Cycle-to-Cycle Control of Multiple Input–Multiple Output Manufacturing Processes

by

Adam Kamil Rzepniewski

B.S., Mechanical Engineering
University of Notre Dame, 1999

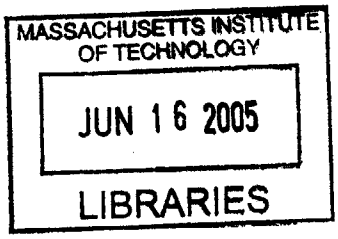
S.M., Mechanical Engineering
Massachusetts Institute of Technology, 2001

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2005



Signature redacted

Signature of Author .....
Department of Mechanical Engineering
May 5, 2005

Signature redacted

Certified by .....
David E. Hardt
Professor of Mechanical Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Lallit Anand
Chairman, Department Committee on Graduate Students

Cycle-to-Cycle Control of Multiple Input–Multiple Output Manufacturing Processes

By

Adam Kamil Rzepniewski

Submitted to the Department of Mechanical Engineering
on May 5, 2005 in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

ABSTRACT

In-process closed-loop control of many manufacturing processes is impractical owing to the impossibility or the prohibitively high cost of placing sensors and actuators necessary for in-process control. Such processes are usually left to statistical process control methods, which only identify problems without specifying solutions. In this thesis, we look at a particular kind of manufacturing process control, cycle-to-cycle control. This type of control is similar to the better known run-by-run control. However, it is developed from a different point of view allowing easy analysis of the process' transient closed-loop behavior due to changes in the target value or to output disturbances. Both types are methods for using feedback to improve product quality for processes that are inaccessible within a single processing cycle but can be changed between cycles. Through rigorous redevelopment of the control equations, we show these methods are identical in their response to output disturbances, but different in their response to changes in the target specification.

Next, we extend these SISO results to multiple input-multiple output processes. Gain selection, stability, and process variance amplification results are developed. Then, the limitation of imperfect knowledge of the plant model is imposed. This is consistent with manufacturing environments that require minimal cost and number of tests in determining a valid process model. The effects of this limitation on system performance and stability are discussed. To minimize the number of pre-production experiments, a generic, easily calibrated model is developed for processes with a regional-type coupling between the inputs and outputs, in which one input affects a region of outputs. This model can be calibrated in just two experiments and is shown to be a good predictor of the output. However, it is determined that models for this class of process are ill-conditioned for even moderate numbers of inputs and outputs. Therefore, controller design methods that do not rely on direct plant gain inversion are sought and a representative set is selected: LQR, LQG, and H-infinity. Robust stability bounds are computed for each design and all results are experimentally verified on a 110 input-110 output discrete-die sheet metal forming process, showing good agreement.

Thesis Supervisor: David E. Hardt
Title: Professor of Mechanical Engineering

ACKNOWLEDGMENTS

To my parents Zdzislaw and Helena Rzepniewski and Sister Eva who have patiently supported me throughout not only this venture, but all things that I have done in my life. Thank you.

I would like to thank my advisor David E. Hardt (Dave) who has spent a considerable amount of time in discussions with me, even when it was a Friday afternoon before a long weekend. Thank you for all of your support, you are the best advisor a student can have.

Thank you to my co-workers. Most specifically, thank you to Catherine L. Nichols who had started as a co-worker, but has become the love of my life. Thank you to Chip Vaughan and Bala Ganesan, who had made beach trips a regular event during the summer months. Thank you to Johnna Powell and Susan Brown with whom I had many conversations; some actually work related! Thank you to Benny Budiman who showed me how to graduate in style. Finally, thank you to my current co-workers Matt Dirckx, Grant Shoji, Kunal Thacker, and Wang Qi who had to put up with me during the last months of this dissertation.

CONTENTS

FIGURES

TABLES

CHAPTER

1

INTRODUCTION

The world of manufacturing has changed a lot since its beginnings. Production has gone from hand-manufactured parts, to machine manufactured parts for non-standardized assemblies, to part standardization and computer numerical control (CNC). In this thesis, we focus on a key technology within manufacturing that has emerged as a result of these changes: the application of feedback, or closed-loop, control to improve process output. The manufacturing control field continues to evolve as more sophisticated methods are developed and feedback is introduced at different points within the manufacturing cycle.

1.1 The Need for Control

The need for control has its roots in uncertainty. We simply do not know all of the input-output causal relations that exist both within and around the process. For example, a typical process may be subject to imprecise knowledge about the process itself or to outside disturbances such as vibration or temperature variations. It is true that one tries to minimize the disturbances either through advanced design, good knowledge of the process, or through control of the environment. However, the extent to which this can be accomplished is limited. For instance, a "true" model of a process is impractical, and usually impossible, and not *all* outside noise can be eliminated. This is especially true in

manufacturing environments where variations in workpiece properties are themselves a source of noise. As a result, process models are usually educated guesses and process outputs are generally described as distributions rather than point values. This is shown in Figure 1-1 where the output probability density is displayed as a function of deviation from target. As will be shown in subsequent chapters, control allows an operator to change these process distributions even in the face of uncertainty.

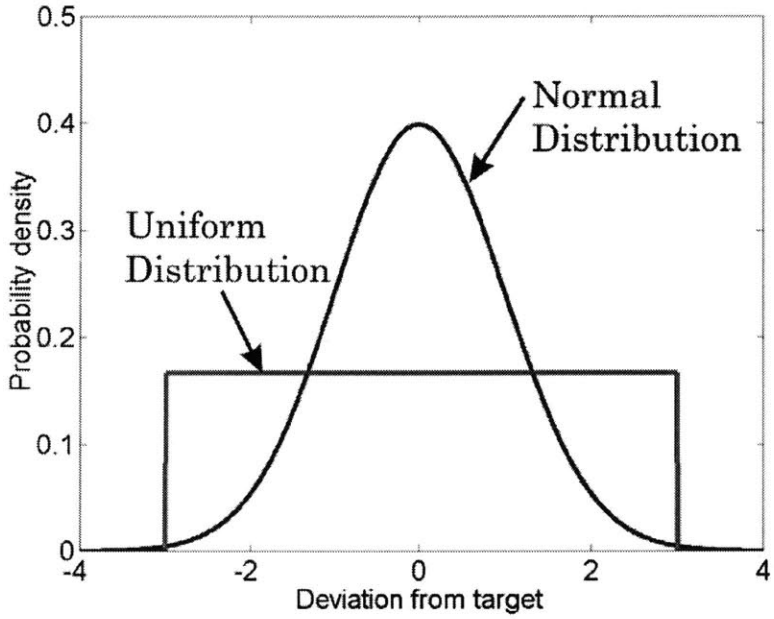


Figure 1-1 Example output distributions.

Simply changing a process is not enough; one must have a way of quantifying the benefits or drawbacks of the result. Measures of evaluating the goodness of a process, based on the idea of output distributions, have been developed by previous researchers, [1]. Here, we will discuss three of the most common: process capabilities C_p and C_{pk}, and the expected loss function $E[L(x)]$. Process capabilities are defined as:

$$C_p = \frac{USL - LSL}{6\sigma}$$

Equation 1-1

$$C_{pk} = \min\left(\frac{USL - \bar{x}}{3\sigma}, \frac{\bar{x} - LSL}{3\sigma}\right)$$

Equation 1-2

where *USL* and *LSL* are the upper and lower specification limits, respectively, $\bar{x}$ is the mean value of the output, and σ is its standard deviation. Note that a Gaussian (normal) output distribution has been assumed in both cases.

One can immediately notice the difference between these two measures. C_p focuses solely on the spread of the output, as compared to the specification limits, while C_{pk} penalizes both spread and non-centering of the output between the specification limits. This may lead to two different measures of goodness for the same distribution. As an example, consider two cases as shown in Figure 1-2. Both of these have unity standard deviation with specification limits at -3 and 3. However, they have very different mean values. In fact, the process on the right has 50% of its output outside the specification limits. C_p is unity for both while C_{pk} is dramatically affected by the difference between the two processes (unity for the left and zero for the right). Thus, both standard deviation and mean value should be considered when assessing the goodness of a process.

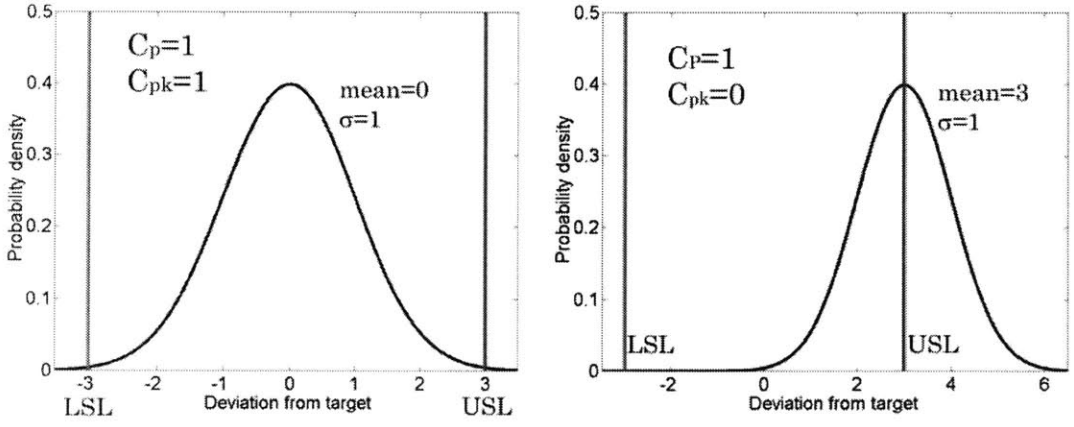


Figure 1-2 Example process distributions with specification limits; lower specification limit (LCL) and upper specification limit (USL). Centered distribution (left) and significantly non-centered distribution (right).

Expected loss is defined as [1]:

$$E[L(x)] = \int L(x) f(x) dx$$

Equation 1-3

where $L(x)$ is the loss function and $f(x)$ is the probability density function for the output, x. If a Gaussian probability distribution is assumed and $L(x)$ is a quadratic in the form of

$$L(x) = k(x - T)^2$$

Equation 1-4

where k is the loss coefficient and T is the target value, then the expected loss is calculated as:

$$E[L(x)] = k\left[\sigma^2 + (\bar{x} - T)^2\right]$$

Equation 1-5

Thus the expected loss function accounts for both the standard deviation and the mean of the output, as did C_{pk}. However, $E[L(x)]$ has two advantages over C_{pk}. First, it uses the coefficient k and the quality loss function to directly translate process statistics into something tangible, such as cost. Second, the expected loss remains continuous over the entire range of outputs, x. This continuity makes the expected loss function the best for comparing processes or calculating an optimum blend of mean shift and variance change.

We have stated, without proof, that control can change the process output distribution. We have also listed three performance metrics each with its benefits. However, we have not yet highlighted the real advantage of feedback: predictability. Rather than a haphazard approach to changing the process and post-calculating the improvement or detriment, closed-loop control gives *a priori* predictable results. Therefore, an operator can design a control law, or input recipe alteration routine, and have a good idea of what the result will be before applying it on a physical process. This ability is especially

important in manufacturing environments where the penalty for mistakes is usually monetary.

Uncertainty will, of course, still play a part. Models used in the control design process do not perfectly reflect reality and outside disturbances continue to exist. However, deviations in the model reflect themselves as changes in performance, the sensitivity to which can be minimized, and outside disturbance rejection is an explicit part of any good controller design process. Thus, when we consider the manufacturing environment with its performance metrics, the goal or motivation for applying control can be stated as: process improvement through mean centering on target and a reduction of the process output distribution. Expressed in terms of each performance function, the goals are to:

$$\max(C_p)$$

Equation 1-6

and/or

$$\max(C_{pk})$$

Equation 1-7

and/or

$$\min(E[L(x)])$$

Equation 1-8

1.2 Manufacturing Process Control: Control Methods

The work in this section has been largely presented in Rzepniewski, Hardt, and Vaughan, [2].

Because a number of control methods exist, we look at a representative set with an eye as to their drawbacks and benefits. The most widely used form of process control is statistical process control (SPC). Pioneered by Shewhart [3], the goal of this approach is to eliminate all sources of assignable causes. Assignable causes are those variations that

have a clear, or assignable, source and can therefore be eliminated. For example, a new operator changing the machine settings at a shift change is a source of variation that is easily eliminated through the use of standard operating procedures (SOP's). Once removed, the process is left only with those sources of random variation that cannot be assigned or removed. Thus, by definition, the process is at its minimum variation state. This state is called "in-control."

SPC is an open loop control method and is typified by run charts, X-bar charts, and S charts that an operator uses to confirm that the process is behaving like a stationary, random process. Although valuable as a process-monitoring tool, SPC methods do not propose a corrective course of action; it is often left to the factory worker or process engineer to find out what to do once an "out of control" signal has been registered on a control chart. Thus, they have no true "controller."

The second level of control is evolutionary operation (EVOP), [4]. This approach is a mix of design of experiments (DOE) and on-line optimization. An experimentally determined process model is continually adjusted and re-evaluated based on incoming data. The newest model is then used to identify the best operating point. Although promising in theory, the EVOP method is practically limited to three variables due to the geometrically growing number of experiments needed to update the models of larger processes, [4].

The final level of control is closed-loop control. This control method involves an *a priori*, explicit process model which is used to design a controller. The designer chooses a controller that gives the desired time response and stability properties for the closed-loop process. Stability is an important consideration since models are approximate and,

thus, do not perfectly reflect the true process. Three types of closed-loop control typically appear in manufacturing processes, as shown in Figure 1-3. Each type involves the process controller, but uses different feedback data.

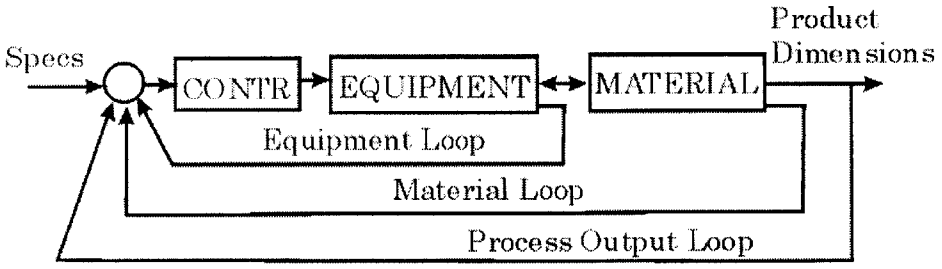


Figure 1-3 Three types of closed-loop control.

The first type, and the most common, involves the equipment loop. Here, the manufacturing machine is controlled to provide consistent (deterministic) energy states to the material. A typical example is a CNC machine. Here, the actuators are precisely controlled to produce a desired machine motion. Unfortunately, although the machine is aware of its own states, it does not have any information about the material states or properties. Thus, it cannot adjust its own state, e.g. the tool position of the CNC machine, to respond to varying conditions and workpiece states or properties, e.g. tool wear, material stiffness, etc.

The second type of closed-loop control involves the material loop. Here, the material states are fed back and this information is then used to infer things about the process output. Potential candidates for material loop feedback include material stress, strain, or temperature. For the CNC example, one could feed back the stress state of the workpiece through measurement of the tool force. This type of feedback, although potentially very useful, is rarely used owing to the difficulty and expense of obtaining and responding to data.

The final type of closed-loop control is process output control. This type is distinguished from material loop control by the *direct* feedback of the desired outputs, i.e. no inferences are needed. For the CNC example, this means a direct feedback of output geometry. Once again, this type of control is rarely used because of the cost and difficulty of its application. It is important to note that the three types of closed-loop control are not exclusive of each other and should be used together to produce the best output.

Out of the methods presented, closed-loop control is the most proactive and allows the user to predict and set the behavior characteristics of a process. It is also clear that anything other than output control neglects to consider process disturbances that occur outside the equipment and material loops. Thus, we will use this type of feedback control.

1.3 Problem Constraints

Having stated a general goal for control in manufacturing and defined our choice of control, we will begin to define our problem by imposing a few constraints. Our first constraint will be placed on the frequency of sensing and control action. We will assume that we can measure and change inputs only *after* a process cycle is over. This limitation may stem from the prohibitively high cost or the impossibility of sensing and actuating during the process cycle itself. This limitation is especially relevant for processes in which a particular part *during* manufacturing is different from itself *after* manufacturing. As an example, consider injection molding where a "hot" part (during manufacturing) will shrink as it is cooled and prepared for use (after manufacturing). In this case, assuming poor predictability of the shrinking behavior, dimension measurements during the process are not useful. Even if we assume good knowledge of shrinking behavior, placing actuators to take action based on this knowledge is difficult and expensive.

Control applied in this manner is called cycle-to-cycle (CtC) or run-by-run (RbR), the difference between these is the subject of Chapter 2.

The second constraint is on the problem form: we will concentrate on addressing the particular problems associated with Multiple Input-Multiple Output (MIMO) CtC processes. This is not because Single Input-Single Output (SISO) problems are uninteresting, but rather because SISO CtC and RbR processes have already received a lot of attention [5, 6, 7, 8, 9]. As part of this second constraint, we will narrow our MIMO outlook to processes with "spatial-type," limited coupling such that a single input influences a finite region of outputs. A 2-dimensional example of this is shown in Figure 1-4. Here, as single input, #3, affects all outputs, #1, #2, #3, #4, and #5. We also see that the magnitude of the influence decreases with increasing distance.

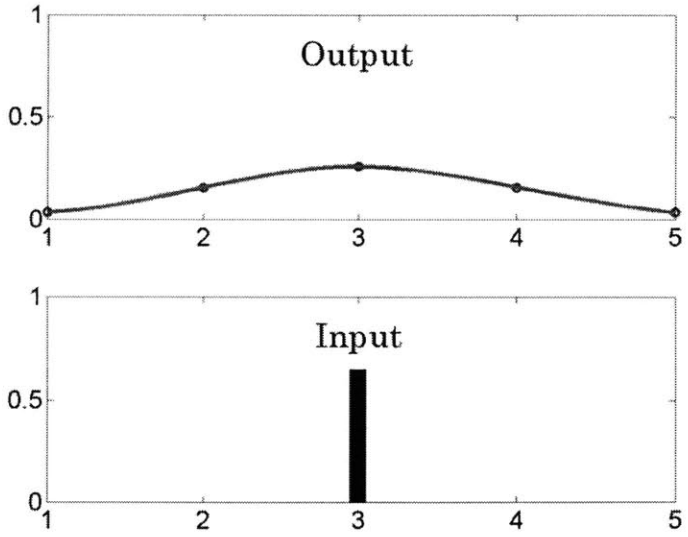


Figure 1-4 Spatially coupled output diagram. A single input at point 3 affects outputs 1-5.

Our last problem constraint has a manufacturing-inspired motivation: the process should require little calibration and should provide good, stable performance. Since we are working in a manufacturing environment, each piece of "data" is potentially a

product. Therefore, any pieces that are out of specification while we are trying to configure our controller or determine our process model, cost us in terms of lost time, lost materials, and lost profits. The requirement for good, stable performance is obvious. However, we will add that fast transient response (achieving process goals with very few runs or cycles) is desirable for parts with low-production numbers, e.g. job-shop environments, which have to cope with both outside disturbances as well as changing target specifications.

1.4 The CtC Control Problem

With our choice of type of control and a number of problem-defining constraints, we finally determine our CtC problem. Specifically, we look at the block diagram of our closed-loop process and the form of all CtC plant models, regardless of particular application.

1.4.1 The CtC Control Loop

Typically, processes include multiple control loops. These loops can be independent of each other or nested inside of other loops. For example, independent actuators in a CNC machine, with their independent controllers, each may be a part of a large motion coordination controller. Because we are using process output control, we will encounter both independent and nested controllers. Figure 1-5 shows a block diagram of such a configuration. Here, we see that "machine controllers and actuators" are a part of a larger "controller." Figure 1-5 also shows that all outside influences, such as temperature variations, are modeled here as output-additive disturbances.

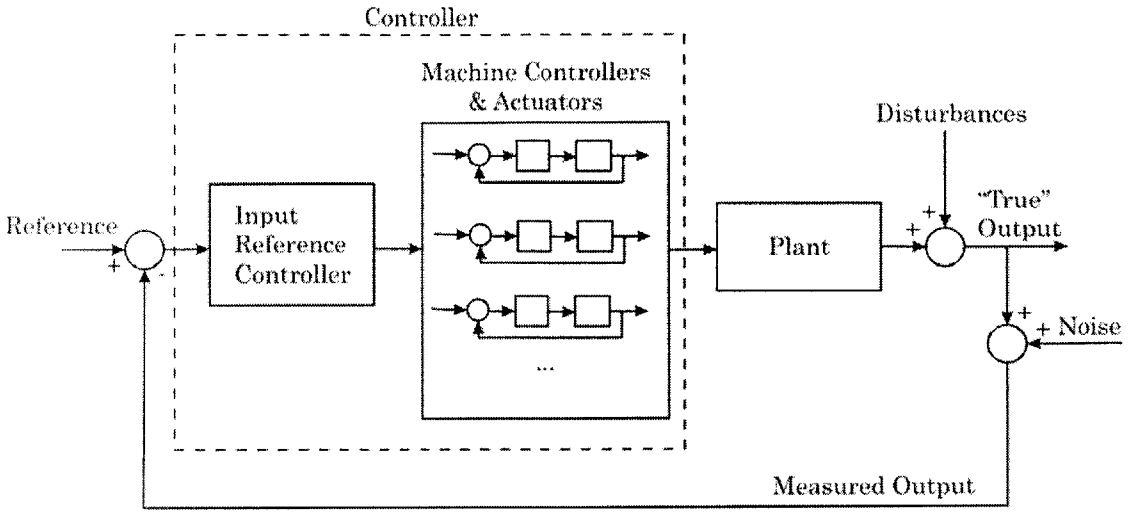


Figure 1-5 Expanded process block diagram for CtC control.

Although Figure 1-5 is a good depiction of reality, it remains quite complex. We will reduce this block diagram to that shown in Figure 1-6 through two assumptions. First, the "controller" is shown to encompass everything between the reference error and the input to the plant. As is shown in Figure 1-5, this means that any machine controllers and actuators are assumed to be part of the general controller. However, we will assume that the machine controllers and actuators faithfully obey the "input reference controller's" commands, see Figure 1-5. Error in this assumption may be modeled as an output additive disturbance. Note that this "faithfulness" assumption, coupled with the cycle-to-cycle control frequency, effectively removes all machine actuator loops from our view. Therefore, whenever we refer to the controller, we are referring to the "input reference controller" in Figure 1-5 and assuming perfect command following from the "machine actuators & controllers."

Second, the original disturbance and noise signals are collapsed into a single disturbance signal with the assumption that the known process output is the "measured" output rather than the "true" output, Figure 1-5. This last assumption is reasonable when

one considers that no information exists about the "true" output; we only know the, additionally noisy, "measured" output. To our benefit, measurement noise is usually orders of magnitude lower than the measurement signal.

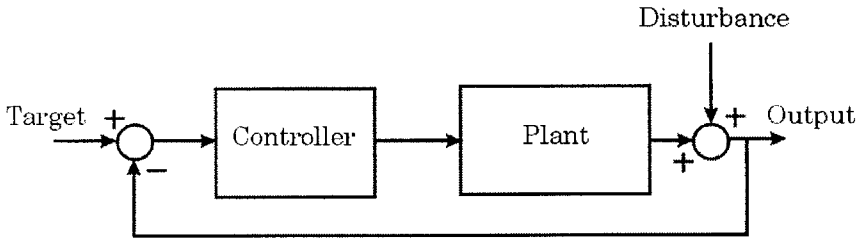


Figure 1-6 Simplified process block diagram.

1.4.2 The CtC Plant Model

We have previously defined cycle-to-cycle control as an approach in which control action takes place only between cycles. This simplification in control frequency has implications for the process model. By definition, a "cycle" is not finished until all of the dynamics have finished. As a result, with our assumed inability to observe during a cycle, the process is a black box; we know what we put in and what came out but we do not know what happened in-between. Therefore, the process dynamics are reduced to a delay element, which is required to maintain causality of the plant model. The delay element is described as $1/z$ by use of the z-transform, [10].

Additionally, the model will contain a gain describing the mapping of input and output changes. Most processes are nonlinear when one considers the full range of inputs. This nonlinearity is problematic, however, when one tries to apply classical control theory to design and analyze controller-plant combinations. In order to linearize our plant, we make a "small deviation" assumption; we assume that control action is taken in a region near the target rather than over the whole range. The effect of this is shown in Figure 1-7. Here, we see that a linear model encompassing the full range of the input, u, can do a

poor job of approximating the true process model. However, if we limit ourselves to operating within a small range δu, a linear model is typically sufficient even for a nonlinear process. The process model is

$$y - y_{operating} = \frac{1}{z} k_P \left(u - u_{operating}\right) \quad \Leftrightarrow \quad \Delta y = \frac{1}{z} k_P \Delta u$$

Equation 1-9

where k_P is the process gain and $y_{operating}$ and $u_{operating}$ are the output and input operating points about which we linearized. The errors resulting from our linearity assumption are modeled as additive disturbances, Figure 1-6.

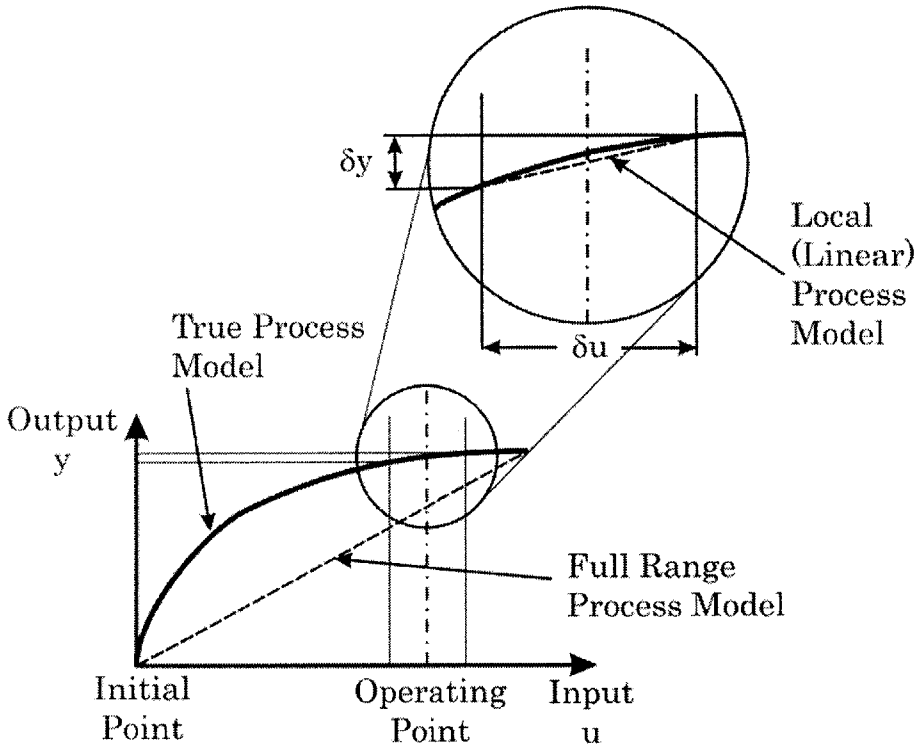


Figure 1-7 Process model linearization through a "small range" assumption.

1.5 Candidate Processes

In defining our problem and searching for ways to solve it, we have to constantly keep an eye on applicability. On the one hand, we want to address specific problems and prove them experimentally on physical processes. On the other hand, we have to remain general

enough to stay relevant to more than just one or two processes. In this section, we give eight examples of processes that lend themselves to cycle-to-cycle control and exhibit regional coupling, and can therefore benefit from the work presented in this thesis. We begin with a process that will become our experimental platform.

1.5.1 Discrete-Die Sheet Metal Forming

The work that led to this thesis was initially started as a short-term project to analyze a process which has existed within MIT's Manufacturing Process Control Laboratory (MPCL) for many years: discrete-die sheet metal stretch forming, [11]. Traditionally, stretch forming involves applying a stretch force to a workpiece and wrapping it around a monolithic forming tool, or die. A schematic of a stretch forming process is shown in Figure 1-8. A unique concept behind discrete-die stretch forming is dividing the traditionally monolithic forming die into many independent segments. These segments can then be rearranged between forming cycles to discretely approximate new target shapes or to eliminate disturbances.

