interval PGMC Control

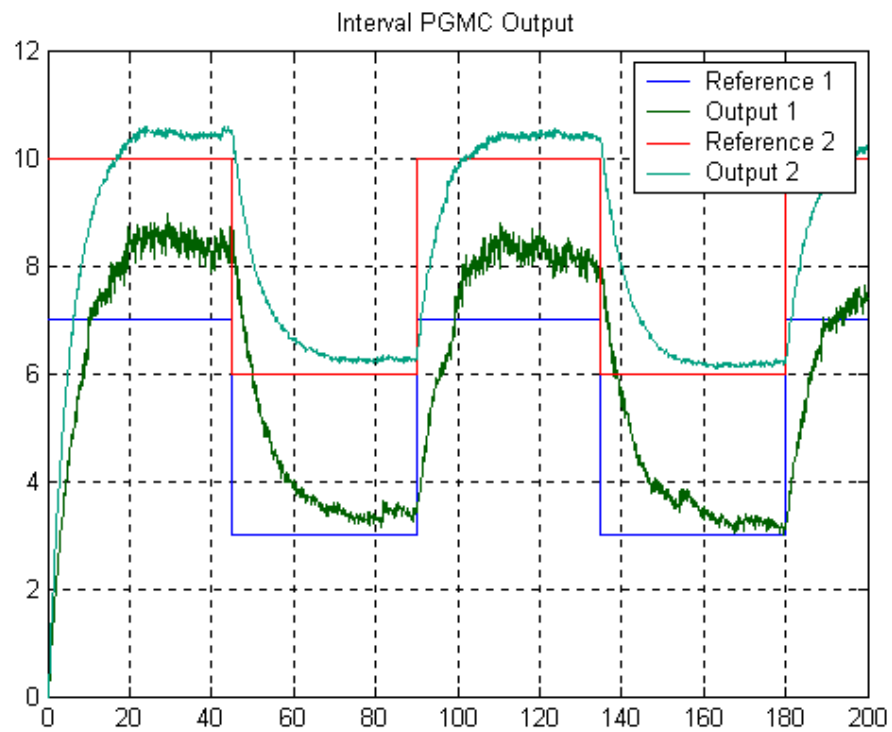


Figure 6-21 Case 2: Interval PGMC Output

From the previous Figures 6-18 thru 6-21 it can be seen that if the parameters change completely uniformly and randomly between their maximum and minimum, and there is no filter on either the adaptive factor or the sampled output, then GMC algorithm can actually perform better than the newly designed algorithm.

Now the most likely scenario for a real process is that each of the parameters varies smoothly between its maximum and minimum between sampling times and that some vary more smoothly than others do. Thus, this next simulation will make the real process equal to the process given in Equation 6-57 where each of the eight parameters varies between its maximum and minimum according to its particular plot given in Figure 6-22. This produces the GMC closed loop results in Figure 6-23,24 and Interval PGMC closed loop results in Figure 6-25, 26.