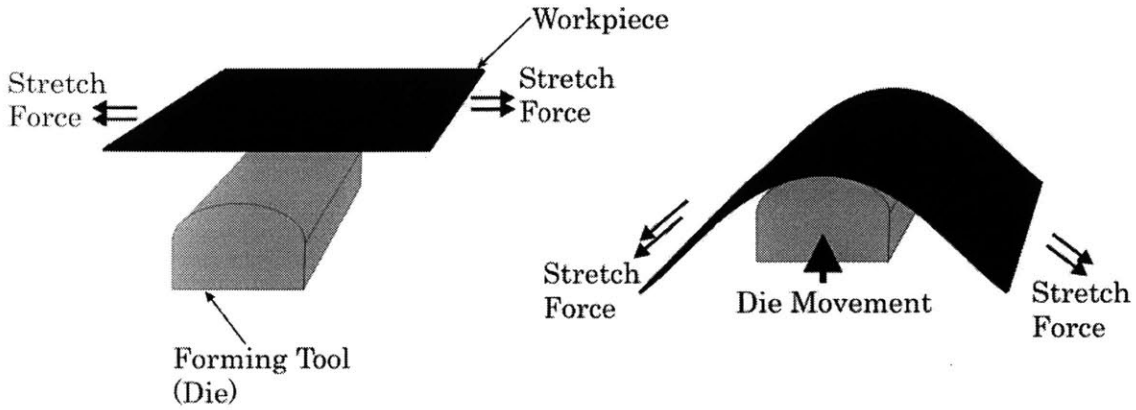


Figure 1-8 Schematic of sheet metal stretch forming.

Photos of MPCL's discrete-die sheet metal forming machine are shown in Figure 1-9. This laboratory-scale process has 24 rows and 23 columns of "pins" which make up the die. Thus the process has divided the monolithic die input into 552 inputs. Although all of these inputs can be used, only 10 rows by 11 columns of pins have been used to define the region of interest for over 8 years. For comparison of results, this is the scale on which we will perform any verifying experiments. A pre-production version of this reconfigurable tooling is shown in Figure 1-10.

The discrete-die sheet metal forming CtC problem may be defined as follows. The outputs are the locus of points which make up the final shape of a sheet metal workpiece. For ease of application, we will only consider those output points which are co-located with the center of each die pin. The inputs are the individual pin positions. A clear, causal relationship thus exists between the inputs and outputs. Because the sheet metal is continuous, each input will affect a region of outputs, as required by the problem constraints. The process, run in a CtC fashion, reduces to a delay and a gain matrix relating individual pin positions to specific points on the output shape.

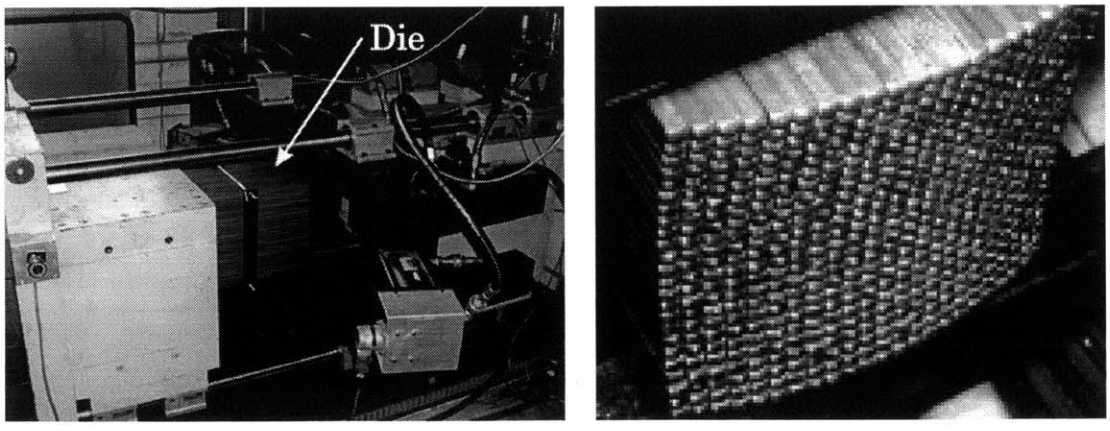


Figure 1-9 Manufacturing process control laboratory's stretch forming machine (left) and die surface detail (right). Die surface dimensions of 11.5 in x 12 in.

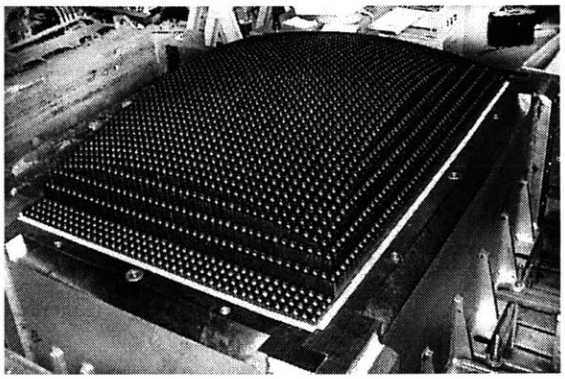


Figure 1-10 Reconfigurable tool produced by Northrop Grumman and Cyril Bath. The forming surface is 4 ft x 6 ft in area. (See Papazian, [12], for design details).

1.5.2 Fine Pressure Control for Hot Micro-Embossing

The idea of affecting regions of outputs may be extended to hot micro-embossing. In this process, a workpiece is heated beyond its glass transition temperature, into a "rubbery" state, and a patterned die is pressed into the softened material. The material is cooled before removal to maintain its patterned shape. Additional details may be found in de Mello, [13].

Working under the assumption that clamping pressure between the pattern and workpiece affects the final output shape, we can design a setup to vary the clamping pressure. This causal relationship assumption may be justified by performing pattern filling studies with different pressures. Rather than limiting ourselves to controlling just the gross, or overall, pressure, we want to also control the local pressure. This will allow us to compensate for changes in pattern shape or density. A schematic of such a design is shown in Figure 1-11.

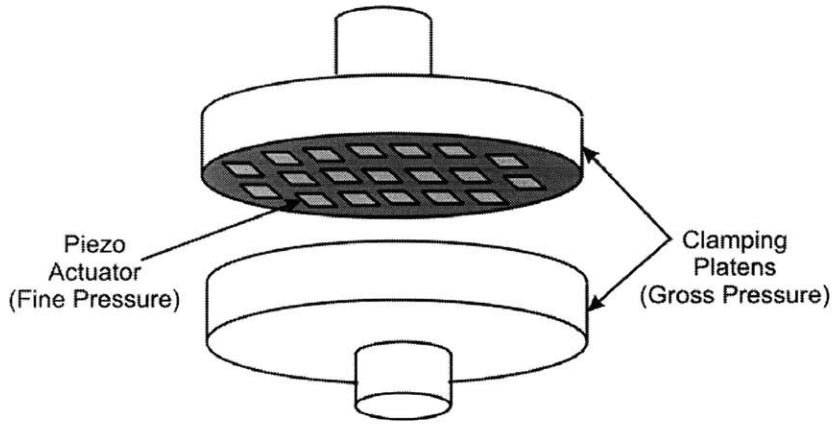


Figure 1-11 Schematic of a gross/fine pressure configuration for hot micro-embossing.

The process is thus described as follows. The outputs are the final part shapes of the workpiece. The inputs are the local pressures. Because the workpiece is not liquid, the outputs will couple according to the temporally-fixed pressure distributions from each input.

1.5.3 Temperature Control for Hot Micro-Embossing

A cause and effect relationship can also exist between the output shape and forming temperature for hot micro-embossing. As with pressure, we want to control the *local* temperature to be able to compensate for changes in pattern shape or density on a single workpiece. The physical setup for controlling local temperatures is similar in concept to local pressures, as shown in Figure 1-12. Here, a grid of local heaters is used to generate the required temperature distribution over the whole workpiece.

The process has the surface profile as its output and the local heater temperatures as its inputs. The outputs are coupled through the conductivity of the workpiece.

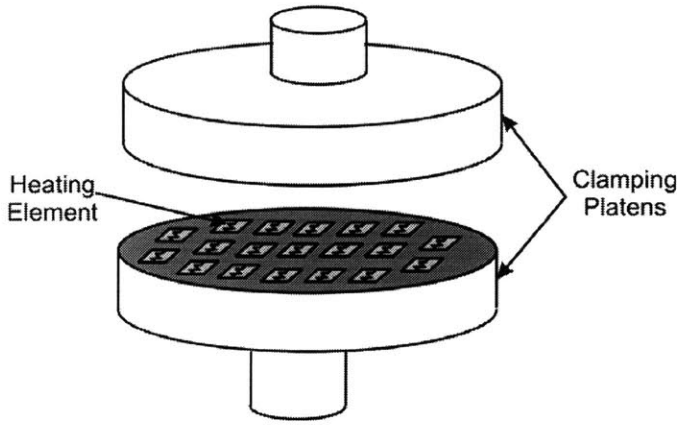


Figure 1-12 Schematic of a heating grid for local temperature in hot micro-embossing.

1.5.4 Binding Pressure Control for Deep Drawing

Deep drawing is a process where a punch forces a flat workpiece into a die cavity. The process involves extensive plastic deformation as the workpiece is restrained at the edges by the pressure between the binding ring and machine structure, as shown in Figure 1-13 (left). Common products of this process include soda cans, pots and pans. Additional details on this process may be found in Kalpakjian, [14].

An important variable in deep drawing is the binding pressure. Under- or over-application can cause wrinkling of the final product. This is especially problematic for deep drawing of non-symmetric products. An idea that fits within our problem definition and lends flexibility to the process is to divide the binding ring into independently-actuated segments, as shown in Figure 1-13 (right). The CtC process may then be defined with the extent of local thinning or wrinkling as the output and the pressure at each binding ring section as the input. The continuity of the sheet among all the inputs will couple the outputs.

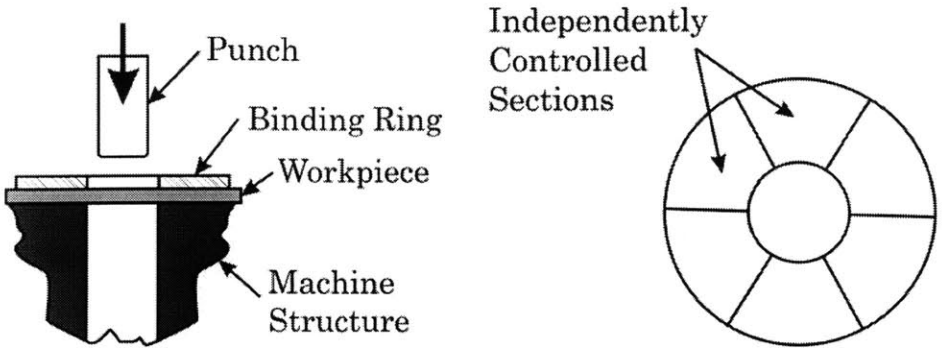


Figure 1-13 Deep drawing process schematic (left), Segmented binding ring (right).

1.5.5 Control of Robotic Fixtures for Manufacturing

Some manufacturing processes require robotic fixtures for precise positioning of the workpiece. These fixtures often over-constrain the three position and three rotational degrees of freedom, resulting in a deflection of the part. In this instance, the structure of the part will couple the effects of actuators which may be located far away from each other. This coupling can be described as spatial with more remote actuators having a lower influence on an output. As an example of this, consider the problem of spot welding a car door panel, Figure 1-14. The challenge is to locate a point of interest, the weld spot, in 3D space. Because of its size and flexibility, the workpiece requires extra support and four actuators are used. These actuators over-constrain the position of the workpiece. As such, small movements in their relative position can be used to alter the relative positions of the points of interest through deflection of the workpiece itself.

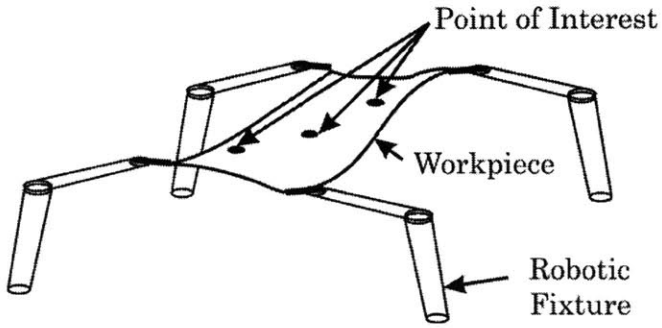


Figure 1-14 Robotic fixtures with sheet workpiece; car door welding example.

1.5.6 Pressure Control in Chemical-Mechanical Polishing

Chemical-mechanical polishing (CMP) is a common process in semiconductor manufacturing. The workpiece to be polished is held to a rotating pressure pad which may be used to provide both positive and negative pressure. It is then brought in contact with a rotating polishing pad which removes material from the workpiece through a combination of caustic agents and physical force. Additional details on this process may be found in Moyne *et al.*, [15].

The CMP process may be adopted to use the methodology contained in this thesis through a small alteration of the top pressure pad. If it is divided into sections, the top pressure pad may be used to increase or decrease the wear rate on a local area of the workpiece. The effects of different neighboring supply pressures would be coupled though the workpiece structure.

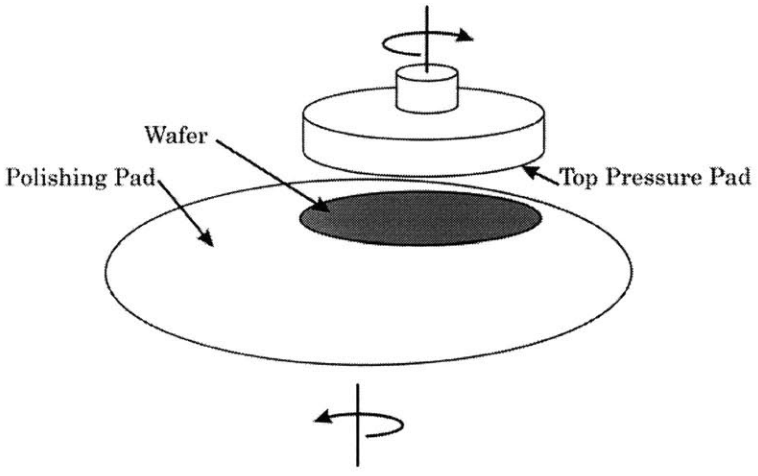


Figure 1-15 Schematic of chemical-mechanical polishing.

1.5.7 Control of Chemical Vapor Deposition

Chemical vapor deposition (CVD) is a material process. Normally, a workpiece is placed within a reaction chamber and a reactive chemical vapor is introduced. The chemical vapor then interacts with the workpiece surface such that a solid film is produced. Additional details may be found in Kalpakjian, [14]. The reactivity of the gas with the surface as well as the reaction rate are functions of the surface temperature. Therefore, if we add the ability to control the temperature locally, we will be able to control the local amount of film that is produced. Naturally, conduction within the workpiece and convection in the reaction chamber will spatially couple the outputs. The CVD process, thus altered, can benefit from the work in this thesis.

1.5.8 Etching Control

As a final example, consider the control of wet etching. This process, as with CMP and CVD, is popular in semiconductor manufacturing. Here, the workpiece is placed in a chemical bath which selectively reacts with the surface to remove material, [14]. Because it is a chemical process, it has a notable dependence on the local concentration of reactant. Therefore, we can control the local reaction rate through the local control of concentration. To achieve this, we imagine an array of micro-pumps which can both expel as well as absorb a small amount of reactant. The effect of any single pump would be spatially distributed among the outputs (surface geometry) through diffusion within the chemical medium.

1.6 Thesis Organization

This chapter served as a brief introduction to the work we will discuss in this thesis. First, we motivated the use of feedback control to improve the output of manufacturing

processes. To do this, we showed three popular performance metrics which allow us to compare process outputs based on statistics. Additionally we stated that feedback control will allow us to predictably change those statistics. Next, we reviewed a number of control methods, focusing on output feedback closed-loop control. Then, we imposed a number of restrictions on the types of processes we wish to address. We confined ourselves only to changing control action between cycles, decided to address MIMO processes, and placed an emphasis on easy calibration of the process model and a requirement on stability in the face of uncertainty. We followed-up the constraints with a more thorough definition of the CtC control loop. Finally, we named a number of processes which can benefit from the application of cycle-to-cycle theory presented in this thesis.

The remaining chapters will build on this foundation. Chapter 2 looks at the application and the differences between SISO run-by-run and cycle-to-cycle control. Through rigorous redevelopment of the control equations, we see that the two methods address the same problem and differ in their solution only in their response to changing target specifications. These SISO results are used to develop an ideal extension of CtC control theory to multiple input-multiple output processes in Chapter 3. We conclude that we only need to know the process model to be able to design a controller. Chapter 4 looks at experimental attempts to identify the specific process model for MIMO sheet metal forming. These attempts are not successful and a simulation-based approach is taken to identifying the uncertainty tolerance of a $2x2$ example MIMO process. Encouraged by the robustness results of Chapter 4, Chapter 5 presents a general, easily calibrated model which can be used on any of the processes that meet our problem constraints. We

properly account for uncertainty in Chapter 6 and present the results of applying three MIMO controllers to the sheet metal forming process, Section 1.5.1, in Chapter 7. Finally, conclusions and future work are discussed in Chapter 8.

CHAPTER

2

EWMA, DOUBLE EWMA AND INTEGRAL CONTROL

Before we can address the issues in MIMO cycle-to-cycle control, we need to understand the equations and assumptions that make up single input-single output run-by-run and cycle-to-cycle control approaches. To do this, we will look at them from the point of view of feedback control theory, focusing on time responses and errors. This is done in contrast to other views which focus on the noise transmission properties. It will become apparent that the feedback control perspective brings to light some assumptions made in the process and control equations, and reduces the number of variables to be considered.

This chapter is divided into two main sections, both concerned only with SISO control. We will not begin to consider MIMO problems until Chapter 3. The first section of this chapter addresses exponentially weighted moving average (EWMA) -based control and integral control. These controllers are meant for processes subject to little or no ramp disturbance. The second section addresses the choice of controllers for processes which are subject to a measurable drift, or ramp disturbance, considering double-EWMA and double integrator-zero controllers in particular. Although this thesis focuses mainly on the application of cycle-to-cycle control to MIMO processes with little or no drift,

because of our choice of experimental platform, Section 2.2 serves as a significant primer for designing MIMO controllers which are able to completely eliminate drift disturbances. In fact, Section 2.2 shows that the design of even SISO controllers is not well understood.

2.1 EWMA and Integral Control

A significant body of work exists for run-by-run processes that are not subject to a ramp disturbance, drift; we will look at these first. The controllers fall into two classes, a single exponentially weighted moving average and an integral controller. We will review their development and the assumptions under which these controllers are equivalent.

2.1.1 Sachs, Hu, and Ingolfsson

Sachs *et al.* [5] presents the following development for a single input-single output process. The true process is governed locally by the input-output (u-y) model

$$y_t = \alpha_t + \beta_t u_t + \varepsilon_t$$

Equation 2-1

where α is the process intercept, β is the process sensitivity, and ε represents noise. It is assumed that ε does not include variation in α and β and that the parameters α and β may themselves be random variables. The process intercept and sensitivity may also change with time. Based on assumptions made about the probability distributions of the above terms [5], one can write the prediction for the process output, $\hat{y}_t$, as:

$$\hat{y}_t = \hat{a}_{t-1} + \hat{b}_{t-1} u_t$$

Equation 2-2

where $\hat{a}$ and $\hat{b}$ are estimates of the process intercept and process sensitivity, respectively.

At this point, we will make a small adjustment to the process equation listed by Sachs *et al.* As one can observe, there does not appear to be a delay element between the input, u, and output, y, when one is required for causality. We believe that this may be a difference between the nomenclature used in this thesis and that used by Sachs *et al.* For us, output at t=2 is the result of input at t=1. It appears that for Sachs *et al.*, t indicates the run number, so that output at t=2 is the result of input at t=2 with the understanding of an existing time (but not run) delay; this is only a different way of counting the inputs and outputs. For consistency with the rest of this thesis, we will take the process equation to be:

$$\hat{y}_t = \hat{a}_{t-1} + \hat{b}_{t-1} u_{t-1}$$

Equation 2-3

Within the context of run-by-run gradual mode control in [5], the process sensitivity, $\hat{b}$, is assumed to stay constant. The process intercept is then adjusted "in an intelligent way," [5], to keep the output on target. Both Muth, [6], and Box and Jenkins, [7], show that the exponentially weighted moving average provides a minimum mean squared error (MMSE) forecast when the process noise, ε, is governed by the integrated moving average (IMA(1,1)) noise model. This is shown in Appendix A. EWMA may be equivalently defined in two forms:

$$\hat{a}_t = \sum_{i=1}^{t} \tilde{w}(1-\tilde{w})^{t-i}\left(y_i - \hat{b}u_{i-1}\right)$$

Equation 2-4

or via the recursive relation:

$$\hat{a}_t = \tilde{w}\left(y_t - \hat{b}u_{t-1}\right) + (1-\tilde{w})\hat{a}_{t-1}$$

Equation 2-5

where $\tilde{w}$ is the weight placed on the most recent data.

The process input, or recipe, may be adjusted by setting the process output, $\hat{y}_t$, equal to the target, T, and calculating the input setting:

$$u_{t-1} = \frac{T - \hat{a}_{t-1}}{\hat{b}}$$

Equation 2-6

Equation 2-5 and Equation 2-6 may be combined to show more clearly the true form of the process controller. Substituting Equation 2-6 into Equation 2-5 yields:

$$\hat{a}_t = \hat{a}_{t-1} + \tilde{w}(y_t - T)$$

Equation 2-7

Writing Equation 2-6 for two consecutive cycles and using Equation 2-7 yields the final form of the controller, [5],

$$u_t - u_{t-1} = -\frac{\tilde{w}}{\hat{b}}(y_t - T)$$

Equation 2-8

or alternatively:

$$u_t = -\frac{\tilde{w}}{\hat{b}}\sum_{i=1}^{t}(y_i - T) + u_1$$

Equation 2-9

Because of its single summation of deviations from target (errors), this controller is an integral controller.

2.1.2 Box and Luceno

Box and Luceno follow a similar development, [8]. The process model takes the form

$$y_t = g u_{t-1}$$

Equation 2-10

where g is the process sensitivity, or gain. It is assumed that the model perfectly describes the process and any deviation from the expected output is caused by a disturbance, d.

Therefore, if it were possible to know the future value of the disturbance, the recipe could be adjusted to perfectly cancel it out:

$$gu_t = -d_{t+1}$$

Equation 2-11

Since no such knowledge exists, a prediction of the future disturbance must be used:

$$gu_t = -\hat{d}_{t+1}$$

Equation 2-12

The adjustment equation shows that the error at the output would thus be the error in predicting the disturbance rather than the disturbance itself:

$$\tilde{e}_{t+1} = d_{t+1} - \hat{d}_{t+1}$$

Equation 2-13

Acknowledging its minimum mean squared error property for IMA(1,1) noise, see Appendix A, Box and Luceno, [8], use the exponentially weighted moving average to predict the future value of the disturbance such that:

$$gu_t = -\tilde{d}_t$$

Equation 2-14

Again, the EWMA relation may be written as

$$\tilde{d}_t - \tilde{d}_{t-1} = \tilde{w}\left(d_t - \tilde{d}_{t-1}\right)$$

Equation 2-15

Combining Equation 2-14 and Equation 2-15 gives the desired forms for the controller:

$$u_t - u_{t-1} = -\frac{\tilde{w}}{g}\tilde{e}_t = -\frac{\tilde{w}}{g}\left(d_t - \tilde{d}_{t-1}\right)$$
$$= -\frac{\tilde{w}}{g}\left(y_t - T\right)$$

Equation 2-16

or

$$u_t = -\frac{\widetilde{w}}{g}\sum_{i=1}^{t}\widetilde{e}_i + u_1 = -\frac{\widetilde{w}}{g}\sum_{i=1}^{t}(y_i - T) + u_1$$

Equation 2-17

As before, this is an integral controller.

2.1.3 Hardt and Siu

A different approach towards designing the controller has been taken by Hardt and Siu, [9].

Like prior researchers, Hardt and Siu, [9], assume a gain-and-delay process model such that

$$y_t = k_p u_{t-1}$$

Equation 2-18

where k_p is the process gain. Any differences between this model representation and reality are assumed to take the form of an additive output disturbance. Note that we have chosen to leave out the operating points $y_{operating}$ and $u_{operating}$, Equation 1-9, which locate the region of relevance for the plant model. Equation 2-18 may be rewritten using the z-transform, [10], as:

$$y = \frac{k_p}{z}u$$

Equation 2-19

With this representation, it is possible to use z-plane analysis, root-locus, to design the appropriate controller. To do this, we write the closed-loop transfer function describing the reference input-process output relationship of the feedback configuration shown in Figure 2-1. Then, we look at both the zeros, values of z at which the transfer function becomes zero, and poles, values of z at which the transfer function is singular, to design our closed-loop process. Simple rules for constructing a root locus, a picture of how the

poles move with different controller gains, may be found in most classical controls books such as Kuo, [10]. The fact that the root locus displays all possible design points for a controller-plant pair makes it a powerful design tool.

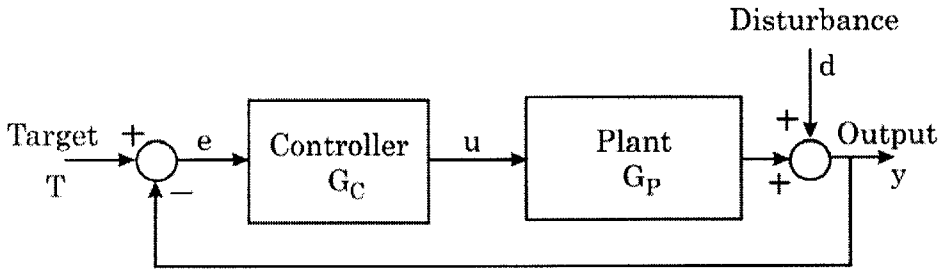


Figure 2-1 Process block diagram.

The stable region of the z-plane is described by a unit-radius circle as shown in Figure 2-2. Various ranges of transient-response performance can be achieved by placing the closed-loop process poles within this region. Infinite settling time is achieved at a radius of 1 and reduces with radius until one-cycle settling time is achieved at the circle's center, Figure 2-2. The settling time, t_s, may be calculated using the definition of the z-transform. A well accepted approximation for settling to within 2% of the input is

$$t_s = \frac{4}{\sigma_s}$$

Equation 2-20

where σ_s is the real-axis coordinate in the s-plane, [10]. The homogenous, or unforced, response settling time, in terms of cycles, becomes:

$$t_s = \frac{-4}{\ln(r)}$$

Equation 2-21

where r is the distance from the origin of the z-plane. This equation is defined for $0 \le |r| \le 1$.

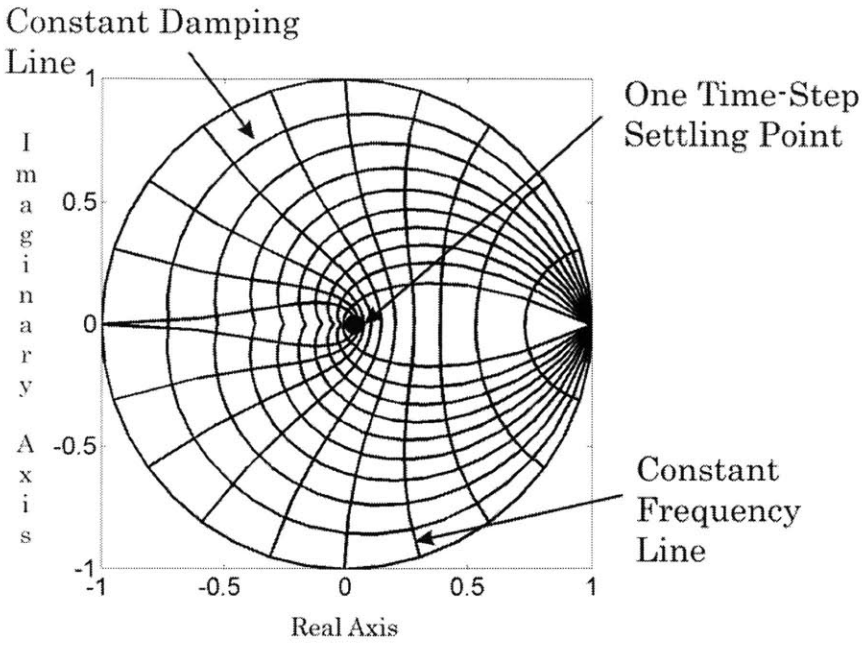


Figure 2-2 z-plane stable region with constant damping and constant frequency lines, [10].

In addition to speed, the location of the closed-loop process poles determines the *type* of response. Non-oscillatory response can be obtained only when all closed-loop poles reside on the real axis between 0 and 1, i.e. [0,1). Oscillations in the response will be present when any closed-loop poles are located anywhere else within the unit circle. Since oscillations involve overshoot, these pole locations may be unacceptable in certain applications, e.g. chemical-mechanical polishing (CMP) where excessive material removal may not be tolerated. For those applications which can tolerate oscillation, Figure 2-2 shows constant damping and frequency lines which can help the designer choose an acceptable transient response, [10].

In addition to the short-term performance, settling time and transient response, one must look at the long-term performance. The final value theorem may be used for this analysis, [10]. The general form of the final value theorem is

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{z\to 1}\left(1 - z^{-1}\right)e(z) = \lim_{z\to 1}\frac{z-1}{z}e(z)$$

Equation 2-22

where e_{ss} is the steady state error and $e(z)$ is the error transfer function. The error transfer function may be written from the block diagram in Figure 2-1 for both "inputs" T and d.

$$\frac{e}{T} = \frac{1}{1 + G_p G_c}$$

Equation 2-23

$$\frac{e}{d} = \frac{-1}{1 + G_p G_c}$$

Equation 2-24

Note that the error's response to the reference input, T, is just the negative of its response to the disturbance. Also, noting that the plant transfer function, from Equation 2-18, is

$$G_p = \frac{y}{u} = \frac{1}{z}k_p$$

Equation 2-25

and using a step input of size c

$$step = \frac{z}{z-1}c$$

Equation 2-26

the steady state error becomes

$$e_{ss} = \lim_{z\to 1}\frac{z-1}{z}\left(\frac{1}{1 + \frac{k_p}{z}G_c}\right)\left(\frac{z}{z-1}c\right) = \lim_{z\to 1}\frac{cz}{z + k_p G_c}$$

Equation 2-27

Therefore, the controller transfer function must have a (z-1) term in the denominator for the steady state error to go to zero from a target or disturbance step input.

It is with these tools of settling time, response characteristics, and steady-state performance that Hardt and Siu, [9], design their controller:

1. According to Equation 2-21, the best settling time is achieved when the closed-loop poles are located at the origin of the z-plane.
2. Non-oscillatory response is achieved on [0,1) of the real axis; the chosen controller should have access to at least parts of this range.
3. The controller transfer function should have a (z-1) in the denominator to eliminate steady-state error.

The commonly-used integral controller is able to satisfy all three of these guidelines:

$$G_c = \frac{u}{e} = \frac{k_c z}{z-1}$$

Equation 2-28

where k_c is the controller gain. When Equation 2-28 is written as a summation, it becomes clear that it is similar to both Equation 2-9 and Equation 2-17

$$u_t = k_c \sum_{i=1}^{t} e_i = k_c \sum_{i=1}^{t} (T_i - y_i)$$

Equation 2-29

2.1.4 EWMA and Integral Control

Having reviewed the development of run-by-run and cycle-to-cycle control equations, let us consider both the differences in approach for obtaining these solutions as well as the differences in the solutions themselves. However, first let us further justify the adjustment to the process equation of Sachs *et al.*, [5], with the knowledge we have gained by developing the two subsequent approaches. The closed-loop transfer function obtained while keeping the original (no delay) model is

$$\frac{y}{T} = \frac{\frac{\beta}{\hat{b}}(z-1+\tilde{w})}{z-1+\frac{\beta}{\hat{b}}\tilde{w}}$$

Equation 2-30

It can be seen that the transfer function becomes unity for the special case of β/b=1, i.e. getting the process gain correct. This has two consequences. First, the output will perfectly follow the target with no delay, an impossibility that would make the system non-causal. Second, the transient response of the output will be completely independent of the EWMA weight $\tilde{w}$. This is impossible when one compares the results of Sachs *et al.*, [5], to the final equations of both Box and Luceno, [8], and Hardt and Siu, [9]. The closed-loop transfer function for the adjusted model (with delay) is

$$\frac{y}{T} = \frac{\frac{\beta}{\hat{b}}\left(z\tilde{w} + z - 1\right)}{z\left(z - 1 + \frac{\beta}{\hat{b}}\tilde{w}\right)}$$

Equation 2-31

This remains consistent with the other developments at all EWMA weights and process gain estimates. Thus, frequency domain analysis motivates our modification to the Sachs *et al.* process equation.

Comparing the development of the controllers, one sees that both Sachs *et al.*, [5], and Box and Luceno, [8], develop their controllers as a means of controlling the spread of the output distribution, as is illustrated in Figure 1-1. This is done by assuming a model of the disturbance noise, IMA(1,1) in this case, and selecting the controller that gives minimum mean squared error, which is the exponentially weighted moving average controller. Hardt and Siu, [9], on the other hand, do not mention the spread of the output distribution in performing their design. Their controller is based on eliminating the difference between the target value and the mean of the output distribution, $e \to 0$, caused by both target and disturbance inputs. It is interesting to note that these different

starting points and different methodologies have arrived at such a similar conclusion for the form of the controller.

It is frequently said that the exponentially weighted moving average and the integral controller are one and the same. To see this, or rather to see under what conditions this is true, we have gone through the task of redeveloping the equations. Here, we repeat the three conclusions:

Sachs *et al.*, [5], Equation 2-9: $u_t = -\frac{\widetilde{w}}{\hat{b}}\sum_{i=1}^{t}(y_i - T) + u_1$

Box and Luceno, [8], Equation 2-17: $u_t = -\frac{\widetilde{w}}{g}\sum_{i=1}^{t}(y_i - T) + u_1$

Hardt and Siu, [9], Equation 2-29: $u_t = k_c\sum_{i=1}^{t}(T_i - y_i)$

Note that the term u_1 does not appear in Hardt and Siu's control equation because the model is assumed to be around $u_{operating}$ with $u_{operating} = u_1$.

It is clear that Sachs *et al.* and Box and Luceno have arrived at exactly the same conclusion. Hardt and Siu however, have arrived at a slightly different one. Careful comparison of all three equations reveals that the integral controller equation has a subscript i on the target value, T, while the other two equations do not. The meaning of this is that integral control does not require a constant target while Equation 2-9 and Equation 2-17 can only be derived while assuming a constant target value. The difference between the two approaches is seen more clearly when looking at the block diagrams in Figure 2-3 and in Figure 2-4. Here, we see that in order to eliminate the constant target assumption for EWMA-based control, we have to include a feed forward loop. With a constant target assumption however, both controllers are equivalent.

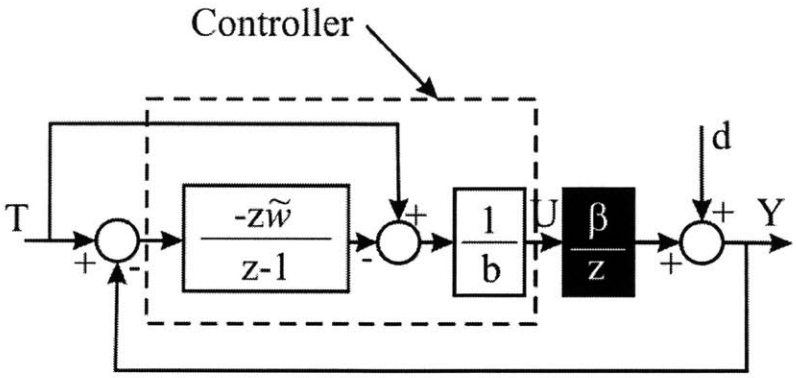


Figure 2-3 EWMA-based control block diagram.

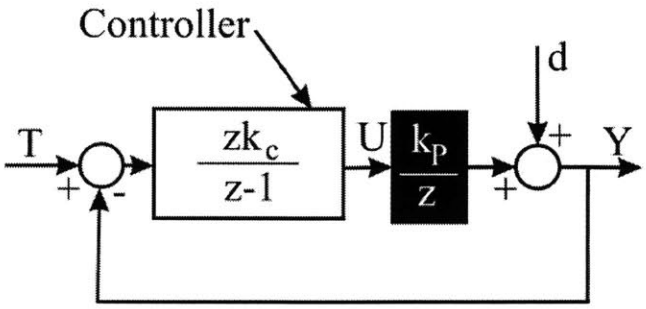


Figure 2-4 Integral-based control block diagram.

Figure 2-3 and Figure 2-4 also show how the EWMA and Integral controllers are equivalent when looked at through the disturbance input. To see this, we apply the principle of superposition for linear systems which, restated, says that the output is the sum of the independent inputs multiplied by their transfer functions. Since we are looking at the disturbance input *d* only, we ignore the target input and quickly see that the block diagrams become exactly the same with the integral control gain equal to the ratio of EWMA weight and process gain estimate for the EWMA-based controllers.

2.1.5 Selecting Integral Controller Gains

With an understanding of the controller forms and origins, we turn our attention to the task of choosing the control gain k_c. Note that the feedback control perspective had reduced the problem to choosing the value of one gain, rather than two apparent gains $\tilde{w}$

and b or g. Our choice of control gain is expected to impact both the average value as well as the distribution of the output. We will consider both in turn.

2.1.5.1 Mean Output Properties

We look at the time behavior of the mean output value by examining the root locus. As previously mentioned, the root locus provides guidance in choosing the control gain by displaying all possible design points at once. The root locus for the integral controller and CtC plant is shown in Figure 2-5. Note that, although there are actually two poles and one zero, a perfect pole-zero cancellation occurs and only a single pole changes location with gain.

As discussed by Sachs *et al.* [5], and Hardt and Siu, [9], a stable output response may be obtained when the loop gain, the product of the controller and plant gains, is on the interval (0,2), i.e. all closed-loop poles remain within the unit circle. The closed-loop process remains non-oscillatory when the value of the loop gain is on the interval (0,1]. It is therefore better to use less than unity loop gain if the process gain is not known perfectly and an oscillatory response is unacceptable. These results are summarized in Table 2-1. In order to satisfy the one time-step settling guideline, the product of the control and plant gains, i.e. the loop gain, must be exactly unity. However, as predicted by Equation 2-21, two-cycle settling times may be achieved when the closed-loop pole remains near to the origin, i.e. the loop gain is merely "close" to unity.

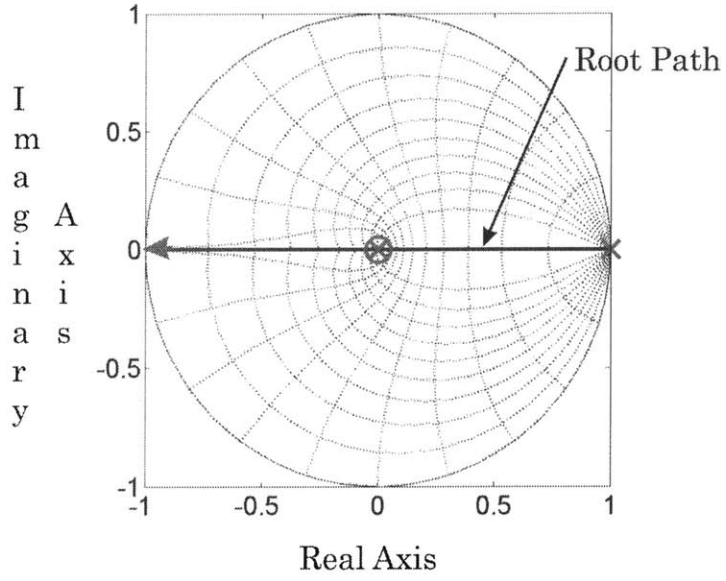


Figure 2-5 Integral controller root locus.

Region #	Location of Pole on the Real Axis	Process Loop Gain $K=k_pk_c$	Expected Performance
1	$(\infty,1]$	$(-\infty,0]$	Unstable
2	(1,0]	(0,1]	Stable, No oscillation
3	(0,-1)	(1,2)	Stable, Oscillatory
4	$[-1,-\infty)$	$[2,\infty)$	Unstable

Table 2-1 Expected process performance as a function of loop gain, k_pk_c for an integral CtC controller.

2.1.5.2 Variance Change

The use of closed-loop feedback changes the noise transmission properties of a process. When subject to CtC control, inherent process noise can be either attenuated or amplified depending on the degree of correlation of the noise process, therefore changing the process output distribution, Figure 1-1. Since closed-form solutions are affected by the form of the noise, let us consider two noise models, IMA(1,1) and simple, correlated noise.

Sachs *et al.*, [5], and Box and Luceno, [8], both use the integrated moving average noise model, IMA(1,1), which may be described by

$$d_t - d_{t-1} = a_t - \theta a_{t-1}$$

Equation 2-32

where a are normal, identically distributed, independent (NIDI) "shocks" and θ is a smoothing constant. Hardt and Siu, [9], on the other hand, use a correlating filter of the form

$$d_t - Pd_{t-1} = (1-P)a_{t-1} \quad \Leftrightarrow \quad d = \frac{1-P}{z-P}a$$

Equation 2-33

where P is the relative weight of the previous process disturbance and a new NIDI component. P is only defined on [0,1].

The similarity of the two noise models was demonstrated by Rzepniewski *et al.*, [2]. Because a are all NIDI with a mean of 0 and a variance σ^2, it is possible to define a new noise term

$$a'_{t-1} = a_t - \theta a_{t-1}$$

Equation 2-34

where the new term's mean will still be 0, and the new variance is $\sigma'^2 = \sigma^2(1+\theta^2)$. The new definition of noise may be substituted into the IMA(1,1) noise model to obtain

$$N_t - N_{t-1} = a'_{t-1} \qquad \Leftrightarrow \qquad N = \frac{1}{z-1}a'$$

Equation 2-35

Thus, the two models have a similar form. However, significant differences between the two models do remain. The integrated moving average model, as the name states, does not have a constant mean. In fact, the IMA model does not have a finite mean as t is allowed to go to infinity. Conversely, the correlating filter presented in Equation 2-33 does has a constant mean and its output, d, remains finite even as t goes to infinity. These effects may be seen in Figure 2-6.

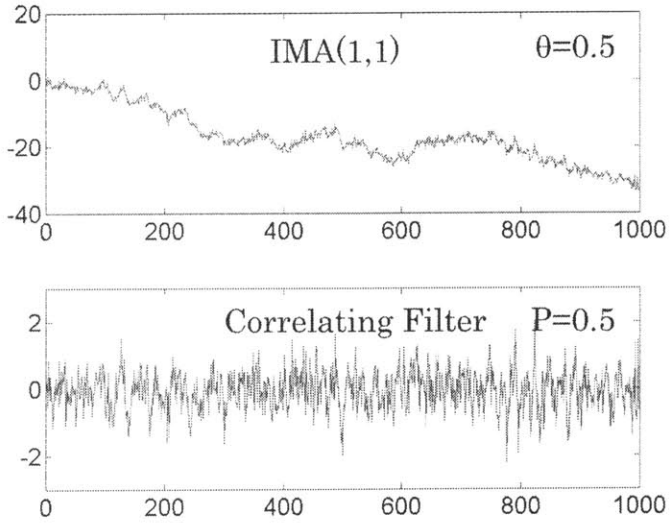


Figure 2-6 IMA(1,1) and correlating filter (Equation 2-33) outputs with *a* defined by N(0,1).

As previously mentioned, Muth, [6], and Box and Jenkins, [7], have shown that the EWMA-based controller is the minimum mean squared error controller for IMA(1,1) noise. Because it has a different form, the correlation filter does not provide the optimal noise for the EWMA/integral controller. This is of small concern, however, since the MMSE characteristic is only available at a single point and we have not found instances where controllers have been tailored for specific noises.

Box and Luceno, [16], present a method for determining the amount of noise amplification at the output given an EWMA controller smoothing constant θ:

$$\theta = 1 - K$$

Equation 2-36

where K is the loop gain ($=k_p k_c$). Under the assumption that the disturbance can be represented by an integrated moving average time series model with a smoothing constant θ_o (i.e. a non-stationary parameter $\lambda_o = 1 - \theta_o$, [16]) the result is:

$$\frac{\sigma^2_{output}}{\sigma^2_a} = 1 + \frac{(K - \lambda_o)^2}{K(2 - K)}$$

Equation 2-37

This is shown in Figure 2-7. Note that the output variance is related to the pre-model NIDI noise a, rather than the process noise d.

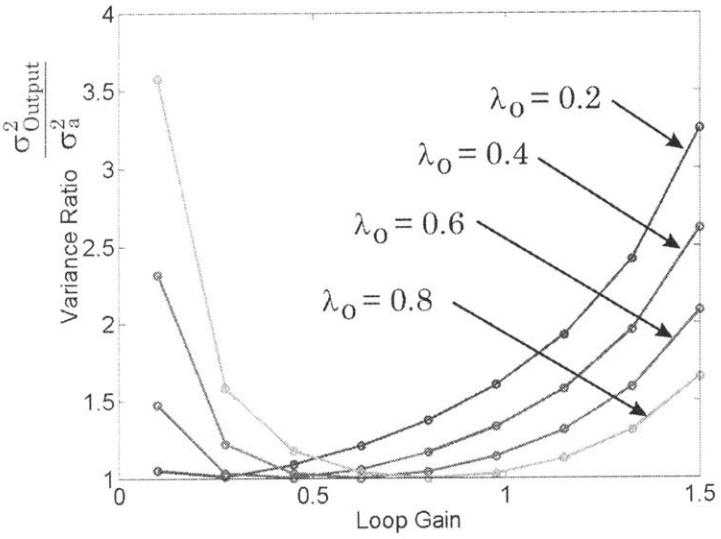


Figure 2-7 Ratio of the variance of the process output and NIDI shocks a, when the additive noise model is described by the IMA(1,1) model.

Siu [17] and Hardt and Siu [9] have shown that the amount of noise amplification from CtC can be accurately predicted for single input-single output processes subject to NIDI noise. This variance ratio for normal, identically distributed, independent (i.e. uncorrelated) noise is:

$$\frac{\sigma^2_{output}}{\sigma^2_a} = 1 + K\frac{1-(1-K)^{2(n-1)}}{(2-K)}$$

Equation 2-38

where n is the number of samples used to compute the variance ratio. Rzepniewski and Hardt, [18], have shown this equation to be equivalent to Equation 2-37 for the case of infinite samples and zero correlation. An analytical answer for the correlated-noise case has not been developed. However, the results of simulated trials are shown in Figure 2-8 and Figure 2-9. Each point on the plot is the result of 5,000 closed-loop runs. Figure 2-8 shows the variance ratio of the output and the NIDI noise shocks a for comparison with

Figure 2-7. Figure 2-9 shows the ratio of the output and the additive noise d to show process variance change. This is done since it is impossible to observe the theoretically underlying noise, a, in any physical process. The conversion between the two may be made through

$$\sigma_d^2 = (1-P)^2\left(\frac{1-P^{2n}}{1-P^2}\right)\sigma_a^2$$

Equation 2-39

where n is the number of points used to compute the ratio.

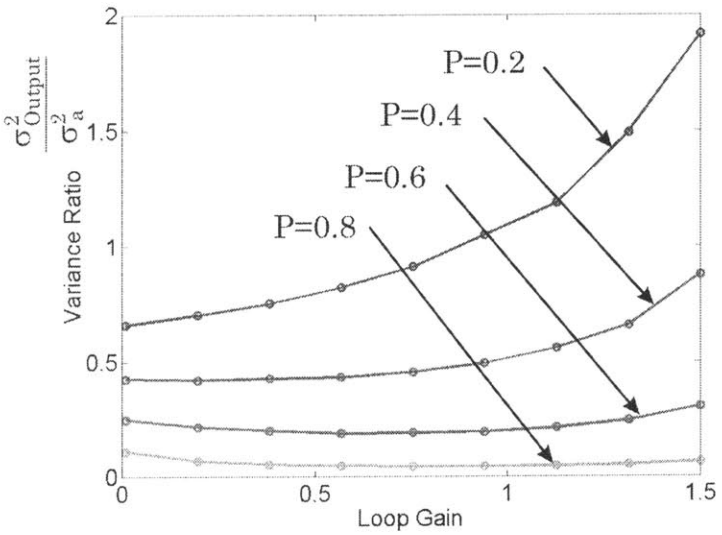


Figure 2-8 Ratio of the variance of the process output and NIDI shocks a, when the additive noise model is described by the correlated noise model, Equation 2-33.

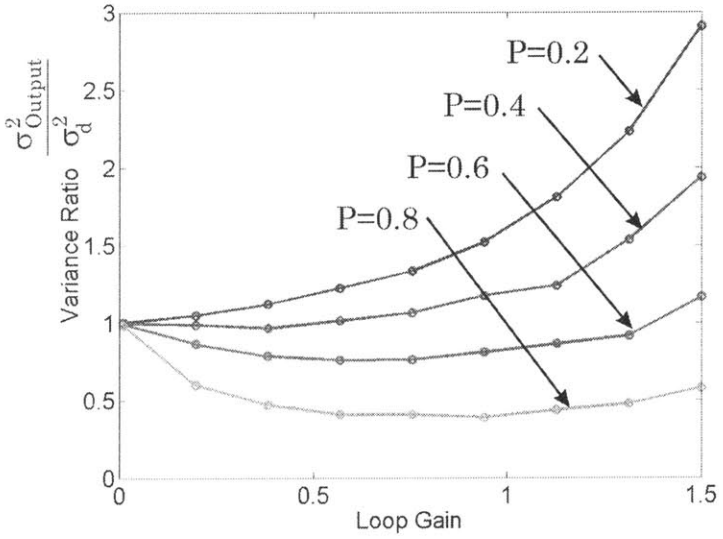


Figure 2-9 Ratio of the variance of the process output and the additive noise, *d*. Noise described according to Equation 2-33.

2.2 Double EWMA and Zero-Double Integral Control

The discussion thus far has centered on the idea of applying closed-loop control to reject step disturbances, or shocks. However, many real-world processes are subject to drift in addition to step disturbances. Examples of process drift are tool wear, thermal drift, and reagent depletion. As in the previous section, we will look at the work of other researchers and add to it by approaching the problem from a controls engineering perspective. In this case, especially, our perspective serves to simplify the problem of controller design.

2.2.1 Chen and Guo

Chen and Guo, [19], propose that processes that are subject to drift are described by the equation

$$y_t = \alpha_t + \delta_t + \beta_t u_t + \varepsilon_t \cong (\tilde{a}_{t-1} + p_{t-1}) + bu_t + \varepsilon_t$$

Equation 2-40

where δ represents the drift effect/disturbance. Two different double EWMA-based forms are available for updating the terms within the process equation. The first set is called the Predictor-Corrector Control (PCC) scheme, [19], and is governed by the following update equations.

$$\tilde{a}_i = \tilde{w}_1(y_i - \beta u_i) + (1 - \tilde{w}_1)\tilde{a}_{i-1}$$

Equation 2-41

$$p_i = \tilde{w}_2(y_i - \beta u_i - \tilde{a}_{i-1}) + (1 - \tilde{w}_2)p_{i-1}$$

Equation 2-42

where $\tilde{w}_1$ and $\tilde{w}_2$ are EWMA weights. Chen and Guo further rewrite the updating equation as

$$\tilde{a}_i + p_i = \tilde{w}_I \sum_{j=1}^{i} e_j + \tilde{w}_{II} \sum_{j=1}^{i} \sum_{k=1}^{j} e_k + \tilde{a}_0 + (\tilde{w}_1 i + 1)p_0$$

Equation 2-43

where

$$\tilde{w}_I = \tilde{w}_1 + \tilde{w}_2 - \tilde{w}_1 \tilde{w}_2$$

Equation 2-44

$$\tilde{w}_{II} = \tilde{w}_1 \tilde{w}_2$$

Equation 2-45

This is described as an integral-double integral controller. Box and Luceno, [16], have shown this to be the minimum mean squared error controller for processes subjected to IMA(2,2) noise. However, Chen and Guo show that the space of possible settings for the two constants $\tilde{w}_I$ and $\tilde{w}_{II}$ is limited. The range of possible combinations are shown in Figure 2-10 when the constants $\tilde{w}_1$ and $\tilde{w}_2$ are each limited to the standard weighting interval of (0,1).

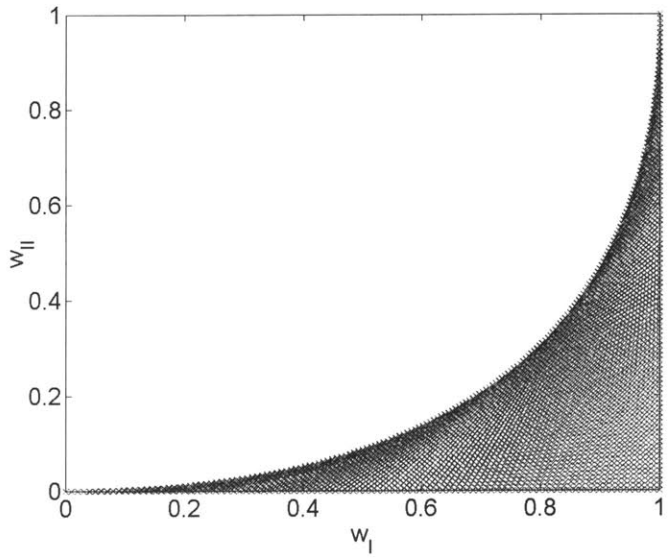


Figure 2-10 Control space of the predictor-corrector controller. Figure reproduced from work by Chen and Guo, [19].

To remedy the limited control space of the PCC formula, Chen and Guo propose a different double EWMA (d-EWMA) formula

$$\tilde{a}_i = \tilde{w}_1(y_i - \beta u_i) + (1 - \tilde{w}_1)(\tilde{a}_{i-1} + p_{i-1})$$

Equation 2-46

$$p_i = \tilde{w}_2(y_i - \beta u_i - \tilde{a}_{i-1}) + (1 - \tilde{w}_2)p_{i-1}$$

Equation 2-47

Combined, the two formulas yield

$$\tilde{a}_i + p_i = \tilde{w}_1 \sum_{j=1}^{i} e_j + \tilde{w}_2 \sum_{j=1}^{i} \sum_{k=1}^{j} e_k + \tilde{a}_0 + (i+1)p_0$$

Equation 2-48

As noted by the original authors, the only difference between the PCC and d-EWMA formulas is the inclusion of the drift term, p_{i-1}, in the estimation of $\tilde{a}_i$, Equation 2-41. The effect of this inclusion is the availability of the whole control space for each of the weights $\tilde{w}_I = \tilde{w}_1$, $\tilde{w}_{II} = \tilde{w}_2$.

2.2.2 Del Castillo

Del Castillo, [20], examines the performance of the double EWMA controller with PCC-form equations with a particular focus on transient performance. Five regions of process gain mismatch, β / b, giving different closed-loop performance are presented as a function of the EWMA weights. These regions are reproduced in Table 2-2. It should be noted that the lower boundary of region 5 is incorrect. A correct version of this is presented in Equation 2-54 and Equation 2-55.