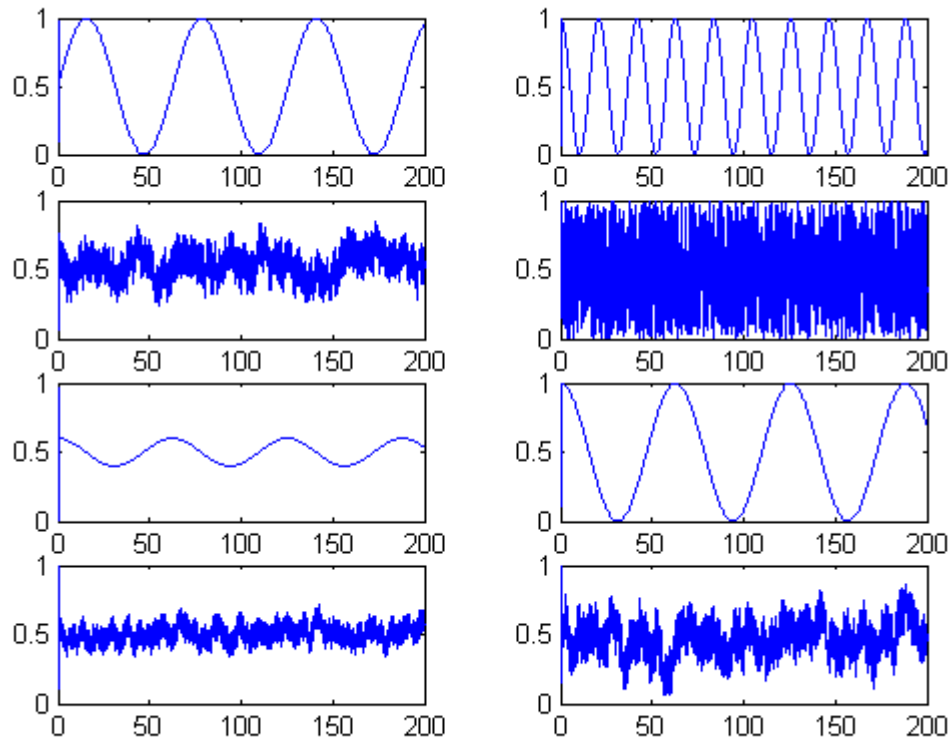


Figure 6-22 Case 3: Change of Parameters thru Time

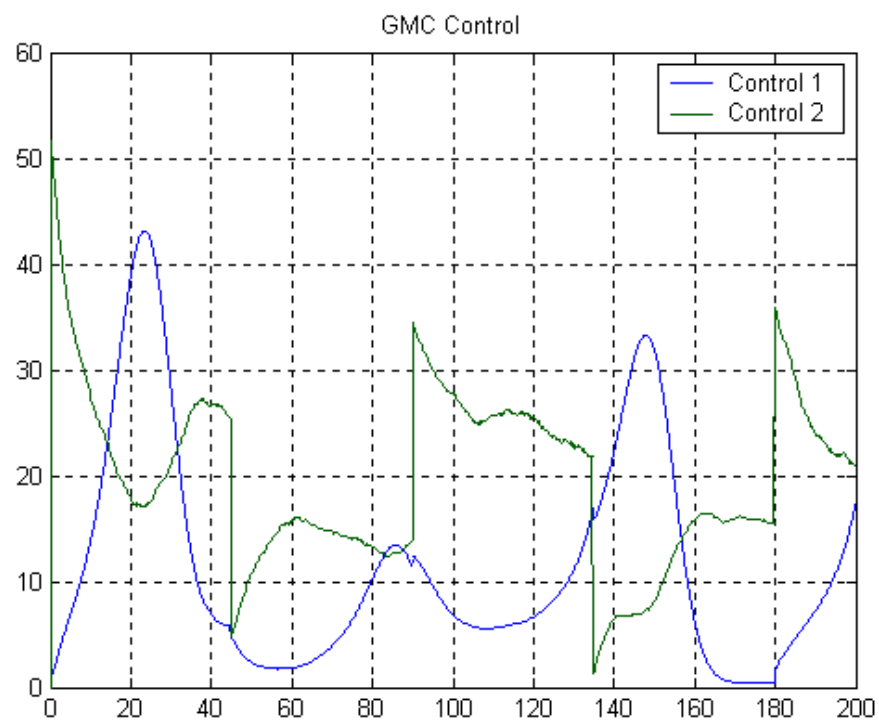


Figure 6-23 Case 3: GMC Control

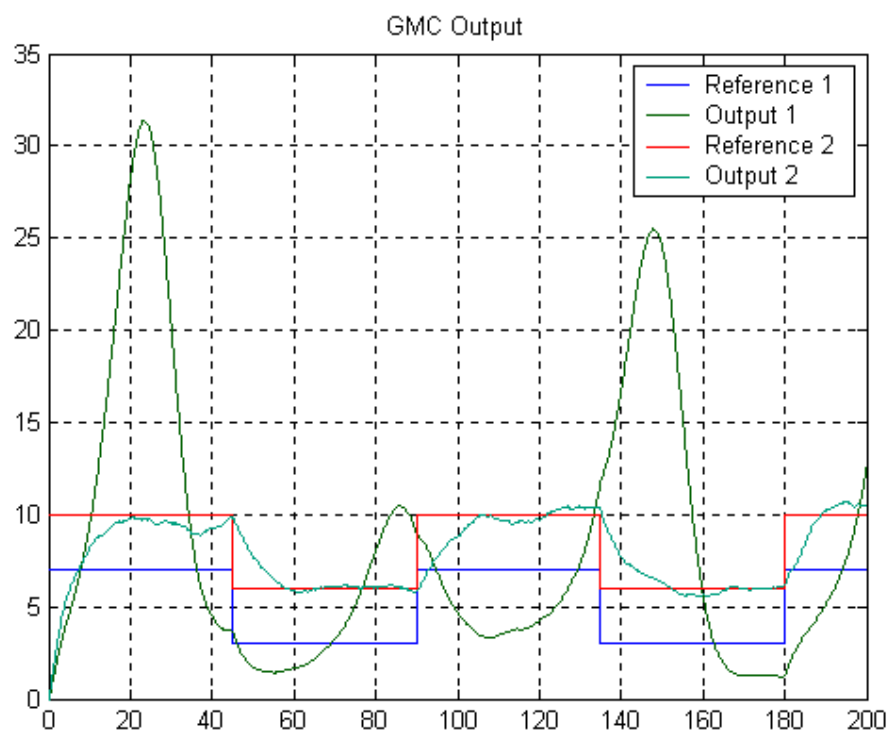


Figure 6-24 Case 3: GMC Output

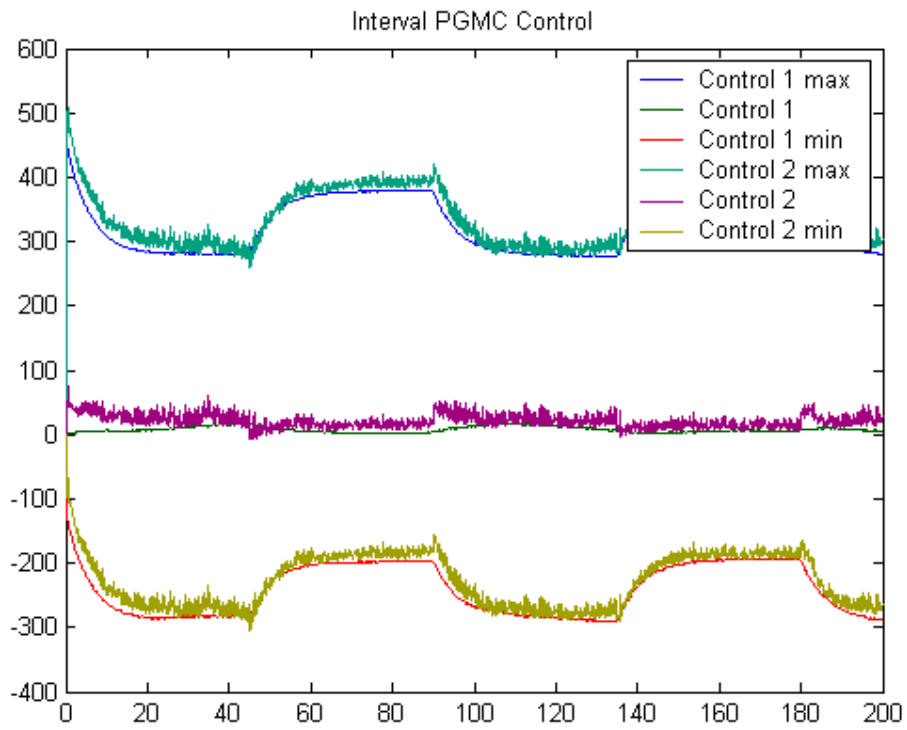


Figure 6-25 Case 3: Interval PGMC Control

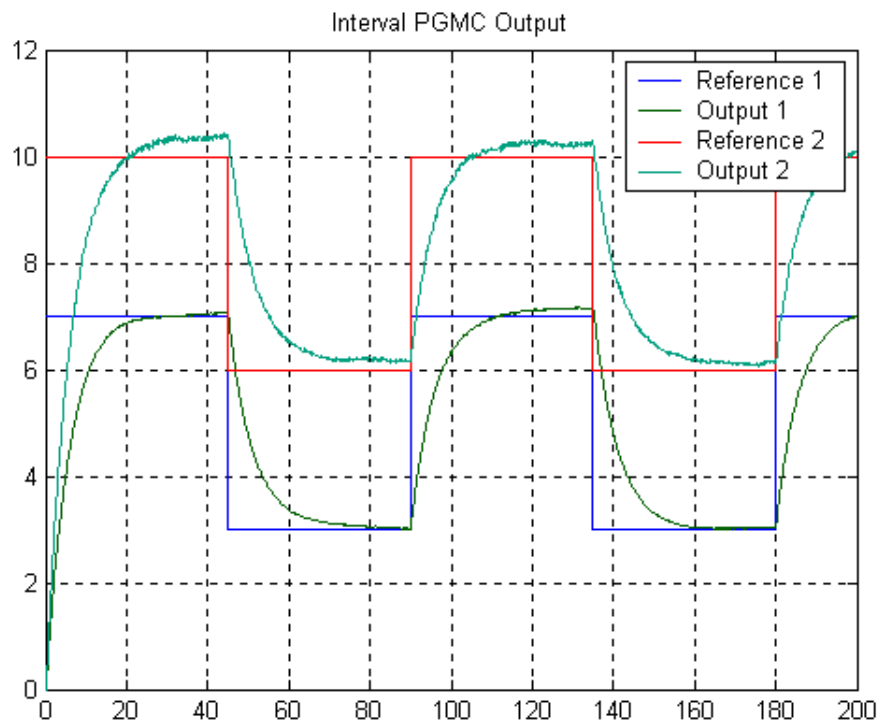


Figure 6-26 Case 3: Interval PGMC Output

From the previous Figures 6-23 thru 6-26, it can be seen that the Interval PGMC's performance is vastly superior to the original GMC method due to the adaptive factor and control smoothing. One additional simulation will be done to show the stability performance of Interval PGMC using the same process as in the last case where each of the model parameters changes smoothly, but with a restriction on the outputs given as:

$$Y1_{\max} = 7 \text{ and } Y2_{\max} = 10$$

$$Y1_{\min} = -5 \text{ and } Y2_{\min} = -5.$$

The closed loop performance with the tighter stability restrictions is given in Figures 6-27,28. The first thing to note is that output restrictions are achieved, which is expected. However, because the inputs and outputs of the model are coupled, the control that is found to satisfy the stability requirements, forces the output Y2 to swing drastically. This is not desirable. Moreover, upon inspection there are possibilities of a more optimal control that could have been implemented. For example, instead of having Y1 be set to its maximum output boundary and then forcing Y2 to swing drastically, a more appropriate control should be implemented that would have set both Y1 and Y2 less than their output boundary and forcing only a minimum swing in each variable. Unfortunately, the method by which this could be done has not been developed.

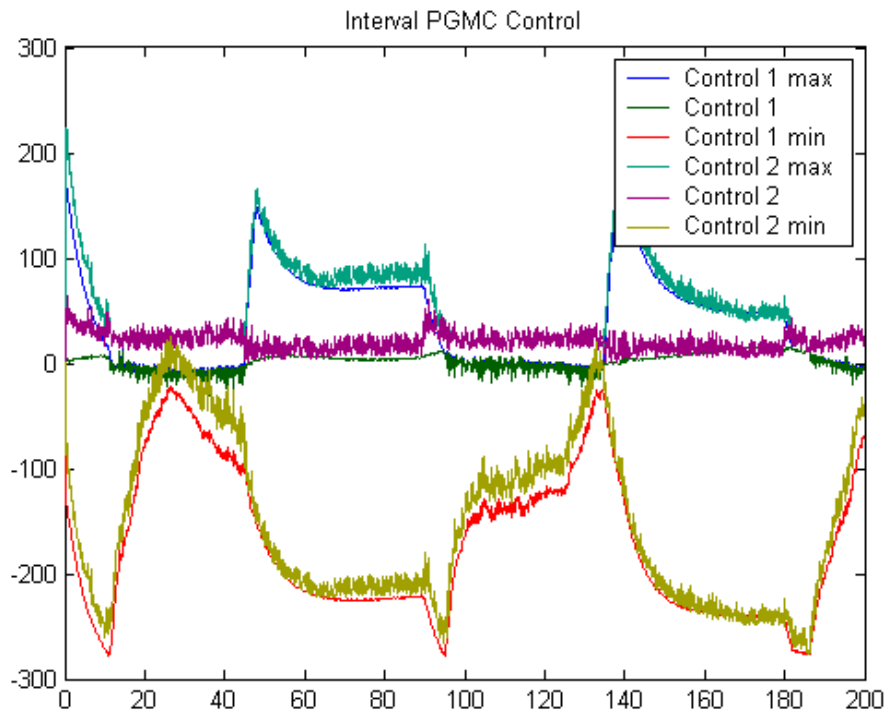


Figure 6-27 Case 4: Interval PGMC Control

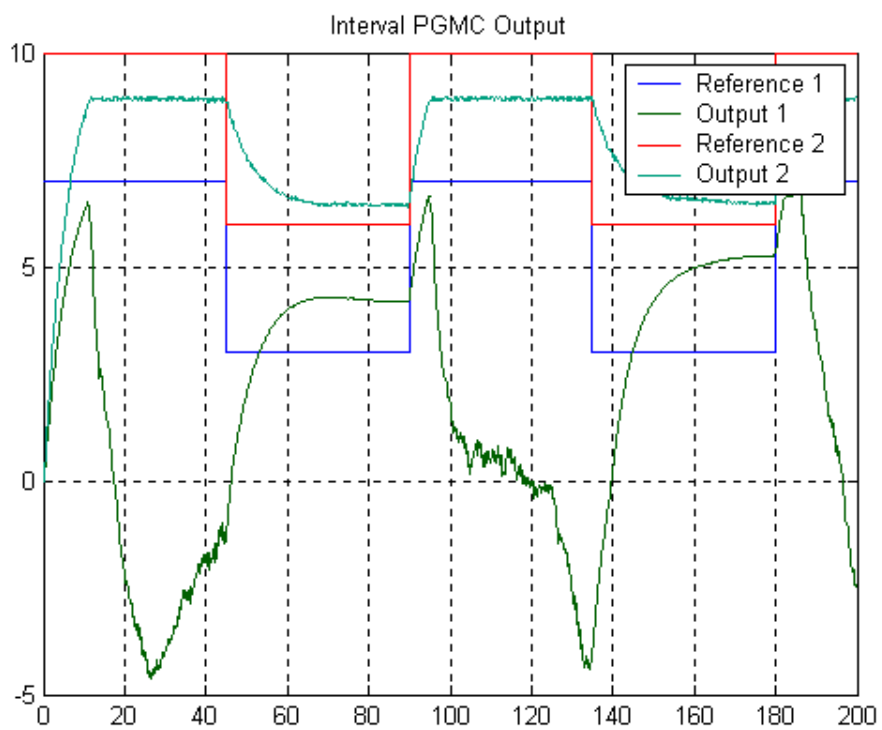


Figure 6-28 Case 4: Interval PGMC Output

6.5.3 Nonlinear Multivariable Simulation

In addition to a linear multivariable system, a nonlinear multivariable example can demonstrate the utility of the proposed algorithm. Consider the following nonlinear multivariable system:

$$\begin{aligned} Y1_k &= a_0 + a_1(Y1_{k-1})^2 \sin(Y1_{k-1}) + a_2 Y1_{k-2} Y2_{k-2} + a_3 U1_k + a_4 U2_k \\ Y2_k &= b_0 + b_1(Y2_{k-1})^2 \sin(Y2_{k-1}) + b_2 Y1_{k-2} Y2_{k-2} + b_3 U1_k + b_4 U2_k \end{aligned}$$

Equation 6-59

with parameter intervals given by:

$$\left\{ \begin{array}{ccc} 100 & \leq a_0 \leq & 120 \\ -.00006 & \leq a_1 \leq & -.000001 \\ .0002 & \leq a_2 \leq & .00035 \\ 5 & \leq a_3 \leq & 6 \\ .1 & \leq a_4 \leq & .2 \end{array} \right\}, \left\{ \begin{array}{ccc} 85 & \leq b_0 \leq & 95 \\ -.00009 & \leq b_1 \leq & 0 \\ .0001 & \leq b_2 \leq & .0004 \\ .1 & \leq b_3 \leq & .2 \\ 4 & \leq b_4 \leq & 5 \end{array} \right\}$$

Equation 6-60

Then the multivariable system can be reduced to the interval problem by:

$$\begin{aligned} H_k^- &= \left[\begin{array}{l} \min(a_0^-, a_0^+) + \min(a_1^-(Y1_{k-1})^2 \sin(Y1_{k-1}), a_1^+(Y1_{k-1})^2 \sin(Y1_{k-1})) + \min(a_2^- Y1_{k-2} Y2_{k-2}, a_2^+ Y1_{k-2} Y2_{k-2}) \\ \min(b_0^-, b_0^+) + \min(b_1^-(Y2_{k-1})^2 \sin(Y2_{k-1}), b_1^+(Y2_{k-1})^2 \sin(Y2_{k-1})) + \min(b_2^- Y1_{k-2} Y2_{k-2}, b_2^+ Y1_{k-2} Y2_{k-2}) \end{array} \right] \\ H_k^+ &= \left[\begin{array}{l} \max(a_0^-, a_0^+) + \max(a_1^-(Y2_{k-1})^2 \sin(Y2_{k-1}), a_1^+(Y2_{k-1})^2 \sin(Y2_{k-1})) + \max(a_2^- Y1_{k-2} Y2_{k-2}, a_2^+ Y1_{k-2} Y2_{k-2}) \\ \max(b_0^-, b_0^+) + \max(b_1^-(Y2_{k-1})^2 \sin(Y2_{k-1}), b_1^+(Y2_{k-1})^2 \sin(Y2_{k-1})) + \max(b_2^- Y1_{k-2} Y2_{k-2}, b_2^+ Y1_{k-2} Y2_{k-2}) \end{array} \right] \end{aligned}$$

$$B^- = \begin{bmatrix} a_3^- & a_4^- \\ b_3^- & b_4^- \end{bmatrix}, \quad B^+ = \begin{bmatrix} a_3^+ & a_4^+ \\ b_3^+ & b_4^+ \end{bmatrix}$$

$$R^- = B^- \cdot [1]_{(mx1 \text{ vector})}, \quad R^+ = B^+ \cdot [1]_{(mx1 \text{ vector})}$$

Equation 6-61

A step response for the system given in Equation 6-59,60 is given in Figure 6-29.