Region #	Process Gain Mismatch Interval β / b	Expected Performance
1	$(-\infty, 0]$	Unstable
2	$\left(0, \frac{4\tilde{w}_1\tilde{w}_2}{(\tilde{w}_1+\tilde{w}_2)^2}\right)$	Stable, Oscillatory
3	$\left[\frac{4\tilde{w}_1\tilde{w}_2}{(\tilde{w}_1+\tilde{w}_2)^2}, \frac{1}{\tilde{w}_1+\tilde{w}_2+\tilde{w}_1\tilde{w}_2}\right]$	Stable, No oscillation
4	$\left(\frac{1}{\tilde{w}_1+\tilde{w}_2+\tilde{w}_1\tilde{w}_2}, \frac{2}{(\tilde{w}_1+\tilde{w}_2)}\right]$	Stable, Oscillatory
5	$\left(\frac{2}{(\tilde{w}_1+\tilde{w}_2)}, +\infty\right)$	Unstable

Table 2-2 Expected process performance as a function of process mismatch given EWMA weights. PCC updating equations used. Results reproduced from Del Castillo, [20]. The lower boundary of region 5 is incorrect; the corrected equation is presented in Equation 2-54 and Equation 2-55.

Confirming the advice of previous researchers, Del Castillo, [20], suggests the use of small EWMA weights to ensure long-run, asymptotic stability. This is emphasized particularly for the case when the model mismatch is large. The author notes that deliberately underestimating the process gain, $\beta / b > 1$, avoids sinusoidal oscillations if the EWMA weights are small. He also notes that the process remains stable for the case of $0 < \beta / b \leq 1$ for all combinations of $\tilde{w}_1$ and $\tilde{w}_2$ on the interval between zero and one. The case for small EWMA gains is tempered by the transient, short-term, response of the

system. Del Castillo shows that small EWMA weights result in long settling times, something which is highly undesirable for small-lot production.

Del Castillo derives the theoretical variance amplification for both the oscillatory and non-oscillatory cases. These equations are very complex and it is not clear what choices of EWMA weights minimize the variance, [20]. To choose a good combination of short-term performance and variance change, Del Castillo provides a program to perform a weight-based optimization.

2.2.3 Control Engineering Perspective

The extension of the integral controller to account for process drift is relatively straightforward. It is still assumed that a simple gain and delay model, Equation 2-18, is sufficient to describe the process. The drift term is then lumped into an additive disturbance at the plant output, as seen in Figure 2-1. As was done by Hardt and Siu, [9], the rest of the analysis is carried out in the z-domain.

The z-transform of a ramp disturbance of slope c is

$$c\frac{z}{(z-1)^2}$$

Equation 2-49

Its effect on the single integral controller from the previous section may be determined by using Equation 2-22 with the ramp function as its input. After simplification, the steady-state error (mean offset of the output) is

$$e_{ss} = \frac{c}{k_p k_c} = \frac{c}{K}$$

Equation 2-50

Thus, the single integral, EWMA controller may be sufficient for small ratios of drift and loop gain.

Application of Equation 2-22 reveals that a double integrator is necessary to completely eliminate steady-state error. Using the results from Chen and Guo, [19], the two double EWMA formulas may be rewritten in their z-transform equivalents. The PCC formula is

$$\frac{u}{e} = \frac{\frac{(\tilde{w}_I + \tilde{w}_{II})}{b} z\left(z - \frac{\tilde{w}_I}{\tilde{w}_I + \tilde{w}_{II}}\right)}{(z-1)(z-1)}$$
$$= \frac{\frac{(\tilde{w}_1 + \tilde{w}_2)}{b} z\left(z - \frac{\tilde{w}_1 + \tilde{w}_2 - \tilde{w}_1\tilde{w}_2}{\tilde{w}_1 + \tilde{w}_2}\right)}{(z-1)(z-1)}$$

Equation 2-51

and the d-EWMA formula is

$$\frac{u}{e} = \frac{\frac{(\tilde{w}_1 + \tilde{w}_2)}{b} z\left(z - \frac{\tilde{w}_1}{\tilde{w}_1 + \tilde{w}_2}\right)}{(z-1)(z-1)}$$

Equation 2-52

Both of these formulas have the required double integrator in the form of $(z\text{-}1)^2$ in the denominator. However, the two are not exactly the same. This may be seen by writing the formulas in a standard form

$$\frac{u}{e} = \frac{k_c z(z-\gamma)}{(z-1)(z-1)}$$

Equation 2-53

Because the operator is free to choose any value of b, the controller gain, k_c, is not limited. However, noting that $\tilde{w}_1$ and $\tilde{w}_2$ are limited to a range of (0,1), the available choices for the zero location, γ, depend on the form of the chosen EWMA equations. The PCC equation allows a range of $0.5<\gamma<1$. This is shown in Figure 2-11. This limited range might have been foreseen by the limited control space shown in Figure 2-10. The

d-EWMA control equations allow the full range of $0<\gamma<1$, as is shown in Figure 2-12. Note that, for the controller and plant pair under consideration, locating the controller zero outside of the interval (0,1) yields a system which is unstable at any gain.

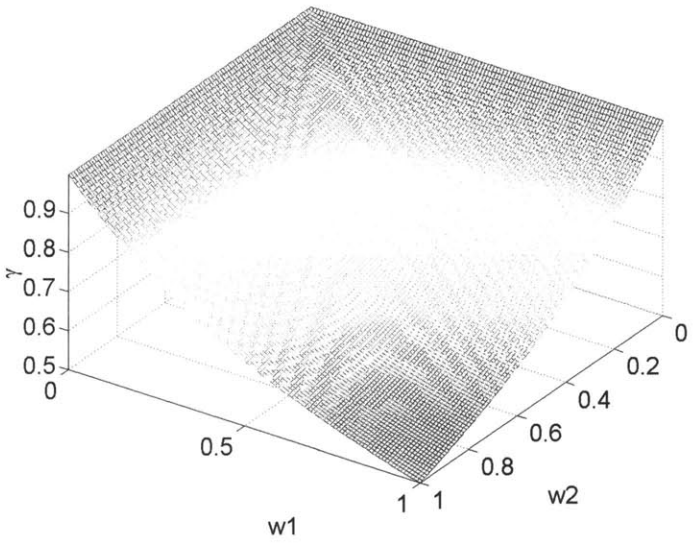


Figure 2-11 Controller zero location, γ, as a function of the EWMA weights for the predictor-corrector controller set of equations.

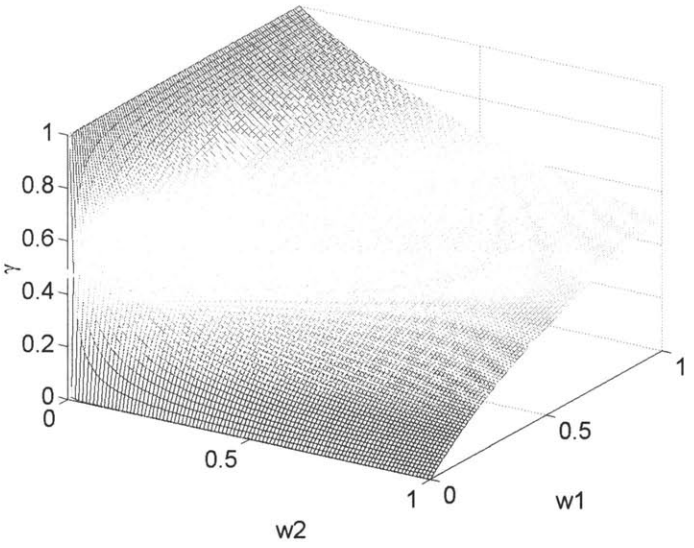


Figure 2-12 Controller zero location, γ, as a function of the EWMA weights for the d-EWMA set of equations.

The placement of the controller zero on the (0,1) interval boundaries yields two different controllers. Placing the zero at 1 leads to a pole-zero cancellation and simplifies Equation 2-53 down to the single-integral case, Equation 2-28. This removes the controller's ability to completely eliminate drift disturbances. On the other hand, placing the zero at 0 reduces the single integral-double integral combination of Equation 2-48 to a pure double integral. While this maintains the ability to completely eliminate drift, root-locus analysis reveals that the closed-loop process is at best marginally stable.

2.2.4 Transient Response

2.2.4.1 Root-Locus Perspective

The controller representation in Equation 2-53 allows further insight into the effects of choosing a combination of $\tilde{w}_1$, $\tilde{w}_2$, and b. Note that, in relation to Del Castillo's parameter β/b, β is a function of the physical plant and cannot be altered. Although it is not entirely clear from Equation 2-51 and Equation 2-52, Equation 2-53 reveals that the double EWMA controllers have only two design parameters: the controller gain and zero. Therefore, there must be coupling of effects from at least two of the parameters $\tilde{w}_1$, $\tilde{w}_2$, and b. A glance at either Equation 2-51 or Equation 2-52 shows that a change in either EWMA weight alters both the control gain and zero location.

A representative root-locus from the double EWMA controller is shown in Figure 2-13. The shape of the root-locus is characterized by two main features, a real-axis line and a circle segment. This representation may be adjusted for any choice of zero location, γ, by realizing that the line segment is always to the left of γ and the circle segment is of radius $(1-\gamma)$ with its center at γ. The motion of the closed-loop poles with increasing gain and zero location held constant is as follows. The poles originate at the +1 point on the

real axis and travel, symmetrically, around the circle until they meet again, break-in, on the real axis. Once there, one closed-loop pole travels towards the zero at γ and the other travels towards negative infinity.

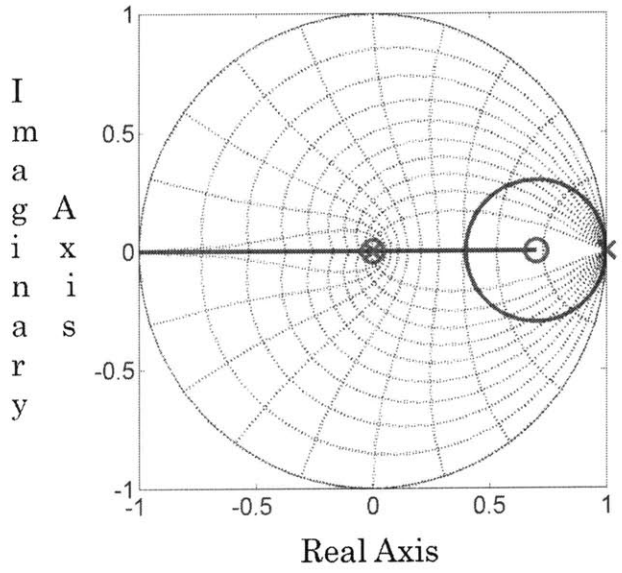


Figure 2-13 Double EWMA root locus.

Understanding the characteristics of the z-plane and how the roots move along the root-locus allows the confirmation of Del Castillo's results from Table 2-2, [20]. Here, it is assumed that the location of the controller zero remains fixed and only the loop gain is varied. Region 1 is defined by negative gains which always yield an unstable response. Region 2 describes the closed-loop poles' movement around the root-locus circle and results in stable, oscillatory response. Region 3 describes the stable, non-oscillatory response that occurs when the poles re-enter the real axis and before the left-bound pole travels past the z-plane origin. While the right-bound pole remains on the positive real axis, non-oscillatory, the left bound pole dictates a dominant, stable, oscillatory response as it travels between (0,-1] on the real axis; this is region 4. The unstable response of

region 5 takes place once the left-bound pole leaves the unit circle, as defined by Equation 2-54 rather than the equation in the table.

The location of the controller zero, γ, dictates the size, and sometimes the availability, of each region. Note that, as described previously, setting γ equal to either 0 or 1 changes the form of the controller. As γ moves from 1 to 0.5, the circle part of the root-locus grows. Since both the zero and the real-axis re-entry point move towards the origin and small settling times are achieved when the closed-loop poles are near the origin, Equation 2-21, this has the result of decreasing the fastest available settling time. This continues until, at γ=0.5, the closed-loop poles re-enter the real axis at the origin and the stable, non-oscillatory part of the root locus is eliminated. Continuing to decrease the value of γ has the effect of lengthening the minimum available settling time. From the transient response and settling time perspective, decreasing γ below 0.5 is undesirable.

Besides affecting the available responses and settling times, moving the controller zero influences the range of stable loop gains. The stable range is defined as (0,2) for γ approximately equal to 1. This becomes (0,4) for γ approximately 0. The upper stability boundary for the loop gain, K, may be derived by writing the closed-loop transfer function and solving for the gain that gives a closed-loop pole placed at -1 on the z-plane. This value is

$$K_{boundary} = \frac{4}{\gamma + 1}$$

Equation 2-54

This equation may be adapted to fit into Del Castillo's performance ranges presented in Table 2-2. Represented in the EWMA format, with PCC base equations, and solved for the ratio β/b, the stability limit becomes

$$\left.\frac{\beta}{b}\right|_{boundary} = \frac{4}{2(\tilde{w}_1 + \tilde{w}_2) - \tilde{w}_1\tilde{w}_2}$$

Equation 2-55

This is the lower boundary for region 5, and thus the correction required in Table 2-2. As Equation 2-54 reveals, γ<0.5 allows a larger range of uncertainty of β. It might therefore be beneficial to have a small γ if stability, rather than performance, is the key issue in a process under consideration.

The above procedure may be repeated to calculate a version of Table 2-2 which is based on the loop gain and zero position. The results of this are presented in Table 2-3. As in Table 2-2, region 3 does not exist for γ>0.5.

Region #	Process Loop Gain K	Expected Performance
1	$(-\infty, 0]$	Unstable
2	$(0, 4(1-\gamma))$	Stable, Oscillatory
3	$[4(1-\gamma), 1/\gamma]$	Stable, No oscillation
4	$(1/\gamma, 4/(1+\gamma)]$	Stable, Oscillatory
5	$(4/(1+\gamma), +\infty)$	Unstable

Table 2-3 Expected process performance as a function of loop gain, *K*, and controller zero position, γ

2.2.4.2 Determining the Controller Gains

The root-locus technique reveals the set of all possible transient response possibilities given a plant and a controller. This allows a designer to choose a desired K and γ without having to run countless experiments or simulations. Note that, although it is possible to set γ exactly, K is a product of user-defined constants and the inexactly-known true-process gain β or k_p.

$$K = k_c k_p = \frac{\tilde{w}_1 + \tilde{w}_2}{b}\beta$$

Equation 2-56

The result of this imperfect knowledge is a slight movement of the closed-loop poles and thus a slight alteration of closed-loop transient-performance.

The task of setting the EWMA weights and b is simple once a set of desired k_c and $\gamma$is chosen. Since $\tilde{w}_1$ and $\tilde{w}_2$ appear in both the controller gain and zero and since b only appears in the controller gain, one should set the EWMA weights first. For the PCC set of equations, the weights are determined according to

$$\tilde{w}_1 = \frac{\tilde{w}_2(1-\gamma)}{\tilde{w}_2 - (1-\gamma)}$$

Equation 2-57

Thus, $\tilde{w}_1$ is determined by a designer's choice of γ and $\tilde{w}_2$. Note that, although the EWMA weights correspond to different terms in the process equation, the PCC equations are symmetric and the places for $\tilde{w}_1$ and $\tilde{w}_2$ may be interchanged in Equation 2-57 without moving the controller zero. This symmetry does not exist for the d-EWMA equations, and the weights obey the relationship

$$\tilde{w}_1 = \frac{\gamma}{1-\gamma}\tilde{w}_2$$

Equation 2-58

The gain, b, for either set of equations may be calculated according to:

$$b = \frac{\tilde{w}_1 + \tilde{w}_2}{k_c}$$

Equation 2-59

Thus, the controller is fully determined.

2.2.4.3 Comparison with Other Researchers

The problem of setting the controller's weights for PCC equations has been previously addressed by both Garcia and Morari, [21], and Del Castillo, [20]. However, valuable insight can be gained from the perspective presented in this chapter. Garcia and Morari

suggest decreasing the EWMA weights if oscillations are present. Del Castillo, in addition to mathematically defining the regions of performance, comments that small EWMA weights may lead to very pronounced transient effects. In the subsequent discussion, only PCC equations are considered and it is assumed that the value of b, but not k_c, is held constant.

The effects of using small EWMA weights may be seen from the root-locus, the governing equations, and Figure 2-11. Small EWMA weights move the controller zero away from the origin, see Figure 2-11. This serves to decrease the circle part of the root-locus, thus changing the range of available transient responses. Increasing γ has a dual effect on the process-settling time. First, it limits the best settling time the closed-loop process could ever achieve. For γ on (0,0.5], the best setting time is achieved at the break-in point, where the circle and straight-line segments meet, located at $r=2\gamma-1$. Equation 2-21 may be applied to compute this time. Second, an increase in γ decreases the stable range of the loop gain. This effect is observed by computing K from Table 2-3. This has the effect of narrowing the stability bounds of the closed-loop process.

Equation 2-59 shows that, in fact, the loop gain, K, does not remain constant when $\tilde{w}_1$ or $\tilde{w}_2$ is changed. As before, b is assumed to stay constant. Root-locus analysis shows that the slowing effect of small EWMA weights may be exacerbated or ameliorated by the weights' coupling to the value of the loop gain. For discussion purposes, a stationary location of the controller zero is assumed. It is acknowledged that this assumption has been shown to be incorrect in the preceding paragraph. For poles located on the circle part of the root-locus, complex poles, decreasing the EWMA weights increases their distance from the origin and slows down the process. However, for real roots, decreasing

the EWMA weights actually moves the dominant pole towards the origin and speeds-up the system.

The total effect of changing the EWMA weights is a combination of the three factors. It changes the available set of transient responses, the distance the closed-loop poles travel for a certain loop gain, and, finally, it changes the loop gain itself, all at the same time. Thus, one should view the suggestion of Garcia and Morari, [21], with caution. Instead, it is suggested that the loop gain and the controller zero location be varied independently by varying $\tilde{w}_1$, $\tilde{w}_2$, and b in concert according to Equation 2-57, or Equation 2-58, and Equation 2-59. Such an approach leaves a minimum set of variables to manipulate (two), and helps predict closed-loop performance.

2.2.4.4 Overshoot

Thus far, the discussion of transient performance has centered on speed of response and oscillation. However, an important aspect of the transient response is overshoot of the target value. Although it is not predicted by the root-locus, it may be computed by simply writing out the closed-loop difference equations and computing the first few cycles. The closed-loop response of a double-EWMA controlled CtC process to a changing target input, T, is

$$y_i = (2-K)y_{i-1} - (1-\gamma K)y_{i-2} + KT_{i-1} - \gamma KT_{i-2}$$

Equation 2-60

The closed-loop response to the disturbance is

$$y_i = (2-K)y_{i-1} - (1-\gamma K)y_{i-2} + d_i - 2d_{i-1} + d_{i-2}$$

Equation 2-61

The importance of including this consideration is shown in Table 2-4. For a unit step jump in the target reference, the overshoot magnitude is (1-K). Given the potentially

desirable one time-step settling point (γ=0.5, K=2), the overshoot represents 100% of the jump in the reference. Similarly, the overshoot for the disturbance input is not negligible. Although it is asymptotically rejected, the disturbance causes a 100% overshoot and a 100% undershoot in the first two time-steps for γ=0.5, K=2. Clearly, the designer should be aware of this phenomenon when choosing the double EWMA controller.

	i-2	*i-1*	*i*	*i+1*	*i+2*
T	0	0	1	1	1
y	0	0	0	K	$(3-\gamma-K)K$
d	0	0	1	1	1
y	0	0	1	$1-K$	$K+K(\gamma-3)+1$

Table 2-4 Closed-loop process responses to a target reference input (top) and disturbance input (bottom). Each response is considered independently.

2.2.5 Process Noise

As with the single EWMA controller, the double-EWMA controller changes the way that noise is transmitted through the system. Box and Luceno, [16], have shown the double-EWMA controller to be the minimum mean squared error controller for processes subjected to IMA(2,2) noise. However, for this discussion, a slightly modified noise model from Equation 2-33 is used. The modification is the addition of a drift disturbance, such that the model is

$$D_t = d_t + 0.5i = Pd_{t-1} (1-P)a_{t-1} + 0.5i$$

Equation 2-62

where i is the run number.

Figure 2-14 and Figure 2-15 show the results of a simulation to obtain the variance ratio of the output and additive disturbance. Figure 2-14 contains the amplification for NIDI noise, P=0.00. Figure 2-15 shows the variance change for highly correlated noise, P=0.8. Both figures show ratios to within 10 percent of the maximum stable loop gain for

different values of γ. Although drift is included in the simulation, the figures show the ratio $\sigma^2_{output}/\sigma^2_d$ to exclude the drift term in the calculation of disturbance, or noise, variance. Also, to get rid of the possibility of including the process transient response in the calculation of variance, 11,000 runs are simulated and only the last 10,000 are used to compute each point on the plot.

Comparison of Figure 2-14 and Figure 2-15 with Figure 2-9 reveals that completely eliminating a drift disturbance results in higher variance amplification. Figure 2-15 shows that noise amplification is greater than one for all values of loop gain when γ=0.2 even for highly correlated noise. The designer should therefore make a conscious choice between obtaining parts that are off target with a narrower distribution and obtaining ones which are on target on average, but come from a wider distribution. As mentioned previously, Equation 2-50 could be used to gauge the magnitude of this offset.

Figure 2-14 and Figure 2-15 also show the importance of properly placing the controller zero. Placing γ next to the origin, in addition to providing undesirable oscillatory response at all gains, greatly amplifies the variance. Theoretically, γ=0.5, K=2.0 is the only choice that can give one time-step settling. However, this results in over $5x$ variance amplification. Continuing to move the zero away from the origin reduces the amount of noise amplification but it also slows down the system settling time. Clearly a trade-off analysis should be completed between speed of response and variance change. Del Castillo suggests a few methods for performing this, [20].

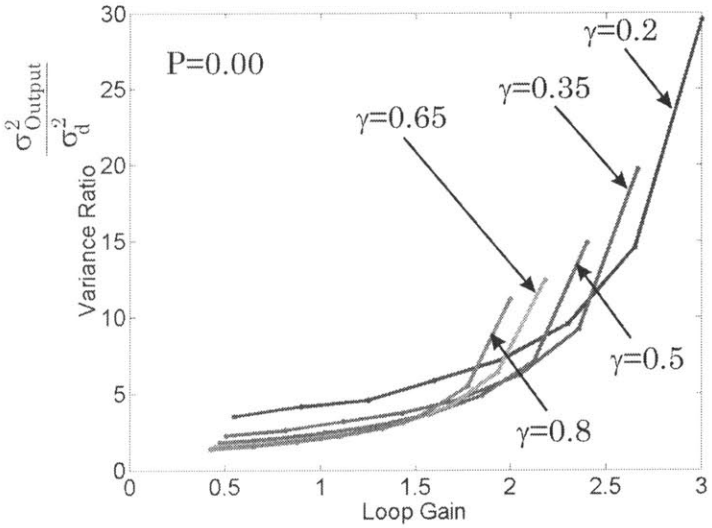


Figure 2-14 Ratio of the variance of the process output and the additive noise, *d*. Noise described according to Equation 2-33, P=0.00.

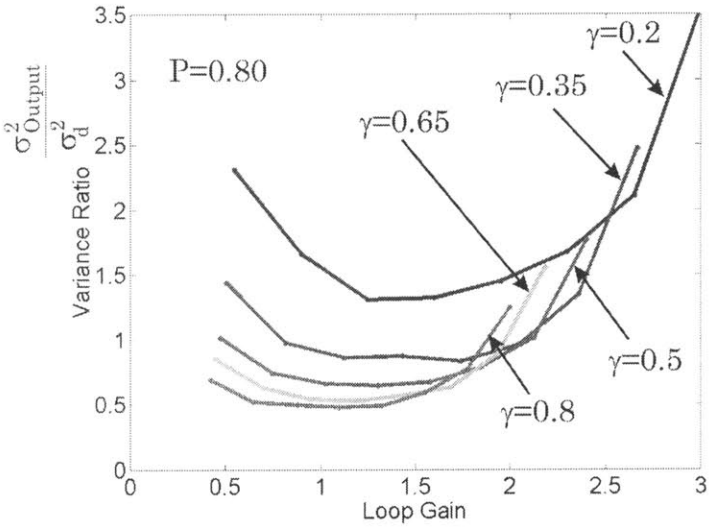


Figure 2-15 Ratio of the variance of the process output and the additive noise, *d*. Noise described according to Equation 2-33, P=0.80.

2.3 Single Input-Single Output Process Conclusions

In this chapter, we used a control engineering perspective to introduce a number of techniques that simplify the controller design process. First, we reduced the apparent design space of the controller parameters by using a mutually-orthogonal representation for the controller gain and location of the controller zero. Then, we used a transfer function representation and the root locus tool to help determine the values of those

parameters. To obtain an acceptable design, one must trade-off between the time response and variance change properties of the closed-loop process. Within time response, one must consider time to settle, oscillations or no-oscillations, and the amount of overshoot. Clearly, the controller parameters must be chosen with much forethought.

CHAPTER

3

IDEAL-CASE EXTENSION TO MULTIPLE INPUT-MULTIPLE OUTPUT PROCESSES

Our understanding of single input-single output processes motivates us to extend our outlook to multiple input-multiple output cycle-to-cycle processes. Even though a rich supply of literature exists for SISO processes, we have not found MIMO CtC processes to be adequately addressed in literature. As described in Chapter 1, we turn our attention to a particular class of these processes, ones that are regionally coupled. This chapter will extend SISO, single integral results to MIMO processes by looking at the transient response, noise change properties, and the stability region of the closed-loop process. A similar extension of SISO results to MIMO processes has been demonstrated before in Rzepniewski and Hardt, [18].

Throughout this chapter, and in subsequent ones, we will use a state-space representation of the process. This is done largely because a significant part of MIMO control theory is developed in this domain. A general state space control problem can be posed as:

$$z\underline{x} = A\underline{x} + B\underline{u}$$

Equation 3-1

$$\underline{y} = C\underline{x} + D\underline{u}$$

Equation 3-2

where $\underline{x}$ are the states, $\underline{u}$ are the inputs, and $\underline{y}$ are the outputs. Here, we have used the underline to signify that these are vector quantities. A, B, C, and D are matrices formed from the description of the process. More details on how to represent a process in state-space form may be found in Example 3-1 and in Friedland, [22].

Example 3-1 Using state-space representation

Obtain the state-space representation for the target input of the closed-loop CtC process shown in Figure 2-4 and reproduced below.

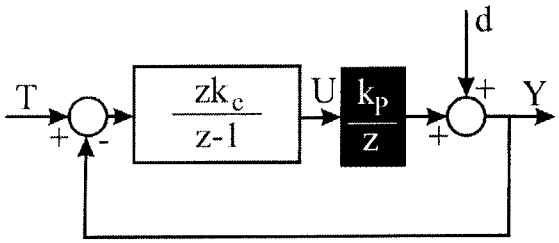


Figure 2-4 Integral-based control block diagram. Controller is shown in white, the plant in black.

First, we can write the closed, loop transfer function:

$$(z-1+k_p k_c)y = k_p k_c T$$

Now, we rearrange the terms

$$zy = (1-k_p k_c)y + k_p k_c T$$

It is easy to compare this equation to Equation 3-1 and Equation 3-2. The final solution is

$A=1-k_p k_c$

$B=k_p k_c$

$C=1$

$D=0$

The CtC plant is simply represented in state-space. Given the plant model

$$\underline{y} = \frac{1}{z} K_P \underline{u}$$

Equation 3-3

We can readily write the state space form as

$$z\underline{x} = [0]\underline{x} + K_p \underline{u}$$
$$\underline{y} = I\underline{x} + [0]\underline{u}$$

Equation 3-4

Additional concepts that are critical to the discussion of MIMO controllers are controllability and observability. Controllability may be defined as the ability to achieve a set of desired states in a finite amount of time. Mathematically, it is defined by the controllability grammian:

$$Q = \begin{bmatrix} B & AB & \cdots & A^{k-1}B \end{bmatrix}$$

Equation 3-5

where the superscript k is the order of the process. According to the algebraic controllability theorem, [22], the time-invariant process $\dot{\underline{x}} = A\underline{x} + B\underline{u}$ is controllable if and only if the rank of the controllability test matrix Q is equal to k.

Observability is defined as the ability to measure (observe) the output. Mathematically, it is defined by the observability grammian:

$$N = \begin{bmatrix} C' & A'C' & \cdots & (A')^{k-1}C' \end{bmatrix}$$

Equation 3-6

According to the algebraic observability theorem, [22]: The unforced time invariant process $\dot{\underline{x}} = A\underline{x}$ with the observation vector $\underline{y}=C\underline{x}$ is observable if and only if the rank of the observability test matrix N is equal to k, the order of the process. Although these concepts do not appear in SISO control, they are important in MIMO control where we may not have the ability to affect or observe all states.

3.1 MIMO Cycle-to-Cycle Controllers

A number of choices for single input-single output CtC control were studied by Siu, [17]. We will look again at the proportional, integral, and proportional-integral controllers with a mind to their applicability to multiple input-multiple output processes.

3.1.1 Proportional Controller

The proportional controller calculates the plant input by scaling the error signal:

$$\underline{u} = K_c \underline{e}$$

Equation 3-7

where K_c is now a gain *matrix*. Siu, [17], in looking at the SISO proportional controller, has shown it to be undesirable because this controller does not eliminate steady-state error. We can apply the proportional controller to a MIMO CtC process to make an additional observation. First, we write out the closed-loop characteristic equation for this controller-plant pair, [10],:

$$\left| zI + K_p K_c \right| = 0$$

Equation 3-8

By considering the smallest MIMO process, a 2-input, 2-output process, we can expand the characteristic equation to

$$z^2 + z\left(k_{p11}k_{c11} + k_{p12}k_{c21} + k_{p21}k_{c12} + k_{p22}k_{c22}\right) + \left(k_{p11}k_{p22} - k_{p12}k_{p21}\right)\left(k_{c11}k_{c22} - k_{c12}k_{c21}\right) = 0$$

Equation 3-9

where k_{pij} and k_{cij} are the i^{th}, j^{th} entry of K_p and K_c. Note, that we have assumed a controllable and observable process. First, we see that we cannot place both the roots of the characteristic equation at the origin of the z-plane. Second, this controller does not have the ability to completely eliminate steady-state error to a step disturbance, one which we are mostly concerned with in this thesis. The proportional controller, although simple to apply, does not give the desired closed-loop behavior.

3.1.2 Proportional-Integral Controller

We will skip the pure integral controller, which will be the subject of much of this thesis, to consider first the proportional-integral (PI) controller. A general PI controller block diagram is shown in Figure 3-1.

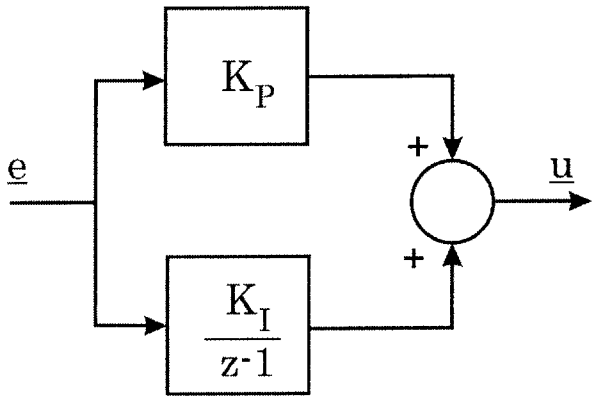


Figure 3-1 Proportional-Integral controller diagram.

We can follow each of the branches separately to assemble the signals the will make up the total controller output $\underline{u}$,

$$\underline{u}\big|_P = K_P \underline{e}$$

Equation 3-10

$$\underline{u}\big|_I = \frac{K_I}{z-1}\underline{e}$$

Equation 3-11

$$\underline{u}\big|_{TOTAL} = \underline{u}\big|_P + \underline{u}\big|_I = K_P \underline{e} + \frac{K_I}{z-1}\underline{e} = \frac{zK_P - K_P + K_I}{z-1}\underline{e}$$

Equation 3-12

This controller has a zero and a pure integrator pole, which will eliminate steady state error. Although it is possible to arbitrarily place the controller zero on the real axis, again assuming controllability and observability, locating it anywhere other than the origin leads to a slow or oscillatory transient response. To see this easily, we examine a SISO process. Figure 3-2 shows a SISO root locus and unit step response when the controller

zero is placed at z=0.5. It is easy to see that the step response contains a mix of both a first-order response shape, from the positive real-axis segment, and oscillation, from the negative real-axis segment. The effect of placing the controller zero on the negative real-axis is shown in Figure 3-3, with the zero at z=-0.5. Here, we see that, although non-oscillatory response is possible, we cannot achieve on time-step settling. Although we have used SISO processes in Figure 3-2 and in Figure 3-3, MIMO processes are expected to be limited to a similar set of possible time responses. This is especially true if we assume that each of the outputs is equally important and that each should have the same time response.

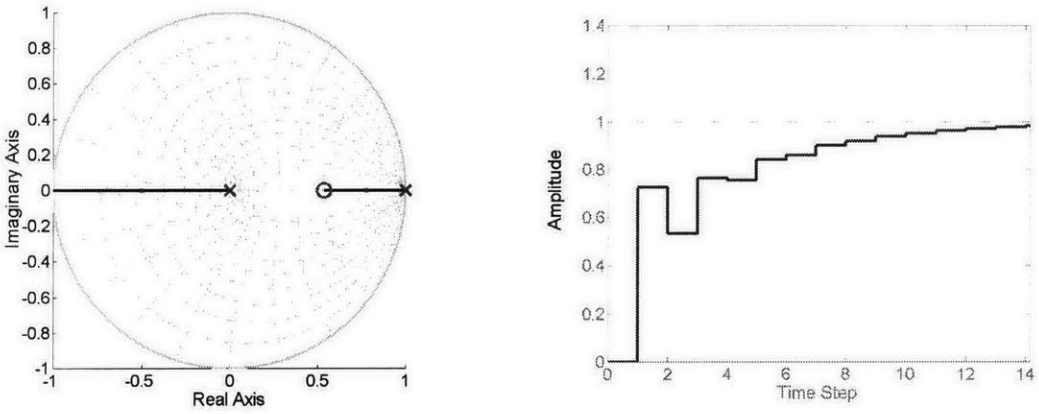


Figure 3-2 Typical SISO PI controller-CtC plant root locus (left) and corresponding unit step response (right) with the controller zero placed at z=0.5.

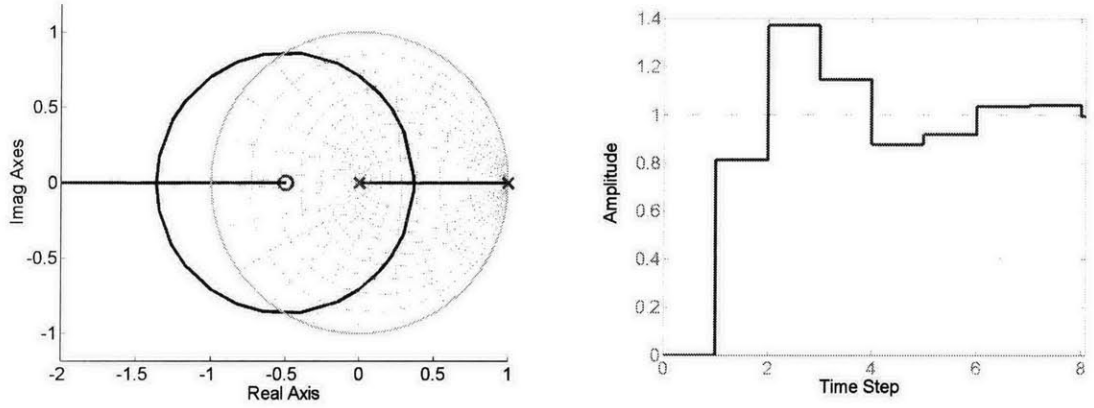


Figure 3-3 Typical SISO PI controller-CtC plant root locus (left) and corresponding unit step response (right) with the controller zero placed at z=-0.5.

When the controller zero is placed at the origin, it cancels the plant pole. This eliminates each of their effects. To place the zero of the controller at the origin, we have a condition, from Equation 3-12, that states

$$K_I = K_P$$

Equation 3-13

where the controller reduces to

$$\frac{z}{z-1} K_c$$

Equation 3-14

This is the MIMO form of the pure integral controller used by Hardt and Siu, [9]. Note that Figure 3-1 originally used a slightly different form of the integral controller. Had we started with Equation 3-14, we could not place the controller zero at the origin without eliminating the proportional portion of the proportional-integral controller.

3.1.3 Integral Controller

In Section 3.1.2 we saw that the PI controller reduces to an integral controller when we try to obtain desired closed-loop properties with our plant. Because of its desirable

noise transmission properties and both transient and steady-state response properties, we will concentrate on extending the integral controller to MIMO cycle-to-cycle processes.

First, we will separate the controller into two parts, the dynamics and the gain matrix:

$$G_c = \frac{z}{z-1} I * K_c$$

Equation 3-15

This operation is allowed because the delay elements, z, are just scalar multipliers of the gain matrix. With this in mind, we can transform the characteristic equation of the closed-loop process from

$$\left| I + \frac{1}{z} K_P \frac{z}{z-1} K_c \right|$$

Equation 3-16

to

$$\left| (z-1)I + K_P K_c \right| = 0$$

Equation 3-17

Assuming controllability, we can arbitrarily place the eigenvalues, poles, of the closed-loop process. This can be seen using a 2x2 MIMO process with

$$K_P = \begin{bmatrix} k_{P11} & k_{P12} \\ k_{P21} & k_{P22} \end{bmatrix}$$
$$K_c = \begin{bmatrix} k_{c11} & k_{c12} \\ k_{c21} & k_{c22} \end{bmatrix}$$

Equation 3-18

When these matrices are substituted into Equation 3-17 and the determinant is taken, we get

$$\begin{aligned} & z^2 \\ & + (k_{c11}k_{P11} + k_{c12}k_{P21} + k_{c21}k_{P12} + k_{c22}k_{P22} - 2)z \\ & + (k_{c11}k_{c22} - k_{c12}k_{c21})(k_{P11}k_{P22} - k_{P12}k_{P21}) - k_{c11}k_{P11} - k_{c12}k_{P21} - k_{c21}k_{P12} - k_{c22}k_{P22} + 1 \\ & = 0 \end{aligned}$$

Equation 3-19

Suppose that we wish to ignore the coupling and place both poles at the origin with a diagonal controller matrix, a case that was problematic when we wanted to shape the homogenous response with a proportional controller gain matrix. By looking at each of the z^n coefficient terms, we notice that it *is* possible to place the poles at the origin to obtain one time step settling and achieve our transient response goal.

3.2 Analysis of the MIMO Integral Controller

Equation 3-18 and Equation 3-19 show that for a 2x2 process there are four unknowns (the controller gains) and only two constraining equations (roots of z). In fact, it is easy to see that given a general n input, n output process, there are n^2 controller coefficients to determine but only n poles to place. The process is always under constrained through pole placement.

A MIMO integral controller is implemented in a *MATLAB* simulation to explore the behavior of an example process. A SISO process is used for comparison of results. The randomly chosen CtC plant is

$$G_P = \frac{1}{z}\begin{bmatrix} 2 & 3 \\ 5 & 7 \end{bmatrix}$$

Equation 3-20

In designing the controller, we have chosen to ignore the coupling effect of the off-diagonal terms and use a diagonal controller gain matrix. This reduces the problem to two unknowns and two constraining equations. Also, we have chosen to place both poles at the origin of the z-plane, theoretically giving us one time step settling. The results from a noise-free simulation are listed in Table 3-1 and selected results are shown in Figure 3-4. The SISO plant settles in one time step, as expected. However, the MIMO plant has a significant transient anomaly. In Figure 3-4 we see that a "transient" value is seen when

switching inputs. To understand its origin, we have to follow the signal flow through the closed-loop process. The process equation is written as:

$$I\underline{x} = \frac{z}{z-1}\frac{1}{z}K_P K_C \underline{e}$$

Equation 3-21

$$zI\underline{x} = I\underline{x} + K_P K_C(\underline{x}_R - \underline{x})$$

Equation 3-22

$$\begin{bmatrix} x_1(i+1) \\ x_2(i+1) \end{bmatrix} = \begin{bmatrix} (1-k_{c11}k_{p11}-k_{c21}k_{p12})x_1(i) - (k_{c12}k_{p11}+k_{c22}k_{p12})x_2(i) \\ -(k_{c11}k_{p21}+k_{c21}k_{p22})x_1(i) + (1-k_{c12}k_{p21}-k_{c22}k_{p22})x_2(i) \end{bmatrix} + \begin{bmatrix} (k_{c11}k_{p11}+k_{c21}k_{p12})x_{R1}(i) + (k_{c12}k_{p11}+k_{c22}k_{p12})x_{R2}(i) \\ (k_{c11}k_{p21}+k_{c21}k_{p22})x_{R1}(i) + (k_{c12}k_{p21}+k_{c22}k_{p22})x_{R2}(i) \end{bmatrix}$$

Equation 3-23

From above, we clearly see that coupling exists between both the outputs and the inputs; this coupling is the cause of the "unusual" behavior seen when using a diagonal controller gain matrix.

Data Type	Results	Controller Gain
Input	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	
SISO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}$	1 (Note that this is also the loop gain)
MIMO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -3.9615 & 1.9975 & 2.0031 & 2 \\ 2 & 2 & 2 & 2 & 2 & -10.0557 & 0.9940 & 1.0057 & 1 \end{bmatrix}$	$\begin{bmatrix} -1.4366 & 0 \\ 0 & 0.6961 \end{bmatrix}$
MIMO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 7.1044 & 1.9975 & 2.0021 & 2 \\ 2 & 2 & 2 & 2 & 2 & 17.0558 & 0.9940 & 1.0067 & 1 \end{bmatrix}$	$\begin{bmatrix} 2.4366 & 0 \\ 0 & -0.4104 \end{bmatrix}$

Table 3-1 Input-output data for a prime-number state matrix.

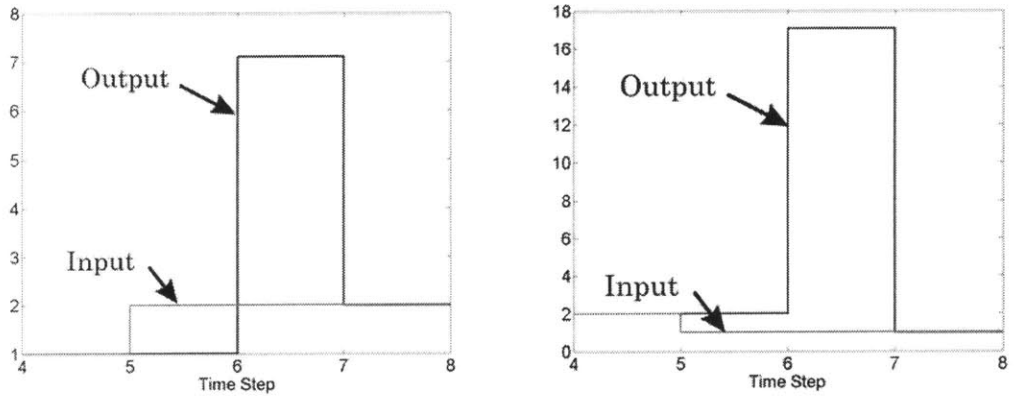


Figure 3-4 Selected input-output data for a plant matrix in Equation 3-20.

To see if this undesirable transient behavior could be a part of the processes that we are trying to address, a second, more realistic, MIMO process gain matrix is analyzed. The gain matrix has weak coupling between the inputs.

$$G_P = \frac{1}{z}\begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$$

Equation 3-24

The closed-loop poles of this process cannot be arbitrarily placed at the origin with a diagonal controller gain matrix without committing the designer to using gains that contain both real and imaginary parts. For this process, the two diagonal matrices are:

$$K_c = \begin{bmatrix} 1+0.1j & 0 \\ 0 & 1-0.1j \end{bmatrix} \qquad K_c = \begin{bmatrix} 1-0.1j & 0 \\ 0 & 1+0.1j \end{bmatrix}$$

Equation 3-25

Thus, a new approach is needed to decide on the controller gains with an under constrained equation process (we only have as many equations as the highest power of z in the closed-loop characteristic equation).

3.2.1 MIMO Gain Matrix: Deadbeat Analog Gain Selection

Our initial attempt at selecting the controller gains has resulted in non-satisfactory transient performance. As a second attempt, we look back to the SISO design guidelines

in Section 2.1.5.1. Since we have previously analyzed a SISO process and have identified a process with all its poles and zeros at the origin as having the best transient response, we would like to match the SISO process function:

$$x(i+1) = x_R(i)$$

Equation 3-26

with a MIMO function. Thus the closed-loop transfer function should take the form:

$$[(z-1)I + K_P K_C]^{-1} K_P K_C = \frac{1}{z} I$$

Equation 3-27

The above constraint on the closed-loop transfer function translates to the following condition:

$$K_P K_C = I \Rightarrow \begin{bmatrix} k_{p11} & k_{p12} \\ k_{p21} & k_{p22} \end{bmatrix} \begin{bmatrix} k_{c11} & k_{c12} \\ k_{c21} & k_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Equation 3-28

Thus, assuming that the plant matrix is invertible, we can obtain unique values for the controller gain matrix that will be free of the poor performance seen at input changes, and in MIMO match the performance of the SISO process. A quick check of the closed-loop characteristic equation reveals that the poles are, indeed, still at the origin. This conclusion for the controller gain matrix was obtained independently, but is the same as that reached by Kosut et al., [23].

Returning to the process composed of prime numbers, Equation 3-20, we can easily find the ideal gain matrix by inverting the process. Results from a *MATLAB* simulation are listed in Table 3-2; the unusual behavior associated with our previous choice of gain matrix is no longer seen.

Data Type	Results	Process Gain	Controller Gain
Input	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$		
SISO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}$	1	1
MIMO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 & 3 \\ 5 & 7 \end{bmatrix}$	$\begin{bmatrix} -7 & 3 \\ 5 & 2 \end{bmatrix}$

Table 3-2 Input-Output data for the process in Equation 3-20 with deadbeat analog gain selection.

Since a full gain matrix is used, we can also obtain useable gains for the second process, Equation 3-24, that was not feasible for implementation with a diagonal controller gain matrix. The results from a *MATLAB* simulation are shown in Table 3-3 and in Figure 3-5. We have thus achieved our desired transient response for MIMO processes.

Data Type	Results	Process Gain	Controller Gain
Input	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$		
SISO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}$	1	1
MIMO output	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.0101 & -0.1010 \\ -0.1010 & 1.0101 \end{bmatrix}$

Table 3-3 Input-output data for the process in Equation 3-24 with deadbeat analog gain selection.

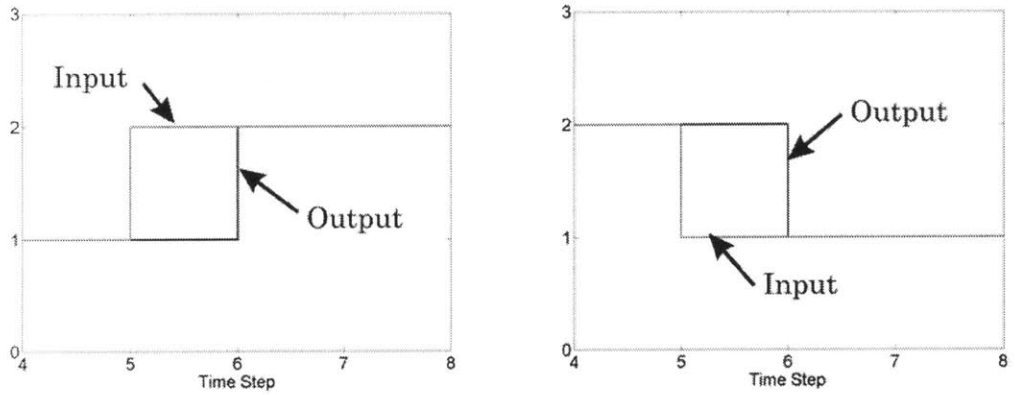


Figure 3-5 Input-output data for process in Equation 3-24 with deadbeat analog gain selection.

3.2.2 Steady State Error

To examine the steady state behavior of our process, we recall the general form of the final-value theorem, Equation 2-22, and apply it to our MIMO process

$$\underline{e}_{ss} = \lim_{z \to 1}(z-1)^2\left[(z-1)I + K_p K_C\right]^{-1}\underline{T}$$

Equation 3-29

where $\underline{T}$ is the target vector. From this form, we can see that we can have zero or finite steady-state errors for step and ramp inputs, respectively.

$$\frac{z}{z-1}C \text{ : step input with size matrix } C$$

$$\left(\frac{z}{z-1}\right)^2 M \text{ : ramp input with slope matrix } M$$

For a general disturbance signal, which may possess dynamics of its own, the steady state error is:

$$\underline{e}_{ss} = \lim_{z \to 1}(z-1)^2\left[(z-1)I + K_p K_C\right]^{-1}\underline{d}$$

Equation 3-30

As in the single input-single output case, a multiple input-multiple output cycle-to-cycle process will have zero steady state error to step inputs and a finite steady state error to ramp, or drift, inputs.

3.2.3 MIMO Noise Amplification

An analog to the noise amplification as a function of the loop gain is drawn in this section. First, we can consider the trivial case of having correctly identified the plant, but selecting the norm of the loop gain to be a factor α:

$$K_P K_C = \alpha I$$

Equation 3-31

The results for this case are just *n* SISO response plots, assuming an *nxn* process, due to the decoupling nature of the chosen controller gain matrix.

A more interesting case is one where the plant matrix is misidentified or not known exactly. This approach will leave the *real* process coupled even after we designed our controller to decouple our model. This will change the expected response making it different from *n* SISO responses. To achieve this, a nominal plant matrix is defined as:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$$

Equation 3-32

The controller matrix is designed according to the deadbeat analog method presented in Section 3.2.1. This yields a fully decoupled process if the nominal process matrix is used in the simulation. To display non-trivial, coupled, results, the plant matrix is perturbed, without readjusting the controller gain matrix, according to:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} + \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} = \begin{bmatrix} 1+\Delta_{11} & 0.1+\Delta_{12} \\ 0.1+\Delta_{21} & 1+\Delta_{22} \end{bmatrix}$$

Equation 3-33

with $\Delta_{11} \neq \Delta_{12} \neq \Delta_{21} \neq \Delta_{22}$. The perturbations are normally distributed with zero mean and a standard deviation of 0.1. The perturbed plant matrix is then used in a simulation of 5,000 cycles subject to additive NIDI, non-correlated, noise at the process output (see Figure 2-4). Figure 3-6 shows the results of 1,000 realizations of the perturbed MIMO plant matrix. One can see that the MIMO case appears bounded by the predicted SISO response. It is reasonable then, to use the SISO variance change plots, Figure 2-9, to predict the variance change of a MIMO process as a function of the norm of the loop gain.

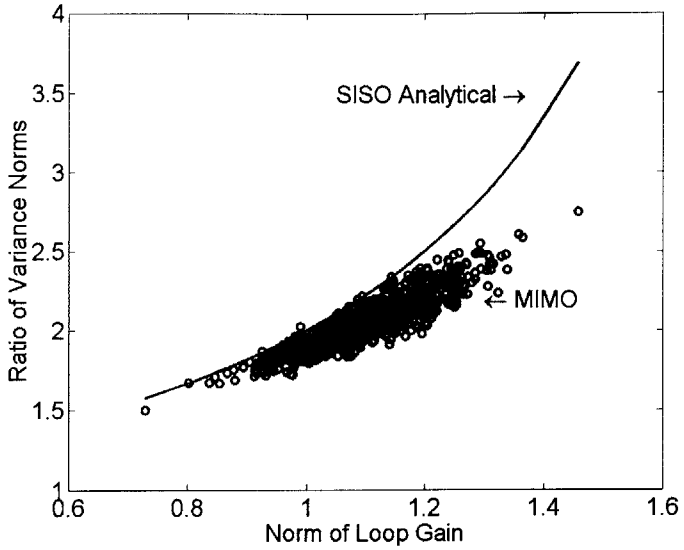


Figure 3-6 Process noise amplification for NIDI noise as a function of the norm of the loop gain.

Tests are also carried out to determine the noise-amplification effects of correlated noise. The in-process noise is passed through a MIMO correlating filter similar to that of Equation 2-33:

$$\underline{d} = \frac{1-P}{z-P}\underline{a}$$

Equation 3-34

Figure 3-7 shows simulation results from 1,000 realizations of the perturbed process matrix, each point is the result of 5,000 closed loop cycles. Again, the obtained MIMO variance change is similar to that obtained for SISO processes.

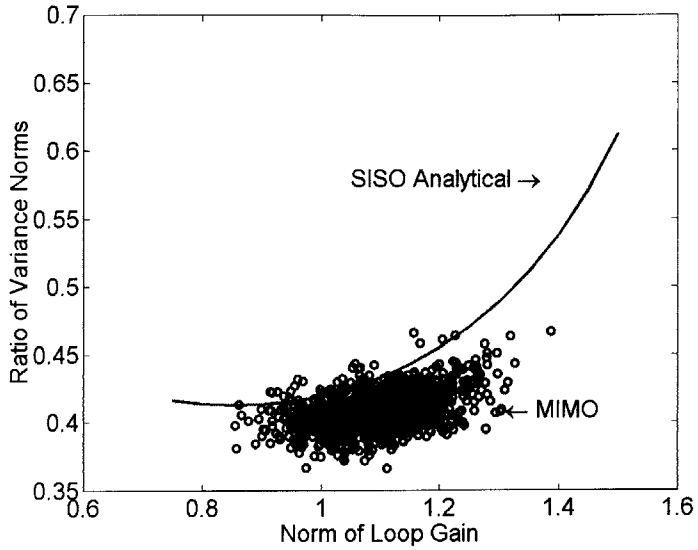


Figure 3-7 Process noise amplification for correlated noise as a function of the norm of the loop gain. MIMO correlation factor *P=0.8*.

3.2.4 Stability Region

As in the single input-single output case, we must consider the stable region of our MIMO process. By looking at the characteristic equation, we can see that it may be written in a familiar form

$$|sI - A_{cl}| = 0$$

Equation 3-35

where

$$A_{cl} \equiv I - K_P K_C$$

Equation 3-36

Since the eigenvalues of A_{cl} are also the poles of the closed loop process, we can express the stability criterion in another form:

$$|eig(A_{cl})| < 1$$

Equation 3-37

for a stable, finite settling time, process. This is exactly analogous to our SISO case where the closed-loop pole also had to remain within the region (-1,1) to be stable and have a finite settling time.

3.3 Multiple Input-Multiple Output Extension Conclusions

The rudimentary extension of cycle-to-cycle control to MIMO processes presented in this chapter appears to give us good results. As an initial attempt, we have overcome the obstacle of an under constrained problem by selecting an uncoupled (diagonal) controller. Unfortunately, this led to undesirable behavior when a change in target value was specified. From our SISO analysis, Equation 2-23 and Equation 2-24, we see that the target and disturbance transfer functions are only different by a sign. Therefore, we can expect this undesirable behavior to manifest itself as a response to process noise even when we hold the target input constant.

As a second attempt for specifying the controller gain matrix, we decided to mimic the ideal single input-single output transfer function. This has allowed us to design a full controller matrix simply by inverting the plant matrix. Analysis of steady state error, which is actually independent of the particular combination of controller and plant matrices, has shown that we maintain the ability to completely reject step disturbances. Next, a simulation-based experiment with a $2x2$ example matrix has shown that the variance change characteristics derived for a SISO process appear similar to those for a MIMO process. Finally, we have specified a stability criterion for the process based on keeping the closed-loop poles within the unit circle centered at the origin of the z-plane.

Many of the results of this chapter have been based on an assumed knowledge of the plant matrix. This knowledge, however, rarely exists *a priori* and is usually difficult to obtain. Attempts at obtaining the plant matrix are the subject of Chapter 4.

CHAPTER

4

PLANT MODEL IDENTIFICATION

In Chapter 3, we saw that the cycle-to-cycle plant model maintains the same form in MIMO as it does in SISO. However, the scalar SISO gain becomes a gain matrix. The fact that we must identify both the magnitude of influence as well as any cross-coupling between the inputs and outputs makes the problem of identifying this matrix much more difficult than identifying just a single gain.