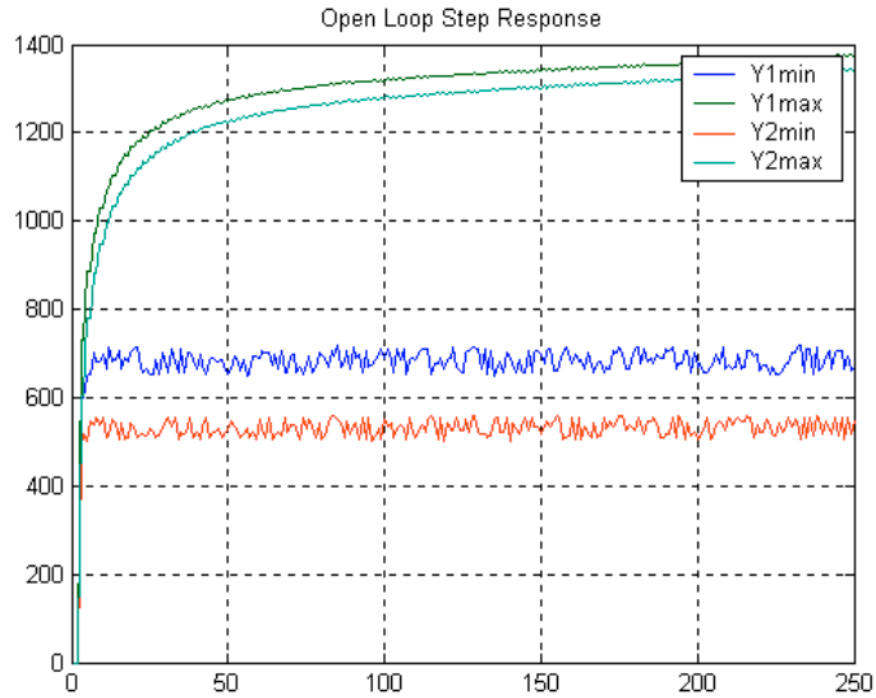


Figure 6-29 Open Loop Step Response

The closed loop system output and control using the normal GMC algorithm is given in Figure 6-30, 31. The closed loop system output and control using the interval algorithm is given in Figures 6-32, 33. In comparing these Figures one can again see the utility of the interval algorithm in coping with the parametric uncertainties for even nonlinear systems. Lastly, in Figures 6-34, 35, the closed loop results are shown for the interval algorithm used in conjunction with a Y_{\max} restriction equal to 1000 for both outputs.