A popular approach at identifying a plant is design of experiments (DOE), [1]. In this approach, a standard model is a full-factorial, it includes influence terms for all possible combinations of the inputs. A representative 2 input-1 output process would therefore be governed by the following full-factorial equation:

$$y_1 = \beta_0 + \beta_1 u_1 + \beta_2 u_2 + \beta_{12} u_1 u_2$$

Equation 4-1

where, y_i is the output, u_i is an input, and β_i is a static gain coefficient. Because there are four unknown coefficients, we must perform four experiments to fully determine Equation 4-1. As a rule, 2^n experiments are required to identify a full-factorial linear model with one output and n inputs.

In our class of process, we usually have n inputs and n outputs. Therefore, blind application of the full-factorial, linear model DOE procedure would require $n2^n$

experiments. However, we make an observation and a simplification to reduce the number of required experiments. First, since we assume a process with n outputs per part, we see that we obtain n pieces of data from each experiment. Second, we simplify our model to include only those terms which are linear in inputs, i.e. the term β_{12} in Equation 4-1 is excluded since it is quadratic in inputs. This is done to be compatible with linear control theory. Note that the term β_0 is also effectively eliminated by use of the operating points as shown in Equation 1-9. The number of required experiments is thus reduced to n. However, even with this simplification, the DOE approach to obtaining the plant matrix becomes grossly inefficient for large numbers of inputs. Remember from Section 1.5.1 that our lab-scale experimental platform has 110 inputs and 110 outputs. Thus a different approach must be taken.

4.1 In-Process Plant Identification

As a way of identifying the plant matrix, we have decided to test for the extent of coupling of outputs which may be present in our test process of sheet metal forming, Section 1.5.1. It is assumed that this will allow us to reduce the scope of our problem by allowing us to eliminate entries in the plant matrix when no coupling is present. Fewer entries to identify will mean fewer experiments. Simulations, discussed in Section 4.1.1 and in Section 4.1.2, are carried out in both *MATLAB* and *ABAQUS*. Simulation is preferred since it gives a noise free environment in which we can observe the transient response of the part points (outputs) directly above each pin (input). Both simulations assume quarter symmetry of the part, therefore only ¼ of the part is simulated. Figure 4-1 shows the simulated quarter symmetry part, a section of a cylinder; symmetry is along the X=1 and Y=1 axes.

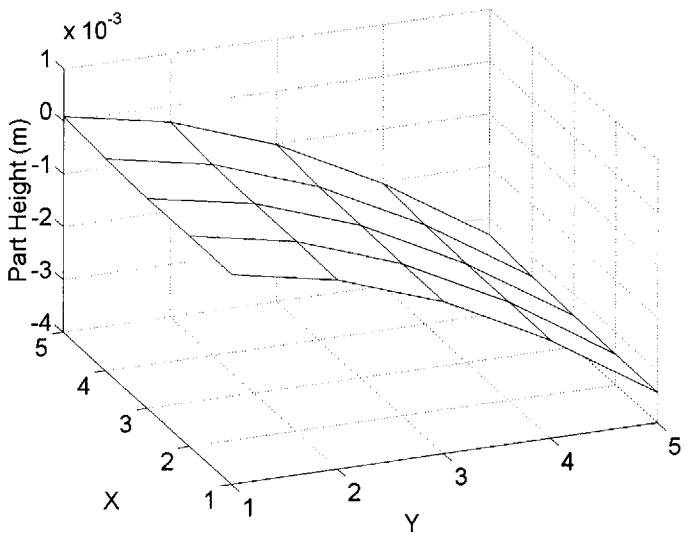


Figure 4-1 Quarter symmetry part used in *MATLAB* and *ABAQUS* simulations. Symmetry along the X=1 and Y=1 axes.

4.1.1 *MATLAB* tests

A *MATLAB* simulation was written that assembled a prospective plant matrix based on an assumed coupling between a single input and a region of outputs, see Figure 1-4. For our sheet metal forming process, this would be the coupling between the extension of a single input pin and the corresponding response from an area on the formed sheet metal shape. A Gaussian function was chosen as the candidate coupling function. The Gaussian distribution has been shown to provide a good approximation for distributed effects of sources. Dore and Gaussard, [24], and Hebert *et al.,* [25], use the Gaussian distribution shape to approximate the sensed effects of point sources in camera imaging. Wan *et al.,* [26], show that the thermal distributions in solids can be well approximated by the Gaussian shape. This approximation is used in favor of a more complex numerical solution, owing to its good performance and resulting quick computation time for the solution, [26].

As an initial set of tests, we perform a *MATLAB* simulation. In these tests, the input-output causal relation is defined by the Gaussian shape. We assume that the coupling spread, usually called the standard deviation, is half the spacing between pins and each distribution is centered on the respective pin. The code to assemble such a plant matrix is included in Appendix H. To test for coupling, we assume a diagonal controller with each non-zero entry equal to 0.9. If the plant were uncoupled, this controller gain would give us one time step settling. Figure 4-2 and Figure 4-3 show the response plots for X and Y points, respectively. We see that the closed-loop responses do not settle in one time step and that they show some oscillation. This oscillation effect appears to be similar to that seen in Figure 3-4 where we failed to fully decouple the process. These plots give us an idea of what to expect from the *ABAQUS* simulation, which is based on actual forming mechanics.

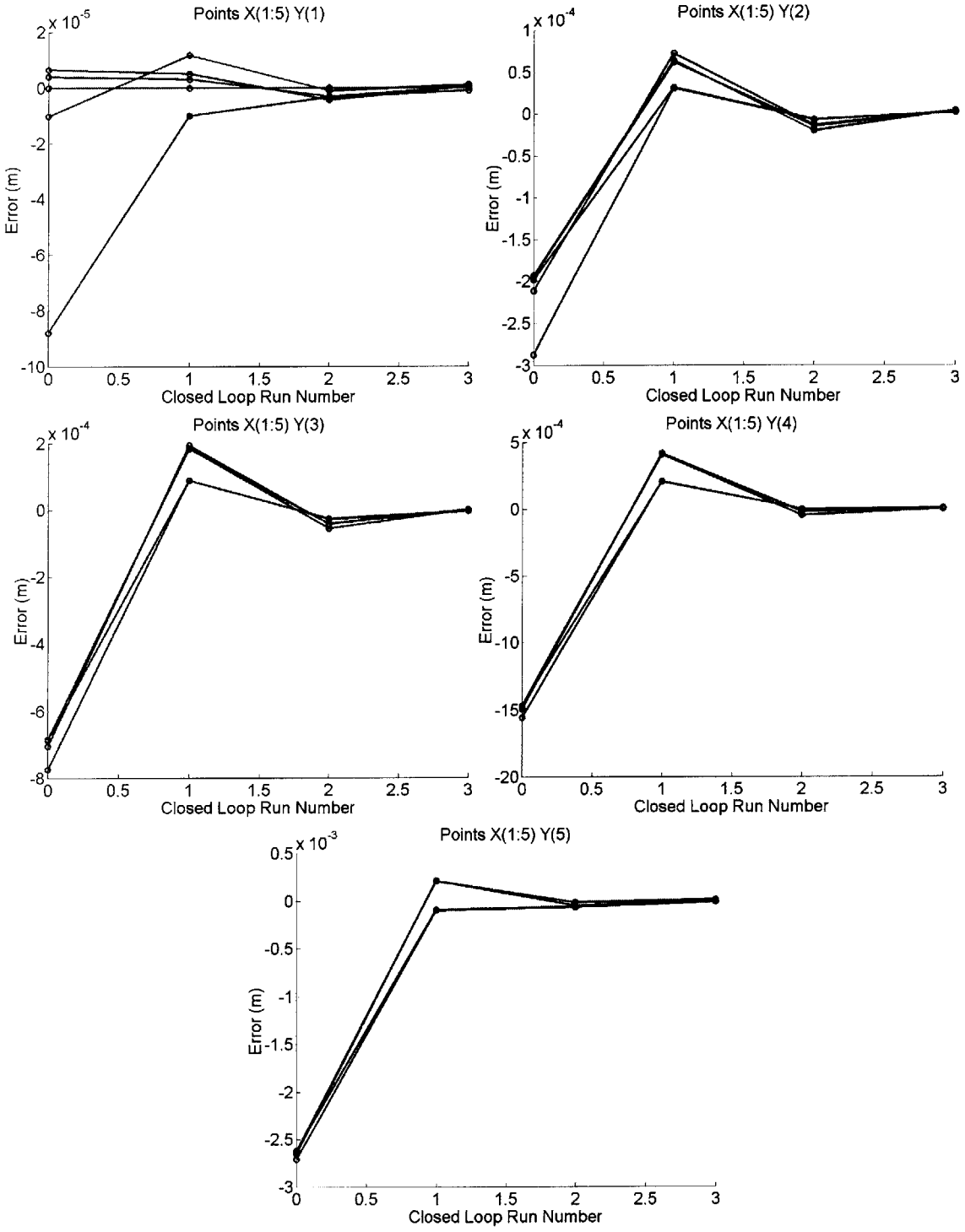


Figure 4-2 X coordinate error response as a function of closed loop run number. Gaussian function description of coupling, uncoupled controller. *MATLAB* simulation. Coordinates labeled as in Figure 4-1.

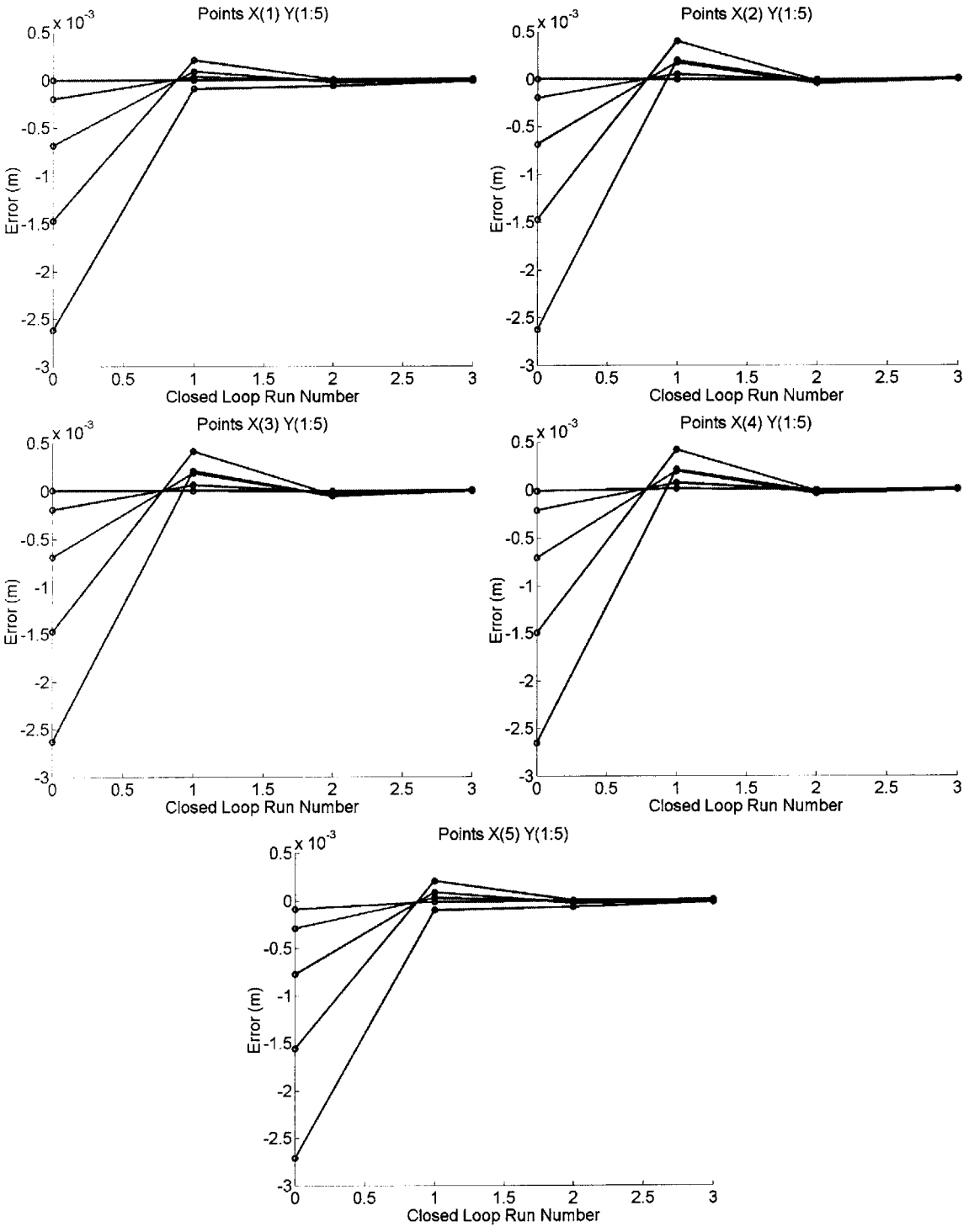


Figure 4-3 Y coordinate error response as a function of closed loop run number. Gaussian function description of plant coupling, uncoupled controller. *MATLAB* simulation. Coordinates labeled as in Figure 4-1.

4.1.2 *ABAQUS* Tests

For a second set of tests, we use an *ABAQUS* simulation of discrete-die sheet metal forming which was presented in Socrate and Boyce, [27]. This simulation was designed to be reflective of the actual sheet-metal forming process and thus includes such behavior as softening of the material with strain. As with the previous tests, we use a quarter symmetry part, as shown in Figure 4-1.

These numerical experiments are carried out following the same procedure that is currently in place on the physical process, [11, 28],:

1. the operator makes an initial guess based on the desired part shape and the part is formed.
2. the die is adjusted by the amount of the error, this generates the second open-loop, guess. The part is then formed.
3. the two previously formed parts are used to determine the plant model gains.
4. once the process gains are identified, the second part is thrown out and the process operates in a closed-loop fashion from the initial guess (run number 0 in Figure 4-4 and Figure 4-5).

The response plots, from a part formed according to the above procedure, are shown in Figure 4-4 and Figure 4-5. For most of the subplots, coupling effects are not seen, at least not like those observed in the *MATLAB* simulation. The responses appear like those of a first-order, uncoupled process. However, we must be cautious in making judgments as even a coupled system can appear first-order when the loop gain is underestimated. Our only hint of a coupled process appears in Figure 4-4 for the points labeled "X(1:5),Y(1)." This set of responses appears to exhibit the coupled response seen in the *MATLAB* tests. Overall, we cannot draw a conclusion as to the complete extent of coupling.

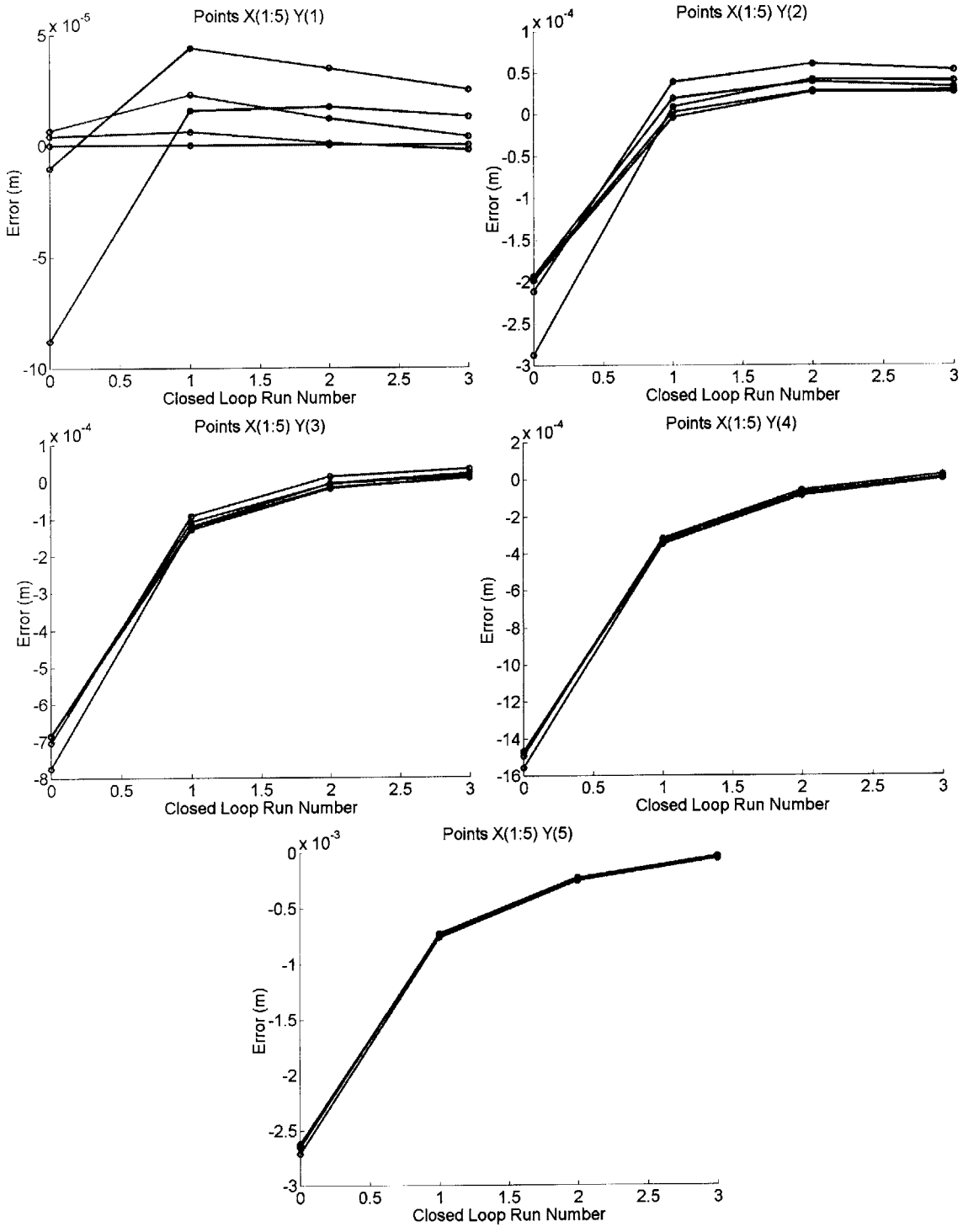


Figure 4-4 X coordinate error response as a function of closed loop run number. *ABAQUS*-based plant model, uncoupled controller. Coordinates labeled as in Figure 4-1.

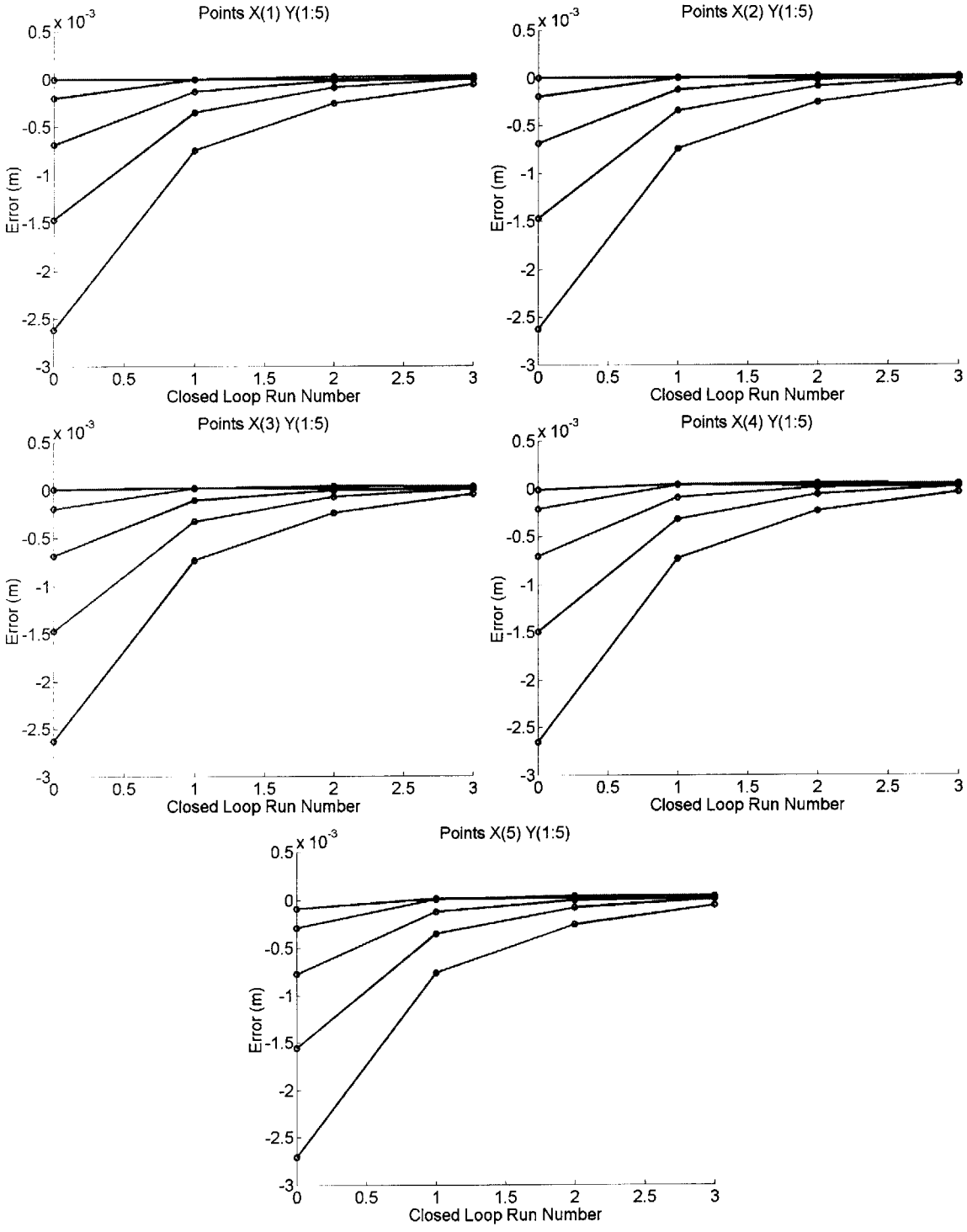


Figure 4-5 Y coordinate error response as a function of closed loop run number. *ABAQUS*-based plant model, uncoupled controller. Coordinates labeled as in Figure 4-1.

4.2 Plant Matrix Perturbation

The level of difficulty associated with obtaining a full-factorial model and the limited success of on-line identification of the region of coupling inherent in a specific process lead us to ask just how well do we have to know the process gain matrix. To answer this question, we set up a Monte Carlo simulation and try a number of different cases.

4.2.1 Pre-Process Plant Misidentification

First, our goal is to determine if the poles and zeros of the closed-loop process move significantly due to the non-exact knowledge of the plant matrix. A Monte Carlo simulation is implemented for the following plant gain matrix with weak coupling (small off-diagonal terms):

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$$

Equation 4-2

The controller gains are selected according to the deadbeat analog method, Section 3.2.1, assuming this nominal plant matrix. Following the calculation of the controller matrix, we perturb the plant matrix to simulate our imprecise knowledge. The plant matrix is perturbed according to:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} + \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} = \begin{bmatrix} 1+\Delta_{11} & 0.1+\Delta_{12} \\ 0.1+\Delta_{21} & 1+\Delta_{22} \end{bmatrix}$$

Equation 4-3

where $\Delta_{11} \neq \Delta_{12} \neq \Delta_{21} \neq \Delta_{22}$. The perturbations are zero mean with a standard deviation of 0.1. Note that one standard deviation represents from 10 to 100 percent of the K_{p11}, K_{p22} and K_{p12}, K_{p21} matrix entries. The process is simulated to run in a noise-free environment and 5,000 realizations of the plant matrix are used in this experiment.

Figure 4-6 shows location of the closed-loop poles and the accompanying zeros on the complex plane. Process zeros are shown according to their position in the process matrix. We can see that most of the poles remain on the real axis and that all poles are contained within a disk of radius approximately 0.4. Although this is different from the ideal behavior where all poles and zeros are located at the origin, this does show that the example process remains robustly stable against these, significant, perturbations. Figure 4-7 shows the settling times to within 2% and 0.2% of the target value as a function of the number of cycles. Again, this confirms that the MIMO process behaves well in the face of plant gain misidentification.

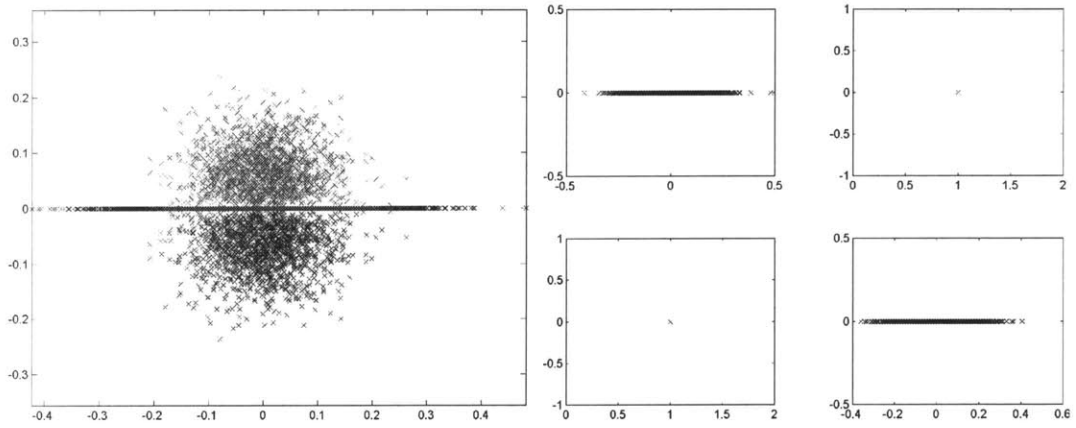


Figure 4-6 Closed-loop process poles (left) and zeros (four graphs on the right) on the complex plane. Zero results are displayed according to their position in the process matrix.

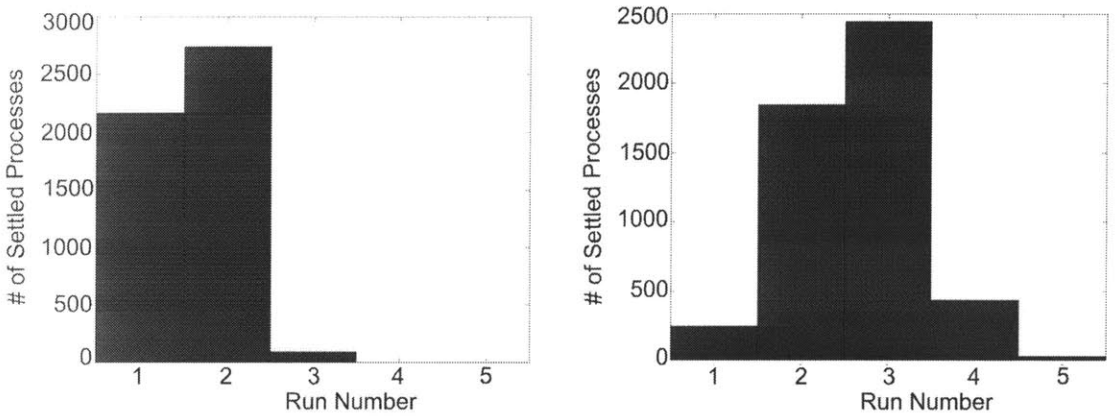


Figure 4-7 Process settling time based on 2% (left) and 0.2% (right) of target value. 5,000 total count.

We can run the same robustness study for the prime process, Equation 3-20. The results are shown in Figure 4-8. Note the change in range of the axes. This plant matrix appears to be much more sensitive to perturbation; added perturbations are still independent and N(0,0.01).

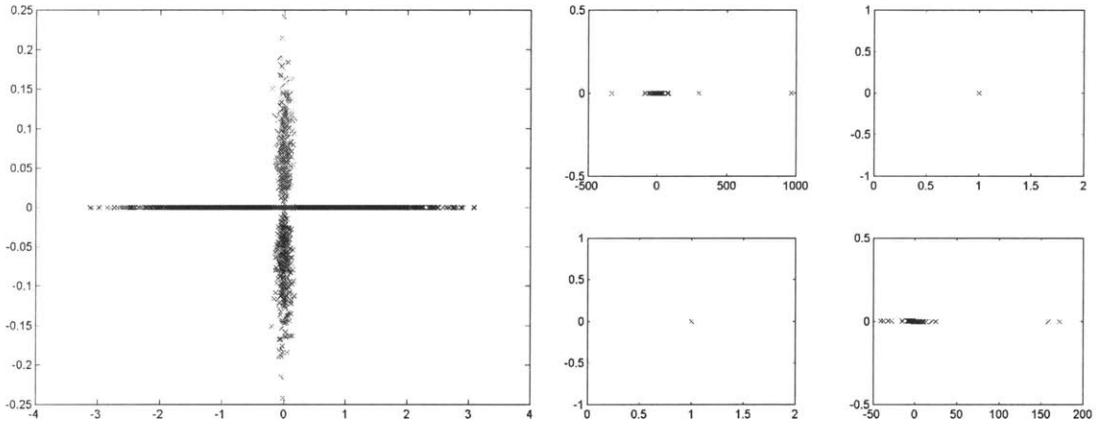


Figure 4-8 Closed-loop process poles (left) and zeros (right) on the complex plane. Zero results are displayed according to their position in the process matrix).

4.2.2 In-Process NIDI Plant Perturbations

A second test is carried out to determine the effects of misidentifying the plant before process runs as well as having slight variations in the plant between iterations. This form of perturbation is assumed as it more closely resembles the problem experienced in

processes where the workpiece changes between runs. The in-process misidentification takes the form of independent perturbation terms, N(0,0.01) which are added to the process matrix. Thus the gain matrix takes the form:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} + \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} + \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix}$$

Equation 4-4

where Δ_{ij} is defined as in Equation 4-3 and the δ matrix is changed between successive runs in the same process. Figure 4-9 shows the results of the Monte Carlo simulation. As expected, we see a small shift to higher settling times. However, no significant changes are seen in the settling time distribution. Again, this bodes well for the robustness of lightly coupled processes.

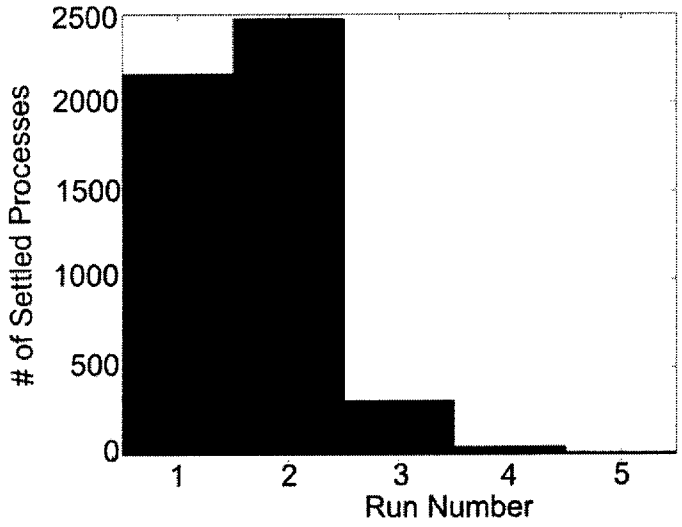


Figure 4-9 Closed-loop process settling time based on 2% of final value. 5,000 total count.

4.2.3 In-Process Coupled Perturbations

A third set of simulations was carried out to observe the effects of coupling of perturbation terms in addition to the previously mentioned NIDI phenomena. The plant matrix takes the form:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} + \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} + \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix} + \varepsilon \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Equation 4-5

where Δ_{ij} and δ_{ij} are defined as in Equation 4-4 and ε is N(0,0.01) and is changed between controller actions. The results of this study are shown in Figure 4-10. Once again, we see an expected trend to higher settling times.

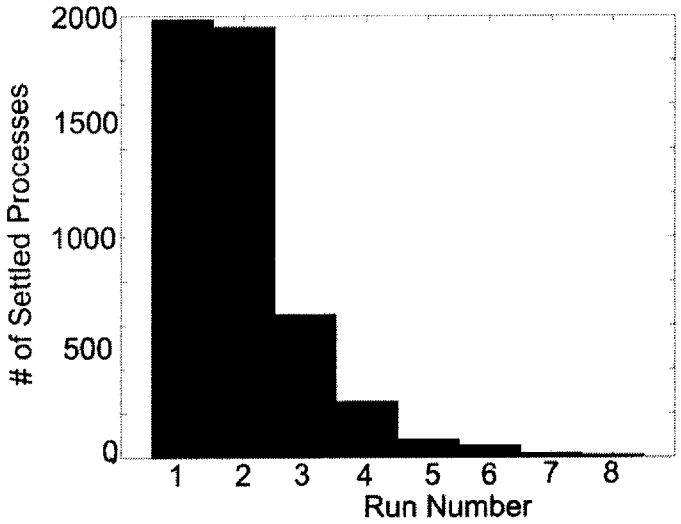


Figure 4-10 Process settling time based on 2% of final value. 5,000 total count.

4.2.4 Misidentification Penalties

Finally, we examine the trends resulting from the misidentification errors introduced above. Two principal cases are examined, plant perturbations before the process and plant perturbations during the process. Each case is examined independently of the other, unlike the "additive" analysis performed above.

First, the case of plant perturbations before the process is examined. This is the case in Equation 4-3. Figure 4-11 shows the percent of processes that settle to within 2% of the target in two time steps or fewer as a function of the standard deviation of the perturbations. Each data point on the formed line represents 5,000 independent trials. We

can see that up to about 0.09 standard deviation there is little penalty for plant misidentification.

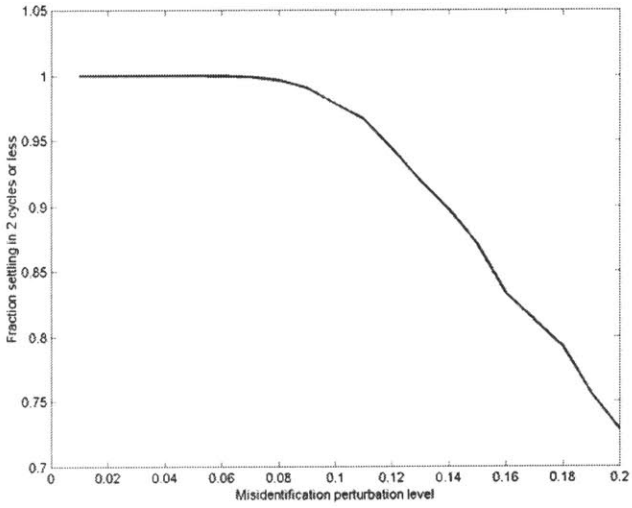


Figure 4-11 Percent of processes settling to within 2% of target in 2 time steps or fewer as a function of pre-process plant misidentification level. Level defined as the standard deviation of additive perturbation in Equation 4-3.

Now, the case of in-process plant misidentification is examined. This is a modification of the case presented in Equation 4-4. The plant matrix is:

$$K_P = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} + \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix}$$

Equation 4-6

where, as before, δ_{ij} are changed between cycles. Figure 4-12 shows the percent of processes that settled to within 2% of the target in two time steps or fewer as a function of the standard deviation of the perturbations. Once again, each data point on the formed line represents 5,000 independent trials. We see that there are more severe penalties for misidentifying the gain matrix during the process. The curve has a downward slope from the very lowest level of noise and it is higher than the slope observed in Figure 4-11. We

see that a standard deviation of 0.08 in plant misidentification now causes only 43% of the processes to settle within two time steps.

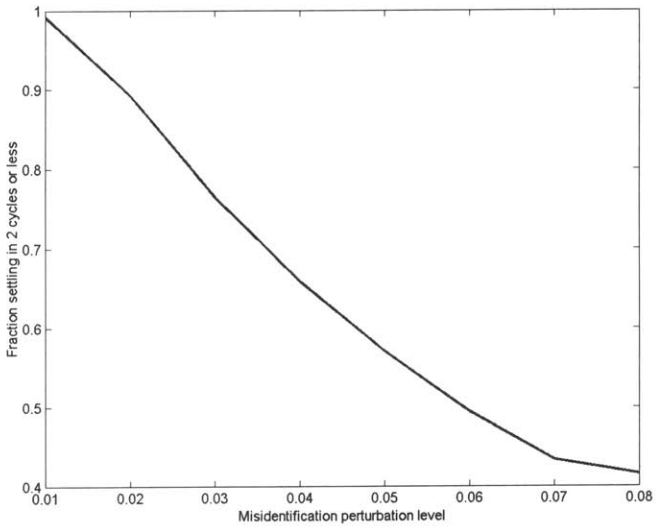


Figure 4-12 Percent of processes settling to within 2% of target in two time steps or fewer as a function of during-process plant misidentification level. Level defined as the standard deviation of additive perturbation in Equation 4-6.

4.3 Conclusions

In this chapter, we attempted to obtain a process model specific to our test process, sheet metal forming. We looked at the commonly used method of design of experiments. However, we quickly saw that this method is not practical for the size of multivariable process that we are trying to address. We also attempted to know the plant gain through in-process identification. This, unfortunately, led to inconclusive results. The difficulty associated with obtaining a particular plant model had prompted us to ask about just how precise we have to be in identifying the plant matrix. As a first attempt to answer this, we examined, through simulation, the stability and performance of a weakly coupled, $2x2$ process. These results indicated that the test plant can tolerate a large amount of

uncertainty while maintaining good performance. The implications of this realization are the subject of Chapter 5.

CHAPTER

5

A GENERAL MODEL FOR COUPLED MIMO PROCESSES – GAUSSIAN INFLUENCE COEFFICIENTS

In Chapter 4 we saw through numerical experimentation with a 2*x*2 weakly coupled process that we can tolerate a lot of uncertainty while remaining stable and providing good performance. While true robustness to uncertainty is the subject of Chapter 6, we will currently assume that the results we obtained from a 2*x*2 process can hold for larger, *nxn* processes. To this effect, we see that, rather than identifying a model form for each independent process under consideration, we can use a generic model which is easily calibrated. Easy calibration is essential, as each process output is a product which costs money. In this chapter, we develop just such a process model for regionally coupled MIMO CtC processes. This model was previously presented in Rzepniewski and Hardt, [29].

5.1 Generic Plant Model

Since the structure of the cycle-to-cycle model is so simple, gain and delay, we concentrate on the description of coupling among the outputs for the generic model. The problems we are attempting to address involve spatially distributed sources acting on a continuum; the influence of each individual input diffuses a limited amount beyond the

surface area that is directly acted on by an input. Thus a model that can efficiently capture this influence without having to populate the full process gain matrix would greatly reduce the number of required calibrating experiments. As presented in Rzepniewski and Hardt, [29], a static Gaussian distribution is proposed to describe the influence of each input in the MIMO cycle-to-cycle process output. As was stated in Section 4.1.1, the Gaussian distribution has been shown to provide a good approximation for distributed effects of sources, [24, 25, 26].

5.1.1 Gaussian Identification

A series of tests are carried out to determine the shape of the point response due to a pin input and to determine whether the Gaussian shape is a good approximation. Trial runs were carried out in a simulation written for *ABAQUS* by Socrate and Boyce, [27]. As in Section 4.1.2, a quarter symmetry part is assumed. The target shape is a 10 inch radius cylinder with the pins spaced at 0.5 inch intervals. Note that these tests, and the rest of the work, rely on the principle of superposition being applicable not only for the same pin from one time step to the next but also from one pin to its neighbors.

The shape of the pin response function is determined by taking the difference between a given die/part and the same die/part with one pin pushed outward by a set amount. Die displacements of 0.02" and 0.05" are used. Figure 5-1 shows the shape of the response for a quarter symmetry part with the (3,3) pin pushed up (left) and the (1,1) pin pushed up (right). First, we look at the fitment of the data points and the 1D Gaussian function. The 1D Gaussian is described as

$$K_P = s\left[\frac{1}{\sigma_X\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{Dist}{\sigma_X}\right)^2\right)\right]$$

Equation 5-1

where s is the scaling value, σ_X describes the coupling spread in the X direction, and *Dist* refers to the distance of the pin under consideration to the pushed-up pin. When this model is extended to MIMO processes, this model takes the form of a diagonal scaling matrix and the Gaussian function description. For example, a 5x5 plant gain matrix will be

$$K_P = \begin{bmatrix} 0.40 & 0.24 & 0.05 & 0 & 0 \\ 0.24 & 0.40 & 0.24 & 0.05 & 0 \\ 0.05 & 0.24 & 0.40 & 0.24 & 0.05 \\ 0 & 0.05 & 0.24 & 0.40 & 0.24 \\ 0 & 0 & 0.05 & 0.24 & 0.40 \end{bmatrix} \begin{bmatrix} s_1 & 0 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 & 0 \\ 0 & 0 & s_3 & 0 & 0 \\ 0 & 0 & 0 & s_4 & 0 \\ 0 & 0 & 0 & 0 & s_5 \end{bmatrix}$$

Equation 5-2

Data points along with minimum squared error fits, in both scale and coupling spread, are shown in Figure 5-2 and Figure 5-3. Figure 5-2 clearly shows the effects of using a quarter symmetry simulation. Note that the coupling spreads are different in each direction because a tensile force is applied solely in the Y direction. Also note that the magnitude of the gain (plotted on the z axis) changes along the part. This phenomenon might be caused by the difference in the direction of forming forces and inputs; pins in the (1,1) position are nearly perpendicular to the output surface while those further away tend to be at a smaller angle (reducing their effectiveness). Overall, the scaled Gaussian function does a good job of approximating the shape of coupling.

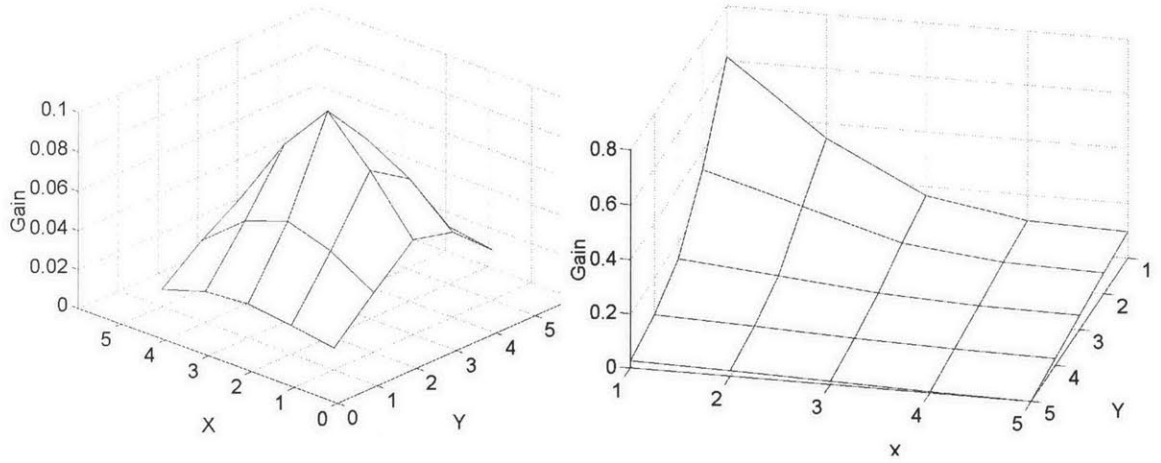


Figure 5-1 Gain magnitudes as a function of pin position for intermediate pin (3,3), (left), and center pin (1,1), (right).

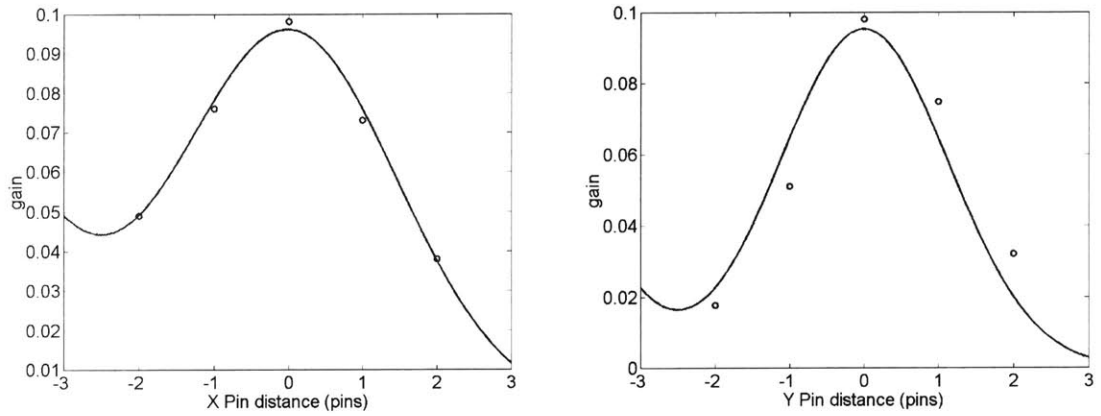


Figure 5-2 Gain spread as a function of pin position for (3,3) pin. X direction, σ_X = 1.5, scale = 0.4 (left), Y direction , σ_Y = 1.1, scale = 0.3 (right).

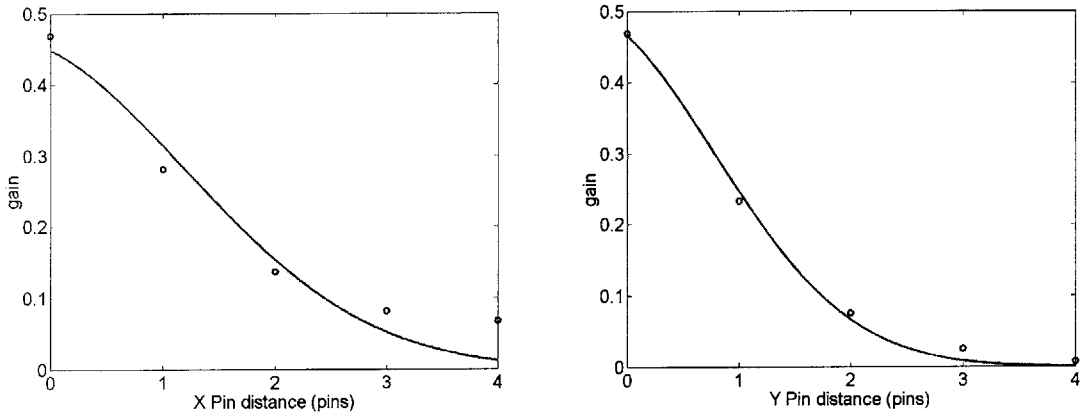


Figure 5-3 Gain spread as a function of pin position for (1,1) pin. X direction, σ_X = 1.6, scale = 0.5 (left), Y direction , σ_Y = 1.13, scale = 0.5 (right).

5.1.2 Scaling the Gaussian

In Section 5.1.1 we introduced a generic model for spatially coupled processes. This model comprises a scaling number and a Gaussian function description of the coupling. From this point forward, we will call this model the Gaussian Influence Coefficient (GIC) model. For a square, n output MIMO process, the GIC model has n+2 unknowns to identify: σ_x, σ_y, and n scaling values. We have used a specialized, "bump" test to determine the coupling spreads σ_x and σ_y. Now, under the assumption that each test gives n pieces of data, we need only a single perturbation, two tests, to determine the scaling for a MIMO process. Therefore, the whole model can be calibrated in just two perturbations, three tests. This fits within the "easily calibrated" requirement outlined in Section 1.3.

In examining the bump tests, we had chosen to uniformly scale the Gaussian spread function to obtain the shape of coupling. However, in a full MIMO process, we have two options for scaling the Gaussian without introducing any added complexity. The plant may be described as

$$\left[\bullet\right]_{output} = \frac{1}{z}\left[\begin{array}{c}Gaussian\\Spread\\Matrix\end{array}\right]\left[\begin{array}{c}Diagonal\\Scaling\\Matrix\end{array}\right]\left[\bullet\right]_{input}$$

Equation 5-3

or as

$$\left[\bullet\right]_{output} = \frac{1}{z}\left[\begin{array}{c}Diagonal\\Scaling\\Matrix\end{array}\right]\left[\begin{array}{c}Gaussian\\Spread\\Matrix\end{array}\right]\left[\bullet\right]_{input}$$

Equation 5-4

Recall that 1/z accounts for the process delay, Section 1.4.2.

The two forms lead to two different interpretations of how an input influences its neighboring outputs. Note that maintaining the proper direction, "input-to-output" is important in the proper interpretation of results; *columns* of the final, scaled spread matrix describe how an input affects the outputs. Equation 5-3 maintains the symmetry of the Gaussian function, merely stretching it up and down by the scaling factor. This can be seen since each scaling factor multiplies a single column of the Gaussian spread matrix. The result of using this formula is presented in Figure 5-4.

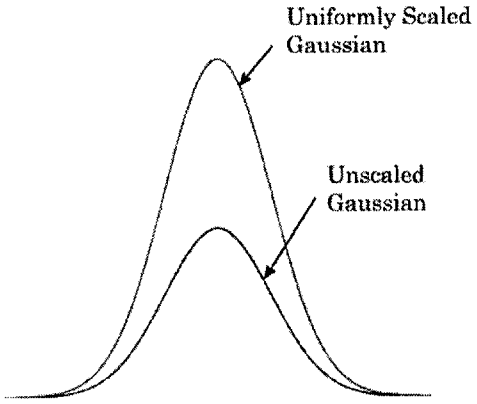


Figure 5-4 Uniformly scaled Gaussian spread function.

Equation 5-4 reverses the order of the Gaussian spread matrix and the diagonal scaling matrix. The net result is that a particular scaling factor will multiply a single row of the Gaussian spread matrix. Since, as mentioned previously, we concentrate on the "appearance" of the column, we can see that the result of this ordering is a non-uniform scaling of the Gaussian function; i.e. a stretching and distortion. This is presented in Figure 5-5 (a) where the scaled Gaussian is both stretched and distorted to the side. The scaling is shown in Figure 5-5 (b).

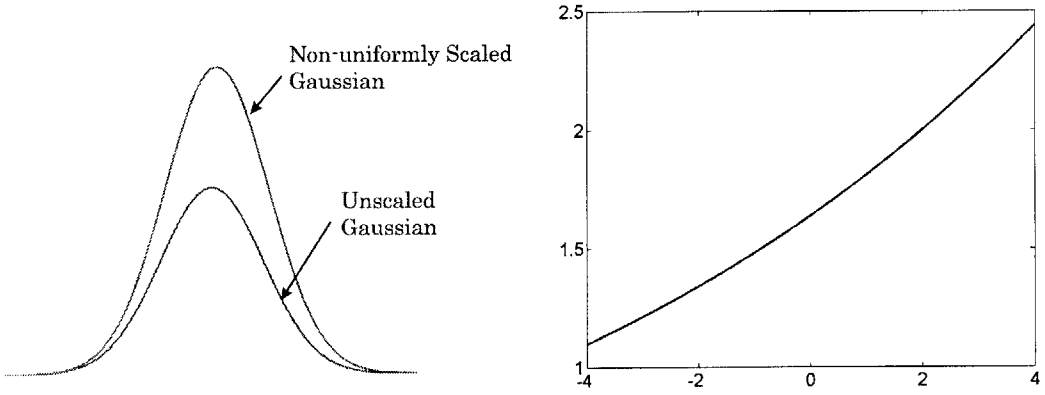


Figure 5-5 Non-uniformly scaled Gaussian spread function (a) and its scaling (b).

With two viable formulas to describe our plant, we need to choose the right one. To proceed, we will use two *ABAQUS* simulated parts and dies and compute the scaling factors. The parts are made on a 10x10 grid of inputs and outputs (thus 100 inputs, 100 outputs). The results of this computation are shown in Figure 5-6 (a) for Equation 5-3 (uniform column scaling) and (b) Equation 5-4 (non-uniform column scaling). The uniform scaling factors are large in magnitude and include negative values. An interpretation of this is that a positive input will actually result in a negative input, a characteristic not observed in any of the plants under consideration. Conversely, we see that non-uniform column scaling provides more intuitive scaling factor results; all are

positive, they are bounded by low magnitudes, and follow a pattern. A second argument for non-uniform column scaling may be obtained from a careful observation of Figure 5-2 and Figure 5-3. We can see that a non-uniform scaling, rather than the uniform one assumed in the figures, will do a better job of approximating the obtained results. These scaling tests reveal that Equation 5-4 is a better choice for the plant model.

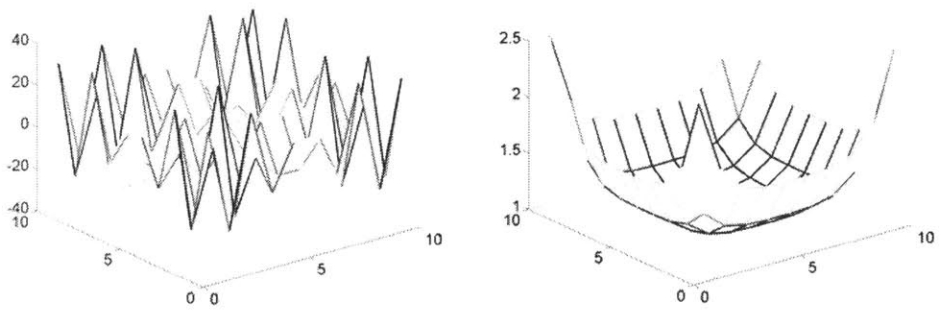


Figure 5-6 Scaling factors shown according to their position in a 10x10 grid of inputs. Uniform column scaling (a) Non-uniform column scaling (b).

Visual inspection of many Gaussian influence coefficient scaling matrices, obtained from actual experimental data, reveals that the scaling factors follow the second order shape shown in Figure 5-6, right. Following this lead, a basic, second order model fit is chosen to approximate the general model. The solution takes the form:

$$A\underline{\beta} = \underline{y}$$

Equation 5-5

where $\underline{y}$ is the diagonal of the scaling matrix rearranged as a vector, $\underline{\beta}$ is a coefficient vector to be determined, and

$$A = \begin{bmatrix} 1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_1^2x_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Equation 5-6

where x_1 and x_2 are the coordinates of the scaling factor y_i. Note that, since we have n^2 entries in the $\underline{y}$ vector and only nine are required to solve for the coefficient vector, β, we have a degree of redundancy. This redundancy should increase the robustness to noise.

5.2 Model Testing

A reasonable test of the goodness of a model is its ability to accurately predict parts given a die shape. With two experiments at different shapes, a single perturbation gain matrix can be calculated. This can then be used with a new input to predict a third shape. This predicted shape can then be compared to the actual shape to evaluate the model goodness. Two previously developed algorithms, the Spatial Coordinate Algorithm (SCA) and the Deformation Transfer Function (DTF), [11, 28], are compared to the Gaussian influence coefficient model. The SCA controller is a diagonal controller and, thus, assumes no coupling. The DTF controller is an attempt at introducing coupling through Fourier transformation. This, however, does not capture a physically intuitive plant model as it couples the whole plant model, [29, 30]. The interpretation of this is that each output will affect each input, regardless of plant size.

Five cylinders of different radii are used to test the three algorithms. The radii are: 6.65 in., 8.65 in., 10.65 in, 12.65 in, and 14.65 in. The different dies are shown in Figure 5-7. The baseline, or ideal, output is the output of the *ABAQUS* simulation of sheet metal forming described in [27]. Because of the large data set, there are 30 possible combinations of models and target parts, therefore a standard numbering convention is chosen. This convention is explained in Table 5-1.