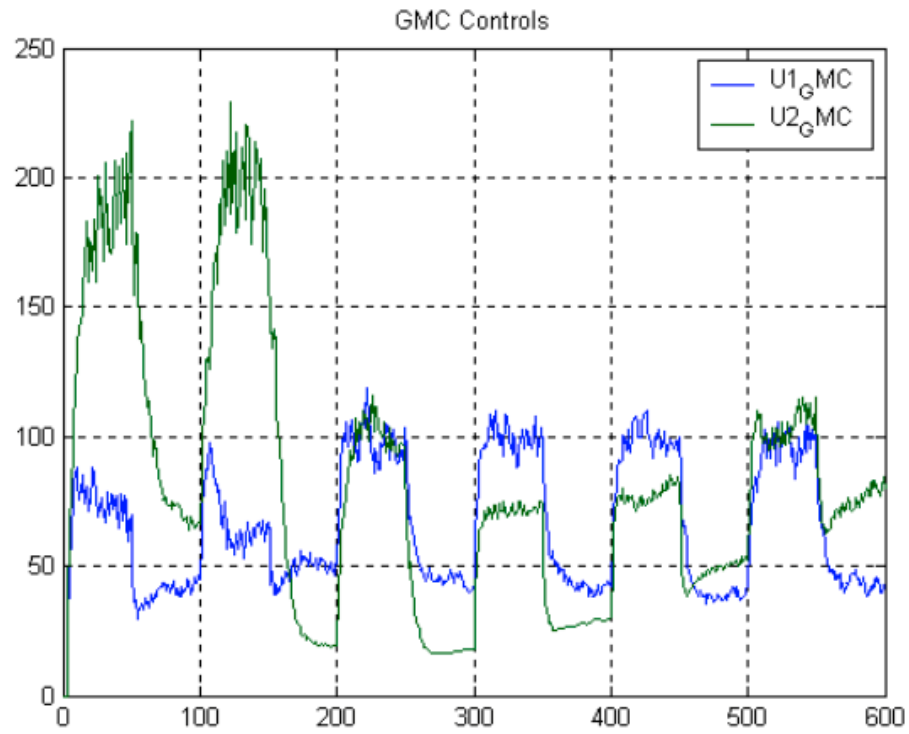


Figure 6-30 GMC Control of Nonlinear Multivariable System

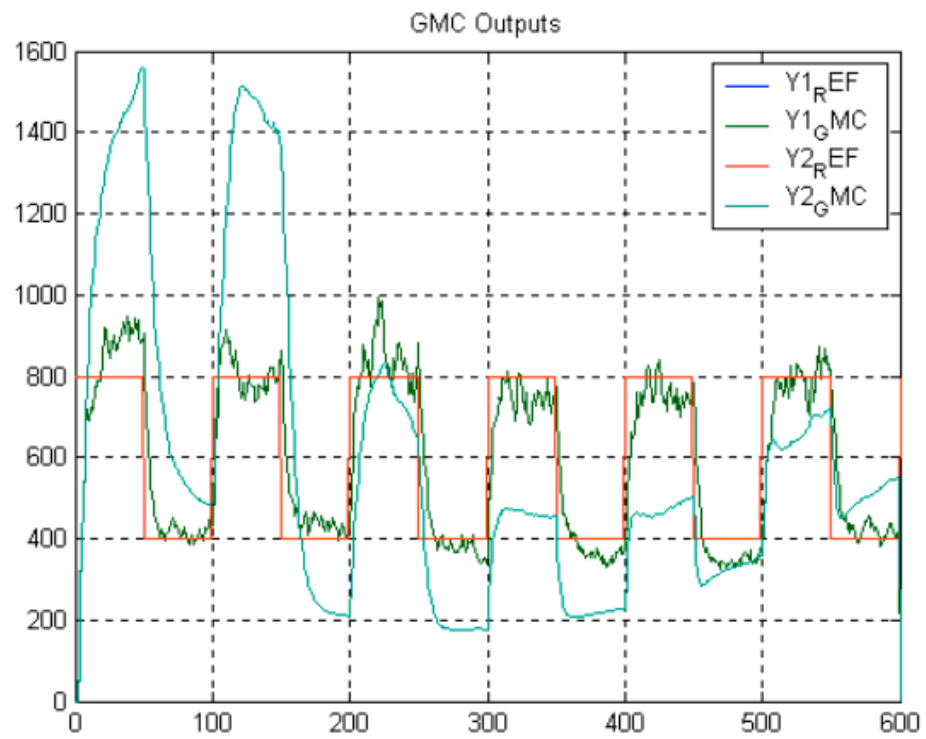


Figure 6-31 GMC Output of Nonlinear Multivariable System

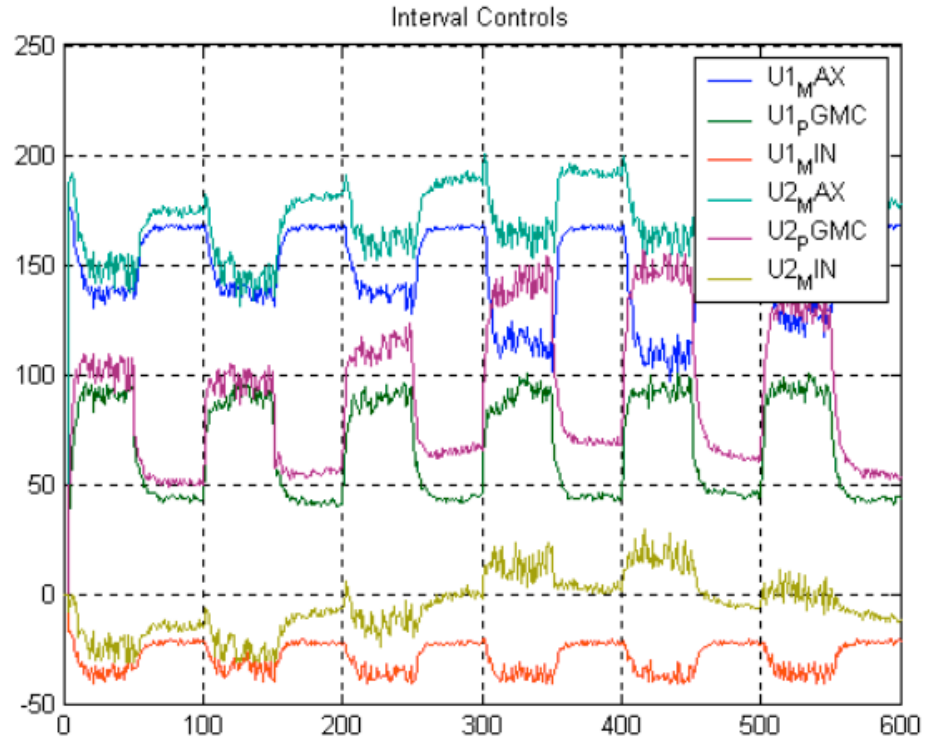


Figure 6-32 Interval Control of Nonlinear Multivariable System

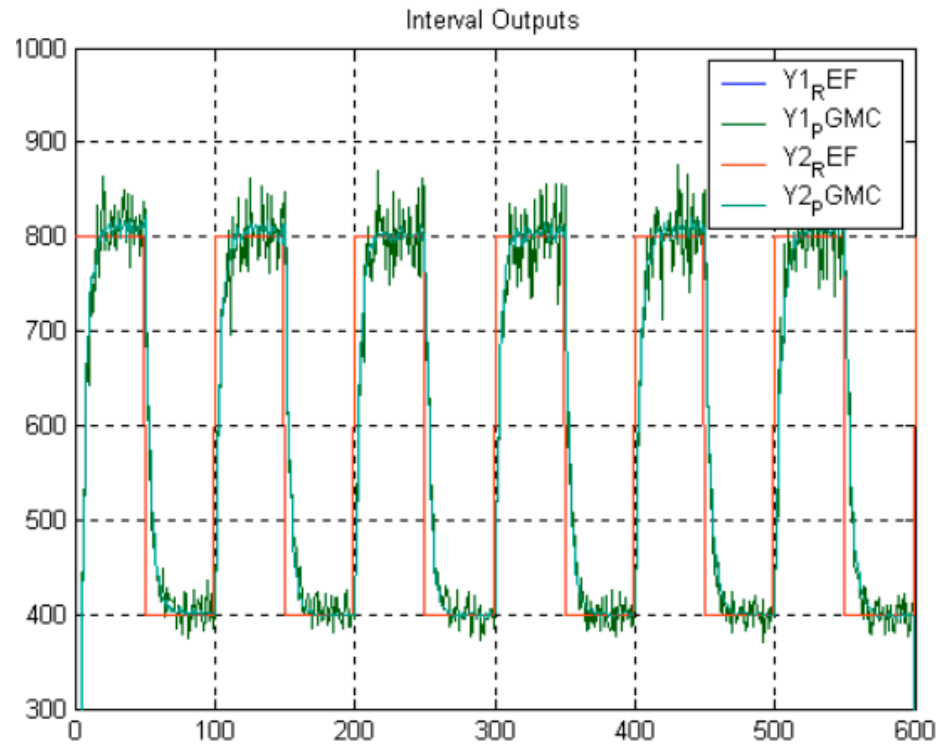


Figure 6-33 Interval Output of Nonlinear Multivariable System

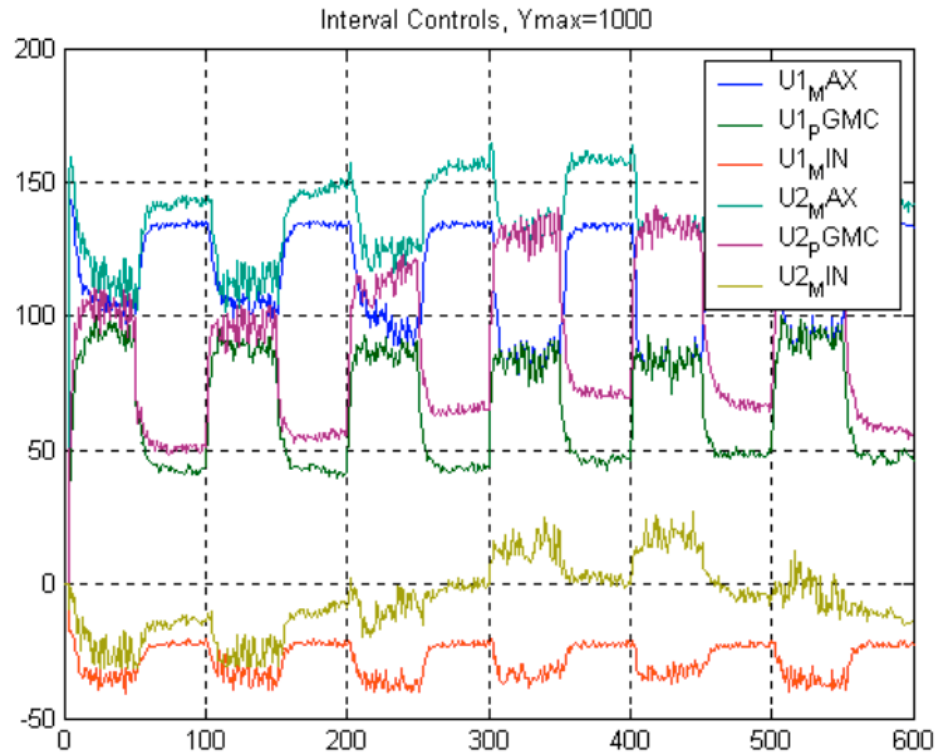


Figure 6-34 Interval Control with $Y_{\max} = 1000$

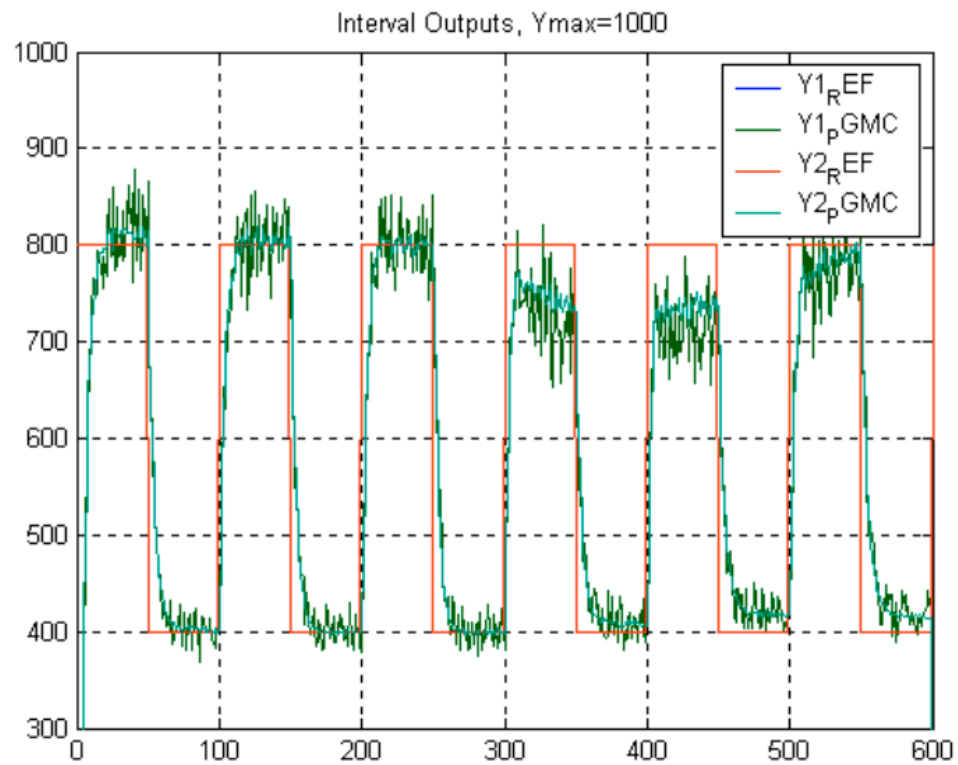


Figure 6-35 Interval Output with $Y_{\max} = 1000$

One other item should be noted for multivariable systems. It is possible in certain circumstances that there may not be a solution to the interval problem when the minimum control boundary is greater than the maximum control boundary. In this case, the order in which the max or min functions in Equation 6-52 becomes the deciding factor for which control boundary will be chosen. Thus, if one stability constraint is more important than the other (i.e. Y_{\max} is more important than Y_{\min}) then this specifies the order in which the max and min functions should be taken. Moreover because the max and min functions are taken across the rows of each control vector, then if necessary the order of the max and min function could be different for each element of the control vector to emphasize the most important stability constraints.

6.6 Discussion

This chapter concludes the creation of the proposed algorithm, which includes control predictions, the use of parameter interval, an adaptive factor, and a stabilizing control interval. The algorithm use of parameter intervals and a corresponding interval problem solution greatly improved the usefulness of the original Generic Model Control method.

CHAPTER 7

CONCLUSION

A multivariable control technique was created for a certain type of nonlinear system with parameter intervals. The control is based upon the feedback linearization scheme called Generic Model Control, and alters the control calculation by including parameter intervals, solving a series of linear inequalities called an interval problem, and also making control predictions. Implementations of the algorithm are done using various arc-welding systems. Also included are several simulations of controlled scalar and multivariable systems. Generally, the contribution of this dissertation in improving the GMC control technique is as follows:

1. Emending the GMC control oscillation through the use of control predictions
2. Determining a method by which suitable gains K_1 and K_2 and the prediction level p can be selected.
3. Expanding the GMC technique to incorporate parameter intervals through the use of an adaptive factor.
4. Ensuring the GMC closed loop stability by the use of parameter intervals and reforming the control calculation into an interval problem.
5. Deriving the closed form solution to the interval problem for $[B] \geq 0$ for scalar and multivariable systems.

7.1 Future Work

The first area for future work is in solving for the particular control in multivariable systems, which will more appropriately satisfy the minimum and maximum output constraints, without forcing one or more outputs to drastically swing through its allowable region, as discussed in the previous section.

The other primary area of future work to be accomplished is in regards to loosening the developed algorithm's model restriction for the interval problem. The closed form solution to the interval problem is only known for $[B] \geq 0$, which for scalar equations is acceptable since $[U]$ can be negative or positive. However, for multivariable systems this is a real restriction because all the elements of $[B]$ have to be greater than or equal to zero, and this thereby excludes processes where a particular input affects one state variable in an inverse manner to any other state variable. Another area for future work is in expanding the interval problem solution for the multivariable linear processes in which the C matrix can also have parameter intervals, because at this time it cannot in order to determine the state variables from the output. Lastly, it is possible that the interval problem and its proposed solution could also be applied to another control techniques besides GMC, and therefore it is possible that the interval problem could be more appropriate according to some criteria with that other control technique.

APPENDICES

LINEAR EXAMPLE

% For the system given in Equation 6-41

```
amin=1.5e-4; amax=3.75e-4; hmin=.5; hmax=.5;
```

% Open Loop Step Response

```
N=50;
U=700*ones(1,N); Y1max=zeros(1,N); Y1min=zeros(1,N); Y1mean=zeros(1,N);
for k=2:N,
    Y1max(1,k) = hmax*Y1max(1,k-1) + amax*U(1,k);
    Y1min(1,k) = hmin*Y1min(1,k-1) + amin*U(1,k);
    Y1mean(1,k) = (hmin+.5*(hmax-hmin))*Y1mean(1,k-1) + (amin+.5*(amax-amin))*U(1,k);
end
figure, plot([1:N], Y1max,[1:N], Y1min,[1:N], Y1mean), legend('Y max', 'Y min', 'Y mean')
```

% Build history of parameters variations (s1 and s2 are the magnitudes of the variations)

```
N=500;
f1=.5*ones(1,N); f2=.5*ones(1,N);
s1=.2; s2=.2;
for ii=2:N,
    f1(1,ii)=max(min(1,f1(ii-1)+s1*(2*rand-1)),0);
    f2(1,ii)=max(min(1,f2(ii-1)+s2*(2*rand-1)),0);
end
```

% Desired Output is Yset

```
Yset=.4*ones(1,N);
```


% Control Parameters: I want it to settle within 4 sampling times(tau=4/.43) and I am going to predict out 3 steps ahead (p=3). Also % during simulation I will vary the F by 50% (dev=.5) and during smoothing I will include the previous input with a weight (W=1).

```
tau=4/.43;
K1=(8/tau);
K2=1/(tau^2);
p=3;
dev=.5;
W=1;
```

% Vector Initializations

```
E=zeros(1,N); Y=0*ones(1,N); U=zeros(1,N); Yref=zeros(1,N);
Hmin=zeros(1,N); Hmax=zeros(1,N); C=zeros(1,N); Ymax=.9*ones(1,N); Umin=zeros(1,N); Umax=zeros(1,N); Ugmc=zeros(1,N);
Ygmc=zeros(1,N); Yrefgmc=zeros(1,N); Egmc=0;
Up=zeros(1,p);
```

for k=3:N-p,

% GMC Control and Output

```
Ygmc(1,k) = (hmin+f1(1,k)*(hmax-hmin))*Ygmc(1,k-1) + (amin+f2(1,k)*(amax-amin))*Ugmc(1,k);
Egmc = Egmc + Yset(1,k)-Ygmc(1,k);
Yrefgmc(1,k) = K1*(Yset(1,k)-Ygmc(1,k)) + K2*Egmc + Ygmc(1,k);
Ugmc(1,k+1)=(Yrefgmc(1,k)-(hmin+.5*(hmax-hmin))*Y(1,k))/(amin+.5*(amax-amin));
```

% Interval Control and Output

```
Y(1,k) = (hmin+f1(1,k)*(hmax-hmin))*Y(1,k-1) + (amin+f2(1,k)*(amax-amin))*U(1,k);
E(1,k) = E(1,k-1)+(Yset(1,k)-Y(1,k));
Yref(1,k) = K1*(Yset(1,k)-Y(1,k)) + K2*E(1,k) + Y(1,k);
Hmin(1,k) = hmin*Y(1,k); Hmax(1,k)=hmax*Y(1,k);
C(1,k) = max(0,(1/((1/amin)+(1/amax)))*((1/amin)*(Yref(1,k)-Hmin(1,k))-(1/amax)*(Yref(1,k)-Hmax(1,k))));
```

```

U1min=-(1/amin)*Hmin(1,k); U2min=-(1/amax)*Hmax(1,k); Umax(1,k+1)=(1/amax)*(Ymax(1,k)-Hmax(1,k));

% Insure that  $U_{k+1}$  is within stabilizing control interval
U(1,k+1)=min([U1min,U2min,0,(1/amax)*(Yref(1,k)+C(1,k)-Hmax(1,k))]),Umax(1,k+1)];
Umin(1,k+1)=max([U1min,U2min,0]);

% Calculate Adaptive Factor
if k==3,
    df=.5;
else
    df=(Y(1,k)-ymin_pred)/(ymax_pred-ymin_pred);
end

yp_1=Y(1,k); Up(1,1)=U(1,k+1); Ep=E(1,k);
if p==1,
    else
        for ii=1:p,

% Interval Control and Output Predictions
            yp=(hmin+df*(hmax-hmin))*yp_1 + (amin+df*(amax-amin))*Up(1,ii);
            Ep = Ep+(Yset(1,k+ii)-yp);
            Yrefp = K1*(Yset(1,k+ii)-yp) + K2*Ep + yp;
            Hminp = hmin*yp; Hmaxp=hmax*yp;
            Cp=(1/((1/amin)+(1/amax)))*((1/amin)*(Yrefp-Hminp)-(1/amax)*(Yrefp-Hmaxp));
            U1minp=-(1/amin)*Hminp; U2minp=-(1/amax)*Hmaxp; Umaxp=-(1/amax)*(Ymax(1,k+ii)-Hmaxp);
            Up(1,ii)=min([U1minp,U2minp,0,(1/amax)*(Yrefp+Cp-Hmaxp)]),(1/amax)*(Ymax(1,k+ii)-Hmaxp));
            yp_1=yp;
            if rem(ii,2)==0,
                df=min(max(df+dev,0),1);
            else

```

```

        df=min(max(df-dev,0),1);
    end
end
end

```

% Average Control Predictions

```

    Utemp=(1/(p+W))*(sum(Up)+W*U(1,k-1));

```

% Again Insure Control is Within Stabilizing Control Interval

```

    U(1,k+1)=min(max(Umin(1,k+1),Utemp),Umax(1,k+1));

```

% Make Control Predictions for Use in Adaptive Factor for the Next Sample

```

    ymax_pred = hmax*Y(1,k) + amax*U(1,k+1);
    ymin_pred = hmin*Y(1,k) + amin*U(1,k+1);
end

```

% Plot Results

```

figure, plot([1:N],U,[1:N],Umin,[1:N],Umax,[1:N],Ugmc), grid, legend('U','Umin','Umax','U_G_M_C')
figure, plot([1:N], Yset, [1:N], Y, [1:N], Ygmc), grid, legend('Yset','Y', 'Y_G_M_C')

```

LINEAR MULTIVARIABLE EXAMPLE

% Simulations for the system given in Equation 6-57

```

t1=2; t2=3; t3=10; t4=12; K1=.5; K2=.06; K3=.08; K4=.4;
nums={K1 K2; K3 K4}; dens={t1 1] [t2 1]; [t3 1] [t4 1]};
sys_ol=tf(nums,dens);
%figure
%step(sys_ol)
%cont=SS(sys_ol,'min')
Ts=.1;

Ac=[-.5 0 0 0; 0 -.1 0 0; 0 0 -.333 0; 0 0 0 -.0833];
Bc=[.5 0; .125 0; 0 .125; 0 .25];
Cc=[.5 0 .16 0; 0 .064 0 .13333];

[A,B,C,D] = C2DM(Ac,Bc,Cc,[0 0;0 0],Ts,'zoh')
[r,m]=size(B); [p,n]=size(C);
s1=.5; s2=.5; s3=.5; s4=.5; s5=.5; s6=.5; s7=.5; s8=.5;
A=[.9500 0 0 0; 0 .99 0 0; 0 0 .9667 0; 0 0 0 .9917];
Amax=[1+Ts*Ac(1,1)*(1-s1) 0 0 0; 0 1+Ts*Ac(2,2)*(1-s2) 0 0; 0 0 1+Ts*Ac(3,3)*(1-s3) 0; 0 0 0 1+Ts*Ac(4,4)*(1-s4)];
Amin=[1+Ts*Ac(1,1)*(1+s1) 0 0 0; 0 1+Ts*Ac(2,2)*(1+s2) 0 0; 0 0 1+Ts*Ac(3,3)*(1+s3) 0; 0 0 0 1+Ts*Ac(4,4)*(1+s4)];

Bmax=[B(1,1)*(1+s5) 0; B(2,1)*(1+s6) 0; 0 B(3,2)*(1+s7); 0 B(4,2)*(1+s8)];
Bmin=[B(1,1)*(1-s5) 0; B(2,1)*(1-s6) 0; 0 B(3,2)*(1-s7); 0 B(4,2)*(1-s8)];

% Open Loop Step Response

sysdl=ss(Amax,Bmax,C,D,Ts);
figure,step(sysdl), title('Maximum Process')

```

```
sysd2=ss(Amin,Bmin,C,D,Ts);
figure,step(sysd2), title('Minimum Process')
```

```
Bmaxi=inv(C*Bmax);
Bmini=inv(C*Bmin);
Bi=inv(C*B);
hmax=C*Amax; hmin=C*Amin; h=C*A;
Cr=C;
Ci=pinv(C);
Rmax=(C*Bmax)*[1;1];
Rmin=(C*Bmin)*[1;1];
```

% Build history of parameters variations (s1 thru s8 are the magnitudes of the variations)

```
Ts=1; Tend=200;
N=round(Tend/Ts);
f1=.5*ones(1,N); f2=.5*ones(1,N); f3=.5*ones(1,N); f4=.5*ones(1,N); f5=.5*ones(1,N);
f6=.5*ones(1,N); f7=.5*ones(1,N); f8=.5*ones(1,N);
for ii=2:N,
    f1(1,ii)=max(min(1,.5+s1*(2*rand-1)),0);
    f2(1,ii)=max(min(1,.5+s2*(2*rand-1)),0);
    f3(1,ii)=max(min(1,.5+s3*(2*rand-1)),0);
    f4(1,ii)=max(min(1,.5+s4*(2*rand-1)),0);
    f5(1,ii)=max(min(1,.5+s5*(2*rand-1)),0);
    f6(1,ii)=max(min(1,.5+s6*(2*rand-1)),0);
    f7(1,ii)=max(min(1,.5+s7*(2*rand-1)),0);
    f8(1,ii)=max(min(1,.5+s8*(2*rand-1)),0);

    %f1(1,ii)=.5+.5*sin(.01*ii);
```

```

%i2(1,ii)=.5+.5*cos(.03*ii);
%i3(1,ii)=max(min(1,f3(ii-1)+s3*(2*rand-1)),0);
%i4(1,ii)=max(min(1,f4(ii-1)+s4*(2*rand-1)),0);
%i5(1,ii)=.5+.1*cos(.01*ii);
%i6(1,ii)=.5+.5*cos(.01*ii);
%i7(1,ii)=max(min(1,f7(ii-1)+s7*(2*rand-1)),0);
%i8(1,ii)=max(min(1,f8(ii-1)+s8*(2*rand-1)),0);

end
figure, plot(Ts*[1:N],f1,Ts*[1:N],f2), legend('Factor for Y1', 'Factor Y2')

```

% Control Parameters: I want it to settle within 20 sampling times(tau=20/.43) and I am going to predict out 4 steps ahead (P=4).
% Also during simulation I will % vary the F by 50% (dev=.1).

```

Tsett_d1=20; Tsett_d2=20;
tau1=Tsett_d1/.43; tau2=Tsett_d2/.43;
K1=diag([(8/tau1) (8/tau2)]);
K2=diag([(1/tau1^2) (1/tau2^2)]);
P=4;
dev=[.1 .1];

```

% Vector Initializations

```

x1=zeros(n,N); y1=zeros(m,N); w1=zeros(m,N); e1=zeros(p,N); E1=zeros(p,1);
x=zeros(n,N); y=zeros(m,N); w=zeros(m,N); e=zeros(p,N); E=zeros(p,1); wmin=zeros(m,N); wmax=zeros(m,N);
xp_min=zeros(n,P); xp_max=zeros(n,P); yp=zeros(m,P); wp=zeros(m,P); ep=zeros(p,1);
xp=zeros(n,p); F=zeros(N,2); ypred_min=zeros(p,N); ypred_max=zeros(p,N); wGMC=zeros(m,N);
T=[1:N].*Ts;

```

% Set the Desired Output yr and the Maximum Output ymax

```

yr=[2*[SQUARE(2*pi*(1/90)*(Ts.*[1:N+P]))]+5; 2*[SQUARE(2*pi*(1/90)*(Ts.*[1:N+P]))]+8];
ymax=[18*ones(1,N+P); 18*ones(1,N+P)]; ymin=[-5*ones(1,N+P); -5*ones(1,N+P)];

```

```

% GMC Control and Output
for kk=2:N,
    Ar=[Amin(1,1)+f1(1,kk)*(Amax(1,1)-Amin(1,1)) 0 0 0; 0 Amin(2,2)+f2(1,kk)*(Amax(2,2)-Amin(2,2)) 0 0;...
        0 0 Amin(3,3)+f3(1,kk)*(Amax(3,3)-Amin(3,3)) 0; 0 0 0 Amin(4,4)+f4(1,kk)*(Amax(4,4)-Amin(4,4))];
    Br=[Bmin(1,1)+f5(1,kk)*(Bmax(1,1)-Bmin(1,1)) 0; Bmin(2,1)+f6(1,kk)*(Bmax(2,1)-Bmin(2,1)) 0;...
        0 Bmin(3,2)+f7(1,kk)*(Bmax(3,2)-Bmin(3,2)); 0 Bmin(4,2)+f8(1,kk)*(Bmax(4,2)-Bmin(4,2))];

    x1(:,kk)=Ar*x1(:,kk-1)+Br*w1(:,kk-1);
    y1(:,kk)=Cr*x1(:,kk);
    e1(:,kk)=yr(:,kk)-y1(:,kk);
    E1=E1+e1(:,kk)*Ts;
    w1(:,kk)=Bi*((K1*e1(:,kk)+K2*E1)*Ts + y1(:,kk) - h*x1(:,kk));

end
figure, plot(T,w1(1,:),T,w1(2,:)), grid, legend('Control 1', 'Control 2'), title('GMC Control')
figure, plot(T,yr(1,[1:N]),T,y1(1,:),T,yr(2,[1:N]),T,y1(2,:)), grid, legend('Reference 1', 'Output 1', 'Reference 2', 'Output 2'), ...
title('GMC Output')

```

% Initial Factor, F, is equal to .5
F(1,:)= [.5 .5];

% Simulate Interval PGMC Control and Output
for kk=2:N,

% Found by changing each parameter by a Factor
 Ar=[Amin(1,1)+f1(1,kk)*(Amax(1,1)-Amin(1,1)) 0 0 0; 0 Amin(2,2)+f2(1,kk)*(Amax(2,2)-Amin(2,2)) 0 0;...
 0 0 Amin(3,3)+f3(1,kk)*(Amax(3,3)-Amin(3,3)) 0; 0 0 0 Amin(4,4)+f4(1,kk)*(Amax(4,4)-Amin(4,4))];
 Br=[Bmin(1,1)+f5(1,kk)*(Bmax(1,1)-Bmin(1,1)) 0; Bmin(2,1)+f6(1,kk)*(Bmax(2,1)-Bmin(2,1)) 0;...
 0 Bmin(3,2)+f7(1,kk)*(Bmax(3,2)-Bmin(3,2)); 0 Bmin(4,2)+f8(1,kk)*(Bmax(4,2)-Bmin(4,2))];

```

x(:,kk)=Ar*x(:,kk-1)+Br*w(:,kk-1);
y(:,kk)=Cr*x(:,kk);

% Calculate Adaptive Factor
if kk==2,
    F(kk,1)=.5; F(kk,2)=.5;
else
    F(kk,:)=[(y(1,kk)-Ypred_min(1,1))/(Ypred_max(1,1)-Ypred_min(1,1))...
              (y(2,kk)-Ypred_min(2,1))/(Ypred_max(2,1)-Ypred_min(2,1))];
end

e(:,kk)=yr(:,kk)-y(:,kk);
E=E+e(:,kk)*Ts;
Yrefk=(K1*e(:,kk)+K2*E)*Ts + y(:,kk);
Yrefk=max([min([Yrefk,ymax(:,kk)],[],2),ymin(:,kk)],[],2);
w_gmc=inv(C*Bmin+[F(kk,1) 0; 0 F(kk,2)]*(C*Bmax-C*Bmin))*(Yrefk - (hmin+[F(kk,1) 0; 0 F(kk,2)]*...
              (hmax-hmin))*x(:,kk));
wGMC(:,kk)=w_gmc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determine Maximum and Minimum Inputs
Lmax1=min(((ymax(:,kk)-hmax*x(:,kk)-C*Bmax*w_gmc)./Rmax));
w_max1=[w_gmc(1,1)+Lmax1; w_gmc(2,1)+Lmax1];
Lmax2=min(((ymax(:,kk)-hmax*x(:,kk)-C*Bmin*w_gmc)./Rmin));
w_max2=[w_gmc(1,1)+Lmax2; w_gmc(2,1)+Lmax2];
w_max=min([w_max1,w_max2],[],2);

Lmin1=max(((ymin(:,kk)-hmin*x(:,kk)-C*Bmin*w_gmc)./Rmin));
w_min1=[w_gmc(1,1)+Lmin1; w_gmc(2,1)+Lmin1];

```



```

0 F(kk,2)]*(hmax-hmin))*xp(:,ii));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

% Determine Maximum and Minimum Inputs

```

Lmax1=min(((ymax(:,kk+ii)-hmax*xp(:,ii)-C*Bmax*wp_gmc)./Rmax));
wp_max1=[wp_gmc(1,1)+Lmax1; wp_gmc(2,1)+Lmax1];
Lmax2=min(((ymax(:,kk+ii)-hmax*xp(:,ii)-C*Bmin*wp_gmc)./Rmin));
wp_max2=[wp_gmc(1,1)+Lmax2; wp_gmc(2,1)+Lmax2];
wp_max=min([wp_max1,wp_max2],[1,2]);

Lmin1=max(((ymin(:,kk+ii)-hmin*xp(:,ii)-C*Bmin*wp_gmc)./Rmin));
wp_min1=[wp_gmc(1,1)+Lmin1; wp_gmc(2,1)+Lmin1];
Lmin2=max(((ymin(:,kk+ii)-hmin*xp(:,ii)-C*Bmax*wp_gmc)./Rmax));
wp_min2=[wp_gmc(1,1)+Lmin2; wp_gmc(2,1)+Lmin2];
wp_min=max([wp_min1,wp_min2],[1,2]);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

% Insure that U_{k+i} is Within Stabilizing Control Interval

```

wp(:,ii)=max([min([wp_max,wp_gmc],[1,2),wp_min],[1,2);

```

end

end

% Average Control Predictions

```

w(:,kk)=[([P:-1:1]/sum([P:-1:1])*wp(1,:); [P:-1:1]/sum([P:-1:1])*wp(2,:)]';

```

```

% Insure that  $U_{k+i}$  is Within Stabilizing Control Interval
w(:,kk)=max([min([w_max,w_gmc],[1,2]),w_min],[1,2]);

% Make Control Predictions for Use in Adaptive Factor for the Next Sample
Ypred_max=C*(Bmax*w(:,kk) + Amax*x(:,kk)); Ypred_min=C*(Bmin*w(:,kk) + Amin*x(:,kk));
ypred_max(:,kk)=Ypred_max; ypred_min(:,kk)=Ypred_min;
end

% Plot Results
figure, plot(T,wmax(1,:),T,w(1,:),T,wmin(1,:),T,wmax(2,:),T,w(2,:),T,wmin(2,:)), legend('Control 1 max', 'Control 1', ...
'Control 1 min', 'Control 2 max', 'Control 2', 'Control 2 min'), title('Interval PGMC Control')
figure,
subplot(2,1,1),plot(Ts*[1:N],F(:,1)), grid, title('F1')
subplot(2,1,2),plot(Ts*[1:N],F(:,2)), grid, title('F2')
figure, plot(T,yr(1,[1:N]),T,y(1,:),T,yr(2,[1:N]),T,y(2,:)), grid, legend('Reference 1', 'Output 1', 'Reference 2', 'Output 2'), ...
title('Interval PGMC Output')

```

NONLINEAR MULTIVARIABLE EXAMPLE

% For the System Given in Equation 6-59.60

```

a10_min=100; a11_min=-.00006; a12_min=.0002; a13_min=5; a14_min=1;
a10_max=120; a11_max=-.000001; a12_max=.00035; a13_max=6; a14_max=1;

a20_min=85; a21_min=-.00009; a22_min=.0001; a23_min=1; a24_min=4;
a20_max=95; a21_max=0; a22_max=.0004; a23_max=1; a24_max=5;

a10=(a10_min+a10_max)/2; a11=(a11_min+a11_max)/2; a12=(a12_min+a12_max)/2; a13=(a13_min+a13_max)/2;
a14=(a14_min+a14_max)/2;
a20=(a20_min+a20_max)/2; a21=(a21_min+a21_max)/2; a22=(a22_min+a22_max)/2; a23=(a23_min+a23_max)/2;
a24=(a24_min+a24_max)/2;
Bgmc=[a13 a14; a23 a24];
Bgmci=inv(Bgmc);
Bmax=[a13_max a14_max; a23_max a24_max]; Bmin=[a13_min a14_min; a23_min a24_min];
Rmax=(Bmax)*[1;1];
Rmin=(Bmin)*[1;1];

% Open Loop Step Response
N=250; U1=100*ones(1,N); U2=100*ones(1,N); Y1min=zeros(1,N); Y2min=zeros(1,N); Y1max=zeros(1,N); Y2max=zeros(1,N);
Y1mean=zeros(1,N); Y2mean=zeros(1,N);

```

for k=2:N-1,

$$Y1min(1,k+1) = a10_min + a11_min*(Y1min(1,k)^2)*sin(1*Y1min(1,k)) + a12_min*Y1min(1,k-1)*Y2min(1,k-1) + \dots$$

$$a13_min*U1(1,k) + a14_min*U2(1,k);$$

$$Y2min(1,k+1) = a20_min + a21_min*(Y2min(1,k)^2)*sin(1*Y2min(1,k)) + a22_min*Y1min(1,k-1)*Y2min(1,k-1) + \dots$$

```

a23_min*U1(1,k) + a24_min*U2(1,k);
Y1max(1,k+1) = a10_max + a11_max*(Y1max(1,k)^2)*sin(1*Y1max(1,k)) + a12_max*Y1max(1,k-1)*Y2max(1,k-1) + ...
a13_max*U1(1,k) + a14_max*U2(1,k);
Y2max(1,k+1) = a20_max + a21_max*(Y2max(1,k)^2)*sin(1*Y2max(1,k)) + a22_max*Y1max(1,k-1)*Y2max(1,k-1) + ...
a23_max*U1(1,k) + a24_max*U2(1,k);
Y1mean(1,k+1) = a10 + a11*(Y1mean(1,k)^2)*sin(1*Y1mean(1,k)) + a12*Y1mean(1,k-1)*Y2mean(1,k-1) + a13*U1(1,k) ...
+ a14*U2(1,k);
Y2mean(1,k+1) = a20 + a21*(Y2mean(1,k)^2)*sin(1*Y2mean(1,k)) + a22*Y1mean(1,k-1)*Y2mean(1,k-1) + a23*U1(1,k) ...
+ a24*U2(1,k);

end
%figure, plot([1:N],Y1min,[1:N],Y1mean,[1:N],Y1max,[1:N],Y2min,[1:N],Y2mean,[1:N],Y2max), grid,
legend('Y1min','Y1mean','Y1max','Y2min','Y2mean','Y2max')
figure, plot([1:N],Y1min,[1:N],Y1max,[1:N],Y2min,[1:N],Y2max), grid, legend('Y1min','Y1max','Y2min','Y2max'),...
title('Open Loop Step Response')

```

% Build history of parameters variation (s1 thru s10 are the magnitude of the variation)

```

Ts=1; Tend=600;
N=round(Tend/Ts);
s1=.5; s2=.5; s3=.5; s4=.5; s5=.5; s6=.5; s7=.5; s8=.5; s9=.5; s10=.5;
f1=.5+s1*(2*rand(1,N)-1); f2=.5+s2*(2*rand(1,N)-1); f3=.5+s3*(2*rand(1,N)-1);
f4=.5+s4*(2*rand(1,N)-1); f5=.5+s5*(2*rand(1,N)-1);
f6=.5+s6*(2*rand(1,N)-1); f7=.5+s7*(2*rand(1,N)-1); f8=.5+s8*(2*rand(1,N)-1);
f9=.5+s9*(2*rand(1,N)-1); f10=.5+s10*(2*rand(1,N)-1);
for ii=8:N,
    f1(1,ii)=.5+.5*sin(.01*ii);
    f2(1,ii)=.5+.5*cos(.03*ii);
    f3(1,ii)=max(min(1, (.5+2.5*s3*(2*rand-1)+f3(1,ii-1)+f3(1,ii-2)+f3(1,ii-3)+f3(1,ii-4)+f3(1,ii-5)+f3(1,ii-6)+f3(1,ii-7))/8),0);
    f4(1,ii)=max(min(1, (.5+s4*(2*rand-1))),0);
    f5(1,ii)=.5+.1*cos(.01*ii);
    f6(1,ii)=.5+.5*cos(.01*ii);

```

```

f7(1,ii)=max(min(1,(.5+1.6*s7*(2*rand-1)+f7(1,ii-1)+f7(1,ii-2)+f7(1,ii-3)+f7(1,ii-4)+f7(1,ii-5)+f7(1,ii-6)+f7(1,ii-7))/(8),0);
f8(1,ii)=.5+.5*sin(.01*ii);
f9(1,ii)=max(min(1,(.5+1.6*s9*(2*rand-1)+f9(1,ii-1)+f9(1,ii-2)+f9(1,ii-3)+f9(1,ii-4)+f9(1,ii-5)+f9(1,ii-6)+f9(1,ii-7))/(8),0);
f10(1,ii)=.5+.5*cos(.01*ii);
end
figure,
subplot(5,2,1), plot(Ts*[1:N],f1)
subplot(5,2,2), plot(Ts*[1:N],f2)
subplot(5,2,3), plot(Ts*[1:N],f3)
subplot(5,2,4), plot(Ts*[1:N],f4)
subplot(5,2,5), plot(Ts*[1:N],f5)
subplot(5,2,6), plot(Ts*[1:N],f6)
subplot(5,2,7), plot(Ts*[1:N],f7)
subplot(5,2,8), plot(Ts*[1:N],f8)
subplot(5,2,9), plot(Ts*[1:N],f9)
subplot(5,2,10), plot(Ts*[1:N],f10)

```

% Control Parameters : I want it to settle within 15 sampling times(tau= 15/.43) and I am not going to predict out any steps

```

Tsett_d1=15; Tsett_d2=15;
tau1=Tsett_d1/.43; tau2=Tsett_d2/.43;
K11=(8/tau1); K12=(8/tau2); K21=(1/tau1^2); K22=(1/tau2^2);

```

```

m=2; p=2;
T=[1:N].*Ts;

```

% Desired Outputs are Yset1 and Yset2 and Maximum and Minimum Outputs Ymax1, Ymax2, Ymin1, and Ymin2

```

Yset1=200*[SQUARE(2*pi*(1/100)*(Ts.*[1:N+P]))]+600; Yset2=200*[SQUARE(2*pi*(1/100)*(Ts.*[1:N+P]))]+600;
Ymax1=1000*ones(1,N+P); Ymax2=1000*ones(1,N+P); Ymin1=-5*ones(1,N+P); Ymin2=-5*ones(1,N+P);

```

```

% Vector Initializations
Y1=zeros(1,N); Y2=zeros(1,N); U1=zeros(1,N); U2=zeros(1,N); E1=0; E2=0;
Y1_gmc=zeros(1,N); Y2_gmc=zeros(1,N); U1_gmc=zeros(1,N); E1_gmc=0; E2_gmc=0;
U1max=zeros(1,N); U2max=zeros(1,N); U1min=zeros(1,N); U2min=zeros(1,N); F1=zeros(1,N); F2=zeros(1,N);
Y1pred_min=zeros(1,N); Y1pred_max=zeros(1,N); Y2pred_min=zeros(1,N); Y2pred_max=zeros(1,N);

% GMC Control and Output
for k=3:N-1,
    Y1_gmc(1,k) = (a10_min+f1(1,k)*(a10_max-a10_min)) + (a11_min+f2(1,k-1)*(a11_max-a11_min))*...
        (Y1_gmc(1,k-1)^2)*sin(Y1_gmc(1,k-1)) + (a12_min+f3(1,k)*(a12_max-a12_min))*Y1_gmc(1,k-2)*...
        Y2_gmc(1,k-2) + (a13_min+f4(1,k)*(a13_max-a13_min))*U1_gmc(1,k) + (a14_min+f5(1,k)*...
        (a14_max-a14_min))*U2_gmc(1,k);
    Y2_gmc(1,k) = (a20_min+f6(1,k)*(a20_max-a20_min)) + (a21_min+f7(1,k-1)*(a21_max-a21_min))*...
        (Y2_gmc(1,k-1)^2)*sin(Y2_gmc(1,k-1)) + (a22_min+f8(1,k)*(a22_max-a22_min))*Y1_gmc(1,k-2)*...
        Y2_gmc(1,k-2) + (a23_min+f9(1,k)*(a23_max-a23_min))*U1_gmc(1,k) + (a24_min+f10(1,k)*...
        (a24_max-a24_min))*U2_gmc(1,k);

    e1_gmc = Ysetl(1,k)-Y1_gmc(1,k);
    e2_gmc = Yset2(1,k)-Y2_gmc(1,k);
    E1_gmc = e1_gmc+E1_gmc;
    E2_gmc = e2_gmc+E2_gmc;

    Yref1_gmc = K11*e1_gmc + K21*E1_gmc + Y1_gmc(1,k);
    Yref2_gmc = K12*e2_gmc + K22*E2_gmc + Y2_gmc(1,k);
    Ugmc=Bgmc*( [Yref1_gmc - (a10 + a11*(Y1_gmc(1,k)^2)*sin(Y1_gmc(1,k)) + a12*Y1_gmc(1,k-1)*Y2_gmc(1,k-1)); ...
        Yref2_gmc - (a20 + a21*(Y2_gmc(1,k)^2)*sin(Y2_gmc(1,k)) + a22*Y1_gmc(1,k-1)*Y2_gmc(1,k-1))] );
    U1_gmc(1,k+1)=Ugmc(1,1);
    U2_gmc(1,k+1)=Ugmc(2,1);
end

```

```
figure, plot(T,U1_gmc,T,U2_gmc), grid, legend('U1_GMC', 'U2_GMC'), title('GMC Controls')
figure, plot(T,Yset1(1,[1:N]),T,Y1_gmc,T,Yset2(1,[1:N]),T,Y2_gmc), grid, legend('Y1_REF', 'Y1_GMC', 'Y2_REF', 'Y2_GMC'),
title('GMC Outputs')
```

% Interval Control and Output

```
for k=3:N-1,
    Y1(1,k) = (a10_min+f1(1,k)*(a10_max-a10_min)) + (a11_min+f2(1,k-1)*(a11_max-a11_min))*(Y1(1,k-1)^2)*...
               sin(Y1(1,k-1)) + (a12_min+f3(1,k)*(a12_max-a12_min))*Y1(1,k-2)*Y2(1,k-2) + (a13_min+f4(1,k)*...
               (a13_max-a13_min))*U1(1,k) + (a14_min+f5(1,k)*(a14_max-a14_min))*U2(1,k);
    Y2(1,k) = (a20_min+f6(1,k)*(a20_max-a20_min)) + (a21_min+f7(1,k-1)*(a21_max-a21_min))*(Y2(1,k-1)^2)*...
               sin(Y2(1,k-1)) + (a22_min+f8(1,k)*(a22_max-a22_min))*Y1(1,k-2)*Y2(1,k-2) + (a23_min+f9(1,k)*...
               (a23_max-a23_min))*U1(1,k) + (a24_min+f10(1,k)*(a24_max-a24_min))*U2(1,k);

    e1 = Yset1(1,k)-Y1(1,k);
    e2 = Yset2(1,k)-Y2(1,k);
    E1 = e1 + E1;
    E2 = e2 + E2;

    Yref1 = max(min(K11*e1 + K21*E1 + Y1(1,k),Ymax1(1,k+1)),Ymin1(1,k+1));
    Yref2 = max(min(K12*e2 + K22*E2 + Y2(1,k),Ymax2(1,k+1)),Ymin2(1,k+1));

    Hmax1 = a10_max + max(a11_min*(Y1(1,k)^2)*sin(Y1(1,k)),a11_max*(Y1(1,k)^2)*sin(Y1(1,k)))...
             + max(a12_min*Y1(1,k-1)*Y2(1,k-1),a12_max*Y1(1,k-1)*Y2(1,k-1));
    Hmax2 = a20_max + max(a21_min*(Y2(1,k)^2)*sin(Y2(1,k)),a21_max*(Y2(1,k)^2)*sin(Y2(1,k)))...
             + max(a22_min*Y1(1,k-1)*Y2(1,k-1),a22_max*Y1(1,k-1)*Y2(1,k-1));
    Hmin1 = a10_min + min(a11_min*(Y1(1,k)^2)*sin(Y1(1,k)),a11_max*(Y1(1,k)^2)*sin(Y1(1,k)))...
             + min(a12_min*Y1(1,k-1)*Y2(1,k-1),a12_max*Y1(1,k-1)*Y2(1,k-1));
    Hmin2 = a20_min + min(a21_min*(Y2(1,k)^2)*sin(Y2(1,k)),a21_max*(Y2(1,k)^2)*sin(Y2(1,k)))...
             + min(a22_min*Y1(1,k-1)*Y2(1,k-1),a22_max*Y1(1,k-1)*Y2(1,k-1));
```


% Calculate Adaptive Factor

```

if k==3,
    F1(1,k)=.5; F2(1,k)=.5;
else
    F1(1,k)=(Y1(1,k)-Y1pred_min(1,k))/(Y1pred_max(1,k)-Y1pred_min(1,k));
    F2(1,k)=(Y2(1,k)-Y2pred_min(1,k))/(Y2pred_max(1,k)-Y2pred_min(1,k));
end

w_gmc=inv(Bmin + [F1(1,k) 0; 0 F2(1,k)]*(Bmax-Bmin))*([Yref1 - (Hmin1+F1(1,k)*(Hmax1-Hmin1)); ...
    Yref2 - (Hmin2+F2(1,k)*(Hmax2-Hmin2))]);
ymax=[Ymax1(1,k); Ymax2(1,k)]; hmax=[Hmax1; Hmax2];
ymin=[Ymin1(1,k); Ymin2(1,k)]; hmin=[Hmin1; Hmin2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine Maximum and Minimum Inputs
Lmax1=min((ymax - hmax - Bmax*w_gmc)./Rmax);
w_max1=[w_gmc(1,1)+Lmax1; w_gmc(2,1)+Lmax1];
Lmax2=min((ymax - hmax - Bmin*w_gmc)./Rmin);
w_max2=[w_gmc(1,1)+Lmax2; w_gmc(2,1)+Lmax2];
w_max=min([w_max1, w_max2], [], 2);

Lmin1=max(((ymin - hmin - Bmin*w_gmc)./Rmin));
w_min1=[w_gmc(1,1)+Lmin1; w_gmc(2,1)+Lmin1];
Lmin2=max(((ymin - hmin - Bmax*w_gmc)./Rmax));
w_min2=[w_gmc(1,1)+Lmin2; w_gmc(2,1)+Lmin2];
w_min=max([w_min1, w_min2], [], 2);
U1max(1,k+1)=w_max(1,1); U2max(1,k+1)=w_max(2,1); U1min(1,k+1)=w_min(1,1); U2min(1,k+1)=w_min(2,1);

```

%%%

% Insure that U_{k+1} is Within Stabilizing Control Interval

```
w=max([w_max,w_gmc,[],2],w_min,[],2);
U1(1,k+1)=w(1,1); U2(1,k+1)=w(2,1);
```

% Make Control Predictions for Use in Adaptive Factor for the Next Sample

```
Y1pred_min(1,k+1) = a10_min + min(a11_min*(Y1(1,k)^2)*sin(Y1(1,k)),a11_max*(Y1(1,k)^2)*sin(Y1(1,k)))...
+ min(a12_min*Y1(1,k-1)*Y2(1,k-1),a12_max*Y1(1,k-1)*Y2(1,k-1))...
+ min(a13_min*U1(1,k+1),a13_max*U1(1,k+1)) + min(a14_min*U2(1,k+1),a14_max*U2(1,k+1));
Y1pred_max(1,k+1) = a10_max + max(a11_min*(Y1(1,k)^2)*sin(Y1(1,k)),a11_max*(Y1(1,k)^2)*sin(Y1(1,k)))...
+ max(a12_min*Y1(1,k-1)*Y2(1,k-1),a12_max*Y1(1,k-1)*Y2(1,k-1))...
+ max(a13_min*U1(1,k+1),a13_max*U1(1,k+1)) + max(a14_min*U2(1,k+1),a14_max*U2(1,k+1));
Y2pred_min(1,k+1) = a20_min + min(a21_min*(Y2(1,k)^2)*sin(Y2(1,k)),a21_max*(Y2(1,k)^2)*sin(Y2(1,k)))...
+ min(a22_min*Y1(1,k-1)*Y2(1,k-1),a22_max*Y1(1,k-1)*Y2(1,k-1))...
+ min(a23_min*U1(1,k+1),a23_max*U1(1,k+1)) + min(a24_min*U2(1,k+1),a24_max*U2(1,k+1));
Y2pred_max(1,k+1) = a20_max + max(a21_min*(Y2(1,k)^2)*sin(Y2(1,k)),a21_max*(Y2(1,k)^2)*sin(Y2(1,k)))...
+ max(a22_min*Y1(1,k-1)*Y2(1,k-1),a22_max*Y1(1,k-1)*Y2(1,k-1))...
+ max(a23_min*U1(1,k+1),a23_max*U1(1,k+1)) + max(a24_min*U2(1,k+1),a24_max*U2(1,k+1));
```

end

% Plot Results

```
figure, plot(T,U1max, T,U1, T,U1min, T,U2max, T,U2, T,U2min), grid,
legend('U1_MAX','U1_PGMC','U1_MIN','U2_MAX','U2_PGMC','U2_MIN'), title('Interval Controls')
figure, plot(T,Yset1(1,[1:N]),T,Y1,T,Yset2(1,[1:N]),T,Y2), grid, legend('Y1_REF', 'Y1_PGMC', 'Y2_REF', 'Y2_PGMC'),
title('Interval Outputs'), axis([0 600 300 1000])
figure, plot(T,F1,T,F2), grid, legend('F1', 'F2'), title('Adaptive Factors')
```

REFERENCES

- [1] P. L. Lee, 1993. Nonlinear Process Control: Application of Generic Model Control. Springer-Verlag, London.
- [2] C. J. Harris, J. B. Gao, 2002. An adaptive multiscale basis method for modelling discrete-time non-linear dynamical systems. Int. J. Control, VOL. 75, NO. 3, 141–153, Taylor & Francis, Ltd.
- [3] D. Coca, S. A. Billings, Non-linear system identification using wavelet multiresolution models. Int. J. Control, 2001, VOL. 74, NO. 18, 1718-1736. , Taylor & Francis, Ltd.
- [4] W.J. Rugh, (1981). Nonlinear System Theory. Baltimore and London: John Hopkins University Press.
- [5] F.J. Doyle III, A. O. Babatunde, R. P. Pearson, 1994, Nonlinear Model-Based Control Using Second Order Volterra Models, E.I. DuPont deNemours & Co., Inc. Experimental Station E1, Wilmington, DE.
- [6] J. Bendat, (1990), Nonlinear System Analysis and Identification from Random Data, John Wiley & Sons, New York, NY.
- [7] T. H. Van Pelt, D. S. Bernstein, 2001, Non-linear system identification using Hammerstein and non-linear feedback models with piecewise linear static maps. Int. J. Control, 2001, VOL. 74, NO. 18, 1807-1823.

- [8] H. A. Barker, A. H. Tan, K. R. Godfrey, 2002, Wiener Models of Direction-Dependent Dynamic Systems, IFAC 15th Triennial World Congress, Barcelona, Spain
- [9] J. C. Gómez , E. Baeyens, 2002, Subspace Identification of Multivariable Hammerstein and Weiner Models, IFAC15th Triennial World Congress, Barcelona, Spain
- [10] P. Crama, J. Schoukens, 2002, First estimates of Weiner-Hammerstein Systems using a Random Multisine Excitation, IFAC 15th Triennial World Congress, Barcelona, Spain
- [11] E. Bai, 2002, A Blind Approach to the Hammerstein-Weiner Model Identification. , IFAC 15th Triennial World Congress, Barcelona, Spain
- [12] J. Abonyi and R. Babuska. Local and global identification and interpretation of parameters in Takagi–Sugeno fuzzy models. In *Proceedings 9th IEEE International Conference on Fuzzy Systems – FUZZ-IEEE 2000*, pages 835–840, San Antonio, USA, May 2000.
- [13] J. Abonyi, R. Babuska, M. Setnes, H.B. Verbruggen, and F. Szeifert. Constrained parameter estimation in fuzzy modeling. In *Proceedings FUZZ-IEEE'99*, pages 951–956, Seoul, Korea, August 1999.
- [14] C. H. Chen, “Probabilistic Neural Networks and General Regression Neural Networks,” *Fuzzy Logic and Neural Network Handbook*. Vol.1, 1993, pp. 1-25.
- [15] Y. Yang, “Minimax Nonparametric Classification – Part II: Model Selection for Adaptation.” *IEEE Transactions on Information Theory*. Vol. 45, no. 7, 1999, pp. 2285-2291.

- [16] D. Lee, "Pattern Sequence Recognition Using a Time-Varying Hopfield Network." IEEE Transactions on Neural Networks. Vol. 13, no. 2, 2002, pp. 330-341.
- [17] M. J. Er, and S. Wu, "Face Recognition with Radial Basis Function Neural Networks." IEEE Transactions on Neural Networks. Vol. 13, no. 3, 2002, pp. 697-709.
- [18] A. K. Jain, R. P. Duin, and J. Mao, "Statistical Pattern Recognition: a Review." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.22, no. 1, January 2000, pp.4-37.
- [19] N. N. Aizenberg, "Proceedings of the Third IEEE International Workshop on Cellular Neural Networks." IEEE International Workshop on Cellular Neural Networks. 1994, pp.207-212.
- [20] G.C. Goodwin, A Brief Overview of Nonlinear Control, CIDAC, Newcastle Australia.
- [21] M. Kothare, P.Campo, M. Morari and C. Nett, A unified framework for the study of anti-windup Designs, Automatica, Col. 30, No. 12, pp. 1869-1886, 1994.
- [22] R. Findeisen and F.Allgower, "An Introduction to Nonlinear Model Predictive Control." Institute for Systems Theory in Engineering, Stuttgart, Germany.
- [23] L. Magni, G. DeNicolao, R. Scattolini, and F. Allgower, "Robust Model Predictive Control for Nonlinear Discrete-Time Systems," International Journal of Robust and Nonlinear Control, 2003,13:229-246.
- [24] J. Rawlings, "Tutorial Overview of Model Predictive Control," IEEE Control Systems Magazine, pp. 38-52, June 2000.

- [25] Krstic, Miroslav, Kanellakopoulos, Ioannis, and Koktovic, Petar, Nonlinear and Adaptive Control Design, John Wiley & Sons, Inc., 1995.
- [26] W. C. A. Maas, Nonlinear H-infinity control: the singular case, Stichting Mathematisch Centrum, Amstersam, Netherlands, 1996.
- [27] J. Abonyi, L. Nagy, and F. Szeifert. Adaptive fuzzy inference system and its application in modelling and model based control. Chem. Eng. Res. Des. (Trans. IChemE), 77, A3:281–290, 1999.
- [28] R. Babuska. Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston, 1998.
- [29] K.M. Bossley. Neurofuzzy Modelling Approaches in System Identification. PhD thesis, University of Southampton, 1997.
- [30] D. W. Clarke, C. Mothadi, and P.S.C. Tuffs. Generalized predictive control - part I. the basic algorithm. Automatica, 23:137–148, 1989.
- [31] D. W. Clarke and C. Mohtadi. Properties of generalized predictive control. Automatica, 25(6):859–875, 1989.
- [32] M. Ayala Botto, T.J.J. van den Boom, A. Krijgsman, and J. S. Costa. Constrained nonlinear predictive control based on input-output linearization using a neural network. International Journal of Control, 72(17):1538–1554, 1999.
- [33] M. Ferch, J. Zhang, and A. Knoll, “Robot Skill Transfer Based on B-Spline Fuzzy Controllers for Force-Control Tasks.” Proceedings of the 1999 IEEE International Conference on Robotics & Automation. May 1999, pp.1170-1175
- [34] K. A. McDonald, and T. J. McAvoy, and A. Tits, (1986). Optimal averaging level control. AIChE J. 32(1), 75.

- [35] E. Hernandez, and Y. Arkun, (1993). Control of non-linear systems using polynomial ARMA models. *AIChE Journal*, 39(3), pp. 446-460.
- [36] N. K. Read, and W. H. Ray, (1998). Application of non-linear dynamic analysis in the identification and control of non-linear systems- Simple dynamics. *Journal of Process Control*, Vol. 8 (1), pp. 1-15.
- [37] Z. Lu, and B. R. Holt, (1990). Non-linear robust control: Table look-up controller design. *Proceedings of the American control conference*, San Diego, CA. Pp. 2758-2763.
- [38] V. Manousiouthakis, (1990) A game theoretic approach to robust controller synthesis. *Com. Chem. Eng.*, 14(4/5), pp.381-389.
- [39] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63:331–349, 1996.
- [40] S. Gibson, B. M. Ninness, 2002, Maximum Likelihood Identification of Bilinear Systems. *IFAC 15th Triennial World Congress*, Barcelona, Spain
- [41] V. Verdult, M., Verhaegen, Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *Int. J. Control*, 2001, VOL. 74, NO. 18, 1824-1836.
- [42] S. K. P. Wong, and D. E. Seborg, (1986b). Control strategies for non-linear multivariable systems with time delays. *Proceedings of the American control conference.*, pp. 1023.
- [43] P. L. Lee, and G. R. Sullivan, and W. Zhou, (1990). A New Multivariable Deadtime Control Algorithm. *Chem. Eng. Comm.*, 91, pp. 49-63.

- [44] M. W. Brown, P.L. Lee, G. R. Sullivan, and W. Zhou, (1990). A Constrained Non-linear Multivariable Control Algorithm. *Trans IchemE*, 68, pp. 464-476.
- [45] P. L. Lee and G. R. Sullivan, 1988. Generic Model Control (GMC). *Computer Chemical Engineering*. Vol.12, No. 6, pp. 573-580.
- [46] R. C. Bartusiak, C. Georgakis, and M. J. Reilly, (1988) Designing Nonlinear Control Structures by Reference Systems Synthesis. *Proc. ACC*, Atlanta GA USA, 1585-1590.
- [47] J. G. Balchen, B. Lie and I. Solberg, (1988) Internal decoupling in nonlinear process control. *Modelling, Identif. and Control* 9:137-148.
- [48] S.L. Liu, (1967) Noninteracting process control. *Ind. Eng. Chem. Proc. Des. and Dev.* 6:460-468.
- [49] R. M. Ansari and M. O. Tade, 2000. *Non-linear Model-Based Process Control*. London: Springer-Verlag.
- [50] P. L. Lee and Y. Samyudia, Robust Generic Model Control (GMC) Design for Uncertain Nonlinear Processes, *IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 2002, pp.1046-1047.
- [51] P. L. Lee and Y. Samyudia, Iterative design of robust generic model controllers for industrial process, *Control Engineering Practice*, December 2003.
- [52] W. Zhoi, P. L. Lee, and G. R. Sullivan, Robust Stability Analysis of Generic Model Control, *Chem. Eng. Comm*, 1992, Vol. 117, pp.41-72.
- [53] P. L. Lee and W. Zhou, A new multivariable deadtime control algorithm, *Chem Eng. Comm.* 91 (1) (1990) 49-63.

- [54] R. H. Dunia, T. F. Edgar, and B. Fernandez, Effect of process uncertainties on generic model control: a geometric approach, *Chem Engr. Sci.* 52 (14) (1997) 2205-2222. 257-6262 X 223
- [55] X. Q. Xie, D. H. Zhou, D.H., and Y. H. Jin, Strong tracking filter based adaptive generic model control, *Journal of Process Control*, pp. 337-350, 1999.
- [56] F. Y. Wang and I. T. Cameron, Control studies on a model evaporation process – constrained state driving with conventional and higher relative degree, *Journal of Process Control*, May 1994 v.4 issue 2, pp. 59-75.
- [57] P. D. Signal, P. L. Lee, Relative Order Reduction of Multivariable Nonlinear Processes for Generic Model Control (GMC), *Chem. Eng. Comm*, 1995, Vol 132, pp. 187-206.
- [58] F. Allgower, Approximate Input-Output Linearization of Nonminimum Phase Nonlinear Systems, *Institute for Automatique*.
- [59] M. Duvall, J. B. Riggs, and P. L. Lee, Multi-model decoupled Generic Model Control, *Control Engineering Practice* 9, pp. 471-487, 2001
- [60] P. D. Signal and P. L. Lee, Generic Model Adaptive Control, *Chem. Eng. Comm.*, Vol. 115, pp. 35-52, 1992.
- [61] M. Ramseier, P. Agrawal, and D. A. Mellichamp, Non-linear adaptive control of fermentation processes utilizing a priori modelling knowledge, *Journal of Process Control*, pp.129-138, 1992.
- [62] E. Weyer, G. Bell, and P. L. Lee, System identification for generic model control, *Journal of Process Control* 9, pp. 357-364, 1999.

- [63] P. L. Lee, Direct Use of Nonlinear Models for Process Control, International Conference on Chemical Process Control (4th), pp. 517-542, 1991.
- [64] N. V. Joshi, P. Murugan, and R. R. Rhinehart, Experimental Comparison of Control Strategies, Control Eng. Practice, Vol. 5, No., pp.885-896, 1997.
- [65] M. Barolo, On the equivalence between the GMC and the GLC controllers, Computers. Chem. Engng Vol. 18, No. 8, pp.769-772, 1994.
- [66] B. V. Babu, and K. Jyotsanu, Temperature Control in an Exothermic Batch Reactor Using Generic Model Control and Globally Linearizing Control.
- [67] M. A. Henson and D. E. Seborg, A Critique of Differential Geometric Control Strategies for Process Control, IFAC 11th Triennial World Congress, Tallinn, Estonia, p.309-316, 1990.
- [68] G. N. Vanderplaats, *Numerical Optimization Techniques For Engineering Design*, 3rd Edition, 1999, Vanderplaats Research & Development, Inc.
- [69] W. Lu, Y. M. Zhang, and W.-Y. Lin, 2004. Nonlinear Interval Model Control of Quasi-keyhole Arc Welding Process. Automatica, 40(5): 805-813.
- [70] Y. Samyudia and P. L. Lee, 2002. Robust Generic Model Control (GMC) Design for Uncertain Nonlinear Processes. Proceedings of the 41st IEEE Conference on Decision and Control.
- [71] E. Walter, and L. Pronzato, Identification of Parametric Models from Experimental Data, Masson, Great Britain, 1997.
- [72] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Bounding Approaches to System Identification, Plenum Press, New York, p.367, 1996.
- [73] E. Hansen and G. W. Walster, Global Optimization Using Interval Analysis, Second Edition, Revised and Expanded, Marcel Dekker, New York, 2004.

VITA

Born 1977 in Winchester, Kentucky

Education:

University of Kentucky, Bachelor of Science in Mechanical Engineering

University of Kentucky, Master of Business Administration

University of Kentucky, Master of Mechanical Engineering

Professional Positions:

Inkjet Controls Lab at Lexmark International

Joseph Michael Istre