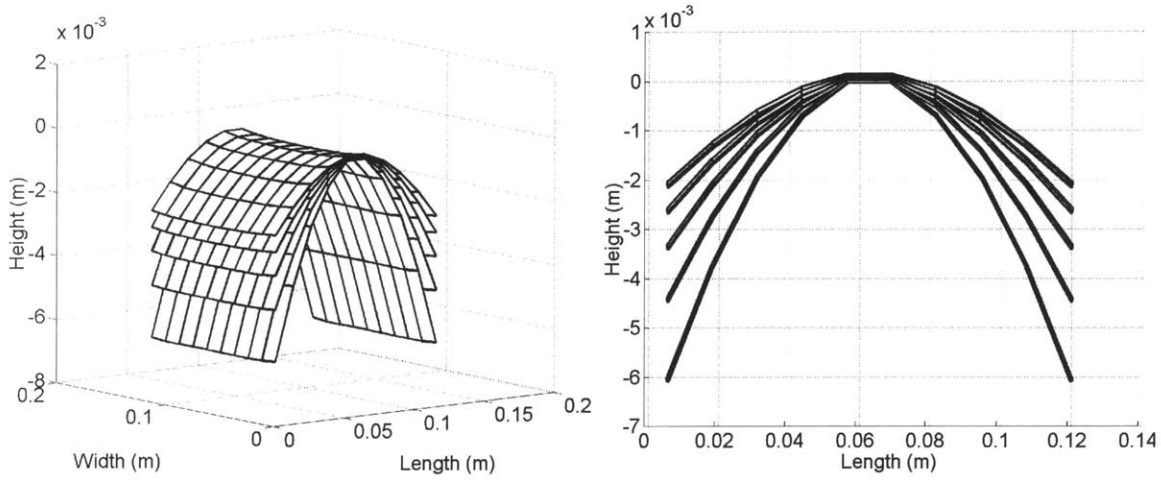


Figure 5-7 Five cylinder dies used for extended model testing.

Parts used for model	Target part	Test number
2 and 3	1	1
2 and 4	1	2
2 and 5	1	3
3 and 4	1	4
4 and 5	1	5
1 and 3	2	6
1 and 4	2	7
1 and 5	2	8
3 and 4	2	9
⋮	⋮	⋮

Table 5-1 Standard test numbering convention for extended model testing.

Figure 5-8, Figure 5-9, and Figure 5-10 show the error between a predicted part and one formed by the *ABAQUS* simulation. Figure 5-8 shows the results of noiseless simulation. We can see that the GIC model gives the best results in all cases. The results from a simulation with noise are shown in. In these results, we do not impose the quadratic scaling coefficient model described in Section 5.1.2. As we can see, neglecting to smooth the GIC scaling coefficients makes the process model susceptible to noise. Figure 5-10 shows the error between the formed and predicted part when the GIC scaling coefficients are smoothed, the quadratic model is used. Note that this reduces the error by nearly half. The GIC algorithm remains the best overall predictor of future parts. The

results of these tests are in line with the physical insights offered into each algorithm; sheet metal is neither fully decoupled (SCA) nor fully coupled (DTF), it is only locally coupled (GIC).

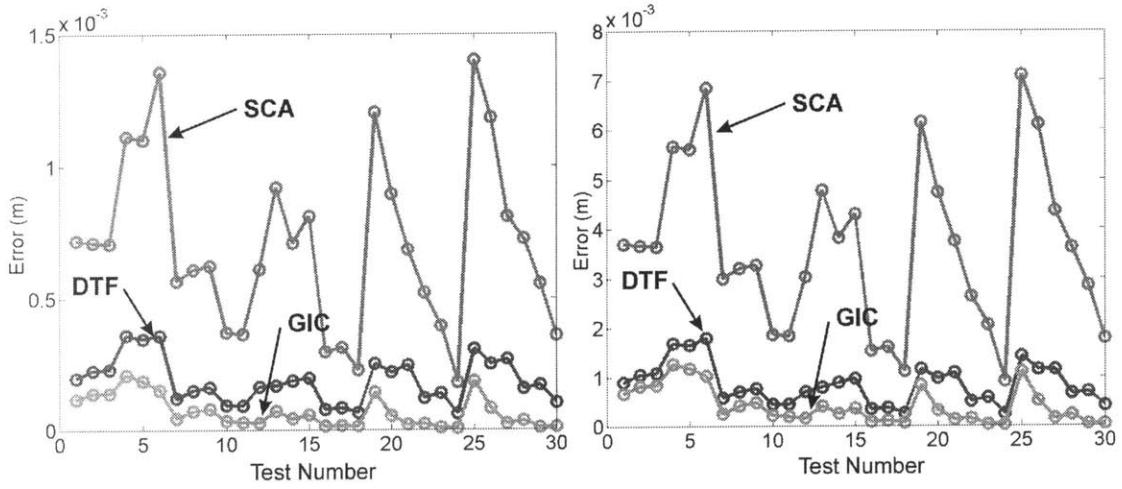


Figure 5-8 Error between predicted and formed part. (Left) Maximum absolute error (right) Norm of part error (*ABAQUS* simulated forming, Noiseless conditions).

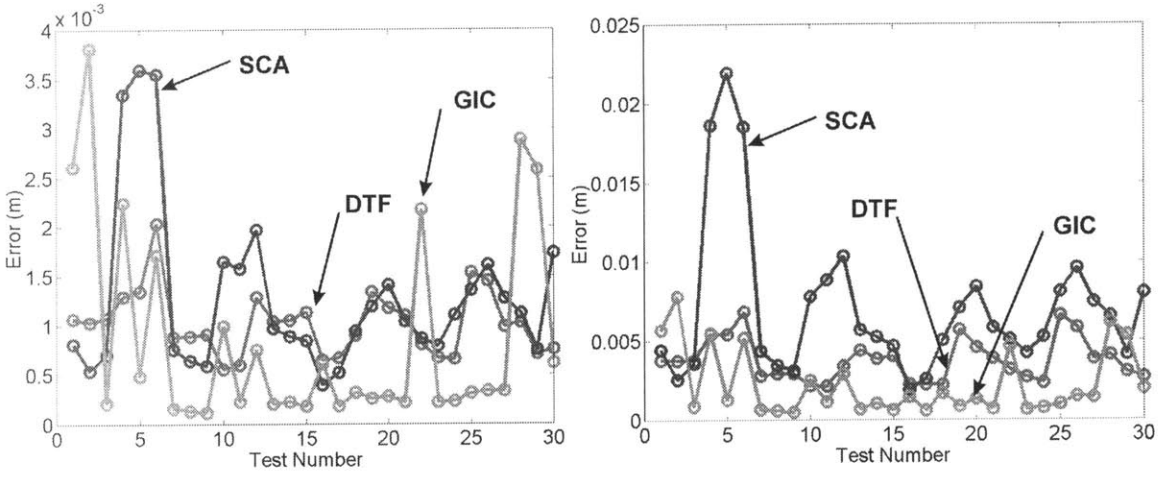


Figure 5-9 Error between predicted and formed part. (Left) Maximum absolute error (right) Norm of part error (*ABAQUS* simulated forming, Noisy conditions).

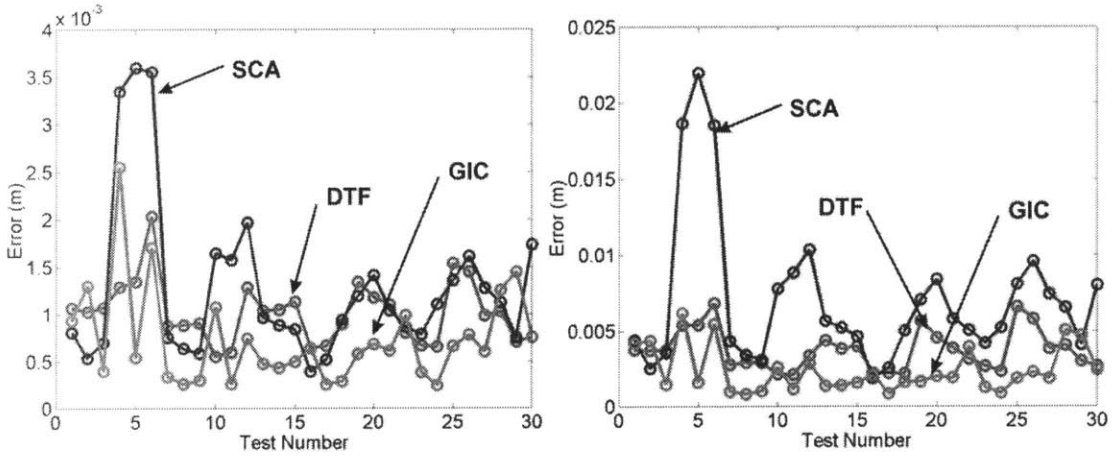


Figure 5-10 Error between predicted and formed part. (Left) Maximum absolute error (right) Norm of part error; smoothed plant matrix (*ABAQUS* simulated forming, Noisy conditions).

5.3 GIC Model Coefficients

In addition to being a good predictor, a good model remains constant for similar target parts or for the same target part between cycles. This consistency in the model is the key to predicting and choosing the desired performance of a controller; it allows us to perform a trade-off between speed of recovery from disturbances and noise amplification, as we had to do with SISO processes in Chapter 2.

In order to view the scaling coefficients more easily, the 10x10 scaling matrix is rearranged column-wise as a vector according to:

$$\begin{bmatrix} column1 & \cdots & column10 \end{bmatrix} = \begin{bmatrix} column1 \\ \vdots \\ column10 \end{bmatrix}$$

Equation 5-7

The scaling vector as a function of test number, Table 5-1, is shown in Figure 5-11. Note that this result was obtained by using noise-free, simulated experiments, with an assumed constant coupling region of $\sigma_x=\sigma_y=1$ pin separation distance. We can see that the scaling

factors remain approximately constant even for different target radii and different starting dies, i.e. different magnitude changes.

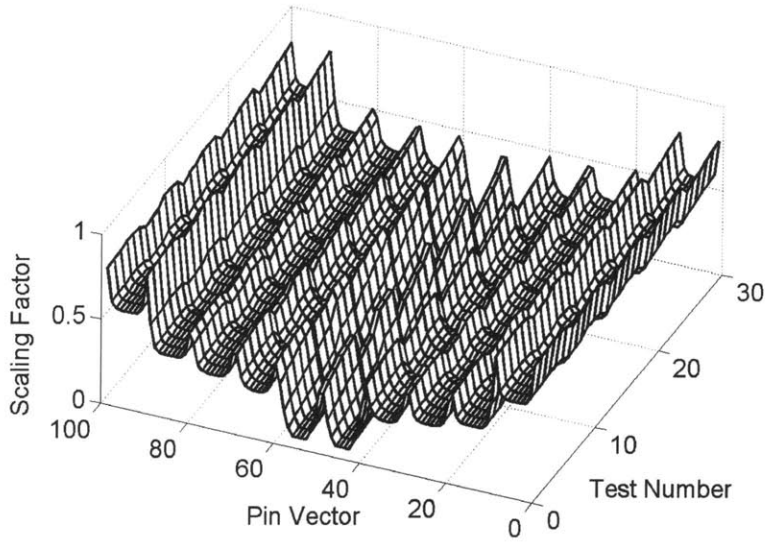


Figure 5-11 Pin scaling vector as a function of test number. Numbering defined in Table 5-1.

Because the scaling remains approximately constant, we can assemble a universal scaling matrix by averaging the available 30 scaling values for each of the 100 pins. This matrix is then used to compute all ten possible combinations of starting and target parts (since there is a total of five parts). The test numbering convention is shown in Table 5-2. The error between a "formed" and predicted part is shown in Figure 5-12. When one compares these results to the GIC algorithm results from Figure 5-8, one observes that this universal model does the best job thus far of predicting a part.

Starting part	Target part	Test number
1	2	1
1	3	2
1	4	3
1	5	4
2	3	5
2	4	6
2	5	7
3	4	8
⋮	⋮	⋮

Table 5-2 Standard test numbering convention for universal model testing.

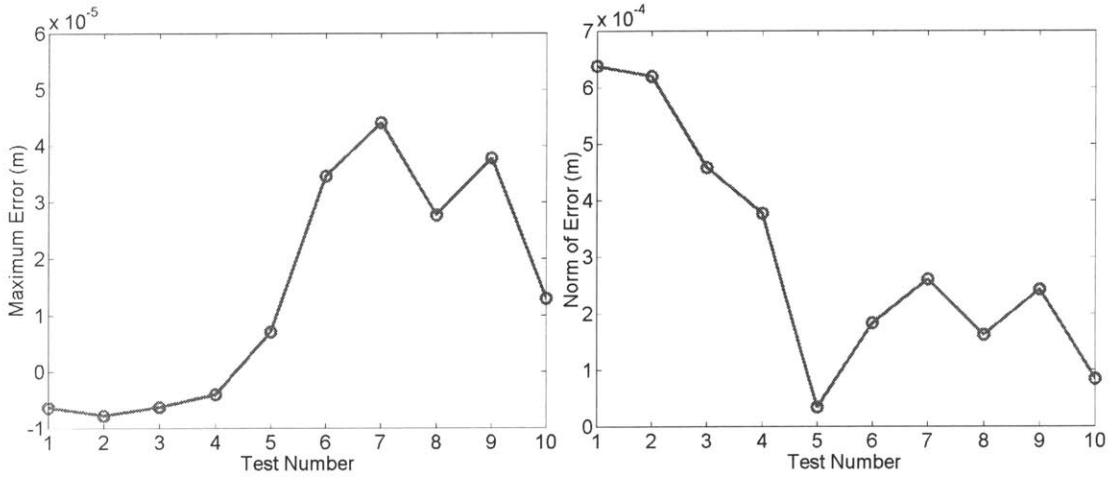


Figure 5-12 Error between formed and predicted part. Maximum error (left), Norm of error (right).

5.4 The GIC Model and the Matrix Condition Number

At the end of Chapter 3 we concluded that, once we know the plant, we can design the controller simply through matrix inversion. In Chapter 4, we saw that identifying a process-specific plant model is difficult. However, we performed a number of experiments with a $2x2$ process that showed a large tolerance for uncertainty in the lightly coupled example process. In this chapter, we presented a generic, physically intuitive plant model, GIC, which is easily calibrated for the class of processes which we are addressing. We have shown through numerical experiments and comparison with a detailed *ABAQUS* simulation that this model does a good job of predicting process output.

At this point, it appears that we can design the controller. However, because the controller relies on matrix inversion, we must check that the plant is easily inverted. To do this, we look at the condition number. The condition number is a good indicator of how close a matrix is to being singular, [31]. A number of approximately one is best. The condition number of an unscaled GIC matrix, based on the matrix Frobenius norm, as a function of matrix size is shown in Figure 5-13. We can see that the GIC matrix is not easily invertible. This result is also found to hold for other, non GIC, models which reflect regional coupling. As a result, we will pursue a different approach towards designing the controller in Chapter 7. Note that one of the benefits of using the "plant inverse" approach to designing the controller gain matrix was complete decoupling of the process outputs. This, however, already would not be possible in the physical process since we are using an approximate process model.

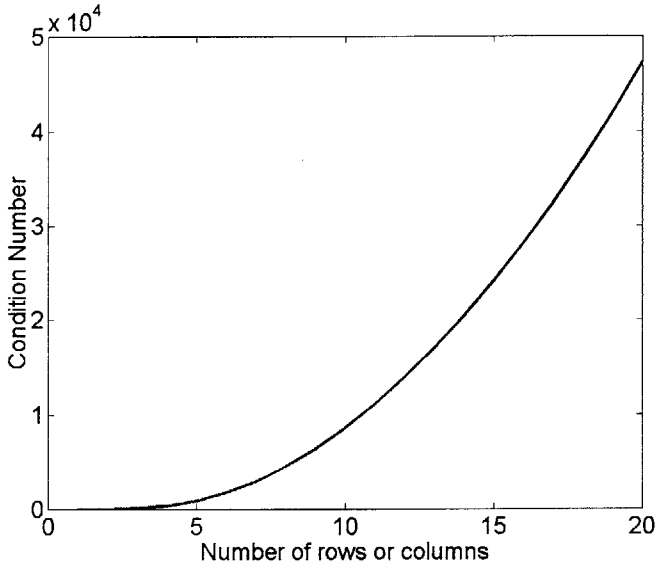


Figure 5-13 Matrix condition number (*Frobenius Norm*) as a function of the number of rows or columns in a square Gaussian matrix. Assumed coupling $\sigma = 1$, no scaling is used.

CHAPTER

6

ROBUST STABILITY AND PERFORMANCE

Although we are ready to design new controllers, we will first look at the proper consideration of robust stability and performance. We do this now because some controller methods allow the inclusion of stability or performance criteria at the design stage, rather than as a check afterwards. Our exploration of tolerance to uncertainty in Chapter 4 was a brute-force method which is not easily extensible to large MIMO processes. Instead of simulation, we turn to MIMO control theory. Robust stability and performance are both considered in the frequency domain through the use of singular values. After defining what is meant by uncertainty, we will look at the small gain theorem, which includes all representations of uncertainty at once. Then, we will customize the description of uncertainty to our particular problems. This will give us a better estimate of the stability bounds. Finally, we will take a brief look at how performance specifications may be included for MIMO processes.

6.1 Defining Uncertainty

In chapter 4, we attempted to identify the true process model. However, we quickly saw the difficulty and the lack of practicality in this approach because of the noise inherent in all processes and the large number of required experiments. Therefore, in Chapter 5, we developed a generic process model which gave us good results. Our

problem, as this stage, is how to define the difference between the true plant and our own model of the plant. In robust control terminology, our task is to define the uncertainty matrix Δ. Note that in the case of general plants, this Δ can be linear time invariant (LTI) or non-LTI. For the work considered in this thesis and the following discussions, we look at Δ as LTI. This is done to complement the description of the LTI process, much as LTI uncertainty is used to determine the stability and performance bounds of LTI SISO processes.

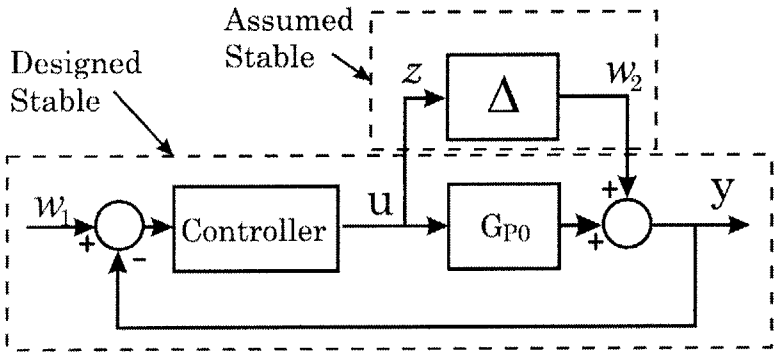


Figure 6-1 General block diagram, additive uncertainty.

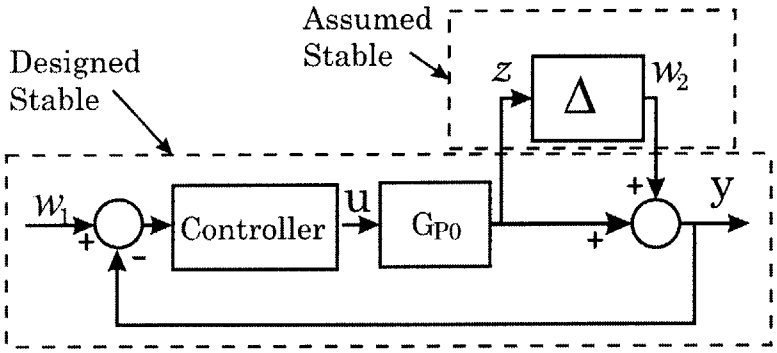


Figure 6-2 General block diagram, multiplicative uncertainty.

Consider a general block diagram with plant uncertainty, Figure 6-1 and Figure 6-2, where we have represented uncertainty as a Δ block. The uncertainty can be both additive, Figure 6-1, and multiplicative, Figure 6-2,

$$G_p(z) = G_{po}(z) + \Delta(z)$$

Equation 6-1

$$G_p(z) = G_{po}(z)(I + \Delta(z))$$

Equation 6-2

The choice between the two descriptions is often left to the designer and the particular process under consideration. However, the multiplicative form is often considered to be more appropriate as it makes the uncertainty scale with the plant. As an example, consider uncertainty in a spring's stiffness, k. We will use the lower case of Δ to denote the fact that δ is a scalar rather than a matrix. We can model the uncertainty, δ, additively as up to 0.1, regardless of the nominal value of stiffness, k_o, by

$$k = k_o + 0.1\delta$$

Equation 6-3

where δ is a real scalar with $|\delta| \leq 1$. Alternatively, we can model the uncertainty as multiplicative and up to 10% of the nominal value by

$$k = k_o(1 + 0.1\delta)$$

Equation 6-4

where, again, δ is a real scalar with $|\delta| \leq 1$. Note that, in both cases, δ is allowed to remain variable with a simple limit on maximum magnitude.

Looking the spring example, we have begun to describe exactly what we mean by the uncertainty model Δ. Ideally, Δ will describe exactly what we do not know. This can be parametric uncertainty, as with spring stiffness parameter, or a question of model fidelity. For example, a plant may act like a 2nd order process at low frequencies and like a higher, unknown order process at high frequencies. As it turns out, we have very little control over Δ itself. Uncertainty, Δ, can be only one of three types, real (non-frequency dependent), complex (frequency dependent), or a mixture of the two. For real, parametric uncertainty, we need only to identify its location, i.e. the parameter that is uncertain, and its magnitude. For frequency dependent uncertainty, we appear to have a harder problem.

After all, a natural question to ask is what order function do we pick to describe our uncertainty in the nominal plant. Here, our limitation in specifying Δ helps us out: we do not have control over the particular form of the frequency function. In other words, we can't specify the form of complex uncertainty; we can only designate it as complex. Therefore, when we compute our stability bounds, we will be guarding against instability caused by uncertainty described by a second-order or a twenty-second-order equation. In addition to being able to specify the type of uncertainty, we have the ability to specify the structure of the Δ matrix. This is shown in Example 6-1.

Example 6-1 mass-spring-damper

Consider the mass-spring-damper system, Figure 6-3, with 10% uncertainty in each of the elements, such that

$$m = m_0(1+0.1\delta_m) \qquad k = k_0(1+0.1\delta_k) \qquad c = c_0(1+0.1\delta_c)$$

Equation 6-5

where m_0, k_0, and c_0 are the nominal mass, stiffness, and damping, and each $|\delta| \leq 1$. Note that we have three separate uncertainties and that there is no cross-coupling in uncertainty.

Given a completely unstructured description of uncertainty, the Δ matrix will have the form

$$\Delta = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \delta_{1,3} \\ \delta_{2,1} & \delta_{2,2} & \delta_{2,3} \\ \delta_{3,1} & \delta_{3,2} & \delta_{3,3} \end{bmatrix}$$

Equation 6-6

where $\delta_{i,j}$ can be real or complex. This is the form of uncertainty which will be considered in Section 6.2 when we compute stability bounds according to the small gain theorem. To more accurately reflect the true nature of uncertainty, Equation 6-5, we can introduce structure into the Δ matrix. Because all three uncertainties are independent (no cross coupling), the Δ matrix will be diagonal. The structured Δ for this problem is specified as

$$\Delta = \begin{bmatrix} \delta_m & 0 & 0 \\ 0 & \delta_k & 0 \\ 0 & 0 & \delta_c \end{bmatrix}$$

Equation 6-7

where, additionally, we know that $|\delta| \leq 1$ and that they are limited to real values. This is the description used when applying μ analysis, Section 6.3. Note that we have not fixed a particular value for each δ, we have merely bounded their magnitudes. This δ are left variable in all Δ analyses since combinations of δ, not necessarily occurring at the boundaries of δ, can cause instability. Also note that in introducing structure into the Δ matrix, we are still not able to specify the exact form of each entry δ, we are merely able to enter 0 where there is no cross-coupling in uncertainty and we are able to designate each δ as real or complex.

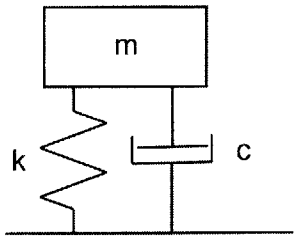


Figure 6-3 Mass-spring-damper system.

Although they do not strictly belong to the uncertainty Δ, we have quietly used the idea of weighting functions to fill-out our descriptions of uncertainty. For example, we have used a fixed weighting function of 0.1 in Example 6-1. The block diagram for just the uncertain mass element from Example 6-1 is shown in Figure 6-4. These weighting functions are a way to overcome some of the limitations in specifying Δ. For example, rather than a fixed value, a frequency-dependent weighting function can be used to reflect good plant knowledge at low frequencies and poor knowledge at high frequencies. To do this, we describe the δ as complex, and use a high-pass filter as a weighting function. Note that these weighting functions merely scale the magnitude, they still do not allow us to specify the equation that makes up δ. A more detailed treatment of the influence of these weighting functions on the computation of stability bounds is presented in Section 6.3.

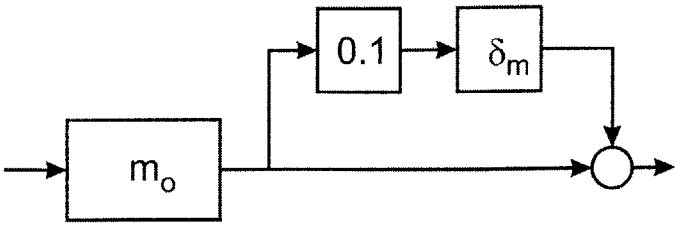


Figure 6-4 Uncertain mass block diagram.

Before we go on, we summarize what we can say about Δ. First, Δ can be real, complex, or a mixture of the two. We cannot specify the particular functions that make up Δ. Second, we can specify the structure of Δ by entering 0's where there is no cross-coupling in uncertainty. Third, we can scale the magnitude of Δ to describe its influence on the nominal plant.

6.2 Small Gain Theorem

The small gain theorem allows us to obtain an estimate of the boundary of process-destabilizing uncertainty based on the infinity norm. Although it can be found in most robust control texts, I credit both Zhou and Doyle, [32], and Maciejowski, [33], for my understanding of the theory.

The small gain theorem does not limit the form of Δ. Therefore, a full, complex Δ matrix is assumed in all cases, regardless of the true nature of uncertainty. The only control that the user has over Δ is its place within the block diagram, i.e. in defining the inputs and outputs of the Δ block. We saw this in Example 6-1 where we were forced to accept a full, complex Δ matrix, while being able to specify that the variation occurred within the mass, spring, and damper elements. Note that we were unable to specify how these three uncertainties interacted (if at all), Equation 6-6. If only the mass and spring varied and the damper was known exactly, the Δ matrix assumed for the small gain theorem would be a full 2*x*2 matrix

$$\Delta = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2} \end{bmatrix}$$

Equation 6-8

where each $\delta_{i,j}$ is allowed to be complex. Because no interaction structure is specified within the Δ matrix, these stability bounds are usually more conservative than those obtained through μ analysis.

We start our development by looking back at the block diagrams in Figure 6-1 and Figure 6-2. Here, w_1 is the reference (target) input, y is the process output, and, with the presence if the Δ block initially ignored, w_2 is treated as an unknown disturbance input and z is treated as a generalized error output. The process equation may be written as

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Equation 6-9

where T_{ij} is the closed-loop transfer function for the i^{th} input and j^{th} output. For additive uncertainty, Figure 6-1, these are defined as

$$\begin{bmatrix} T_{11} = G_{po}G_c\left(I + G_{po}G_c\right)^{-1} & T_{12} = \left(I + G_{po}G_c\right)^{-1} \\ T_{21} = \left(I + G_cG_{po}\right)^{-1}G_c & T_{22} = -\left(I + G_cG_{po}\right)^{-1}G_c \end{bmatrix}$$

Equation 6-10

and for multiplicative uncertainty, Figure 6-2, they are defined as

$$\begin{bmatrix} T_{11} = G_{po}G_c\left(I + G_{po}G_c\right)^{-1} & T_{12} = \left(I + G_{po}G_c\right)^{-1} \\ T_{21} = G_{po}G_c\left(I + G_{po}G_c\right)^{-1} & T_{22} = -G_{po}G_c\left(I + G_{po}G_c\right)^{-1} \end{bmatrix}$$

Equation 6-11

Note that, in calculating these transfer function we have made use of the following known matrix property: for compatible matrices A and B,

$$B\left(I + AB\right)^{-1} = \left(I + BA\right)^{-1}B$$

Equation 6-12

with A defined as the identity matrix and B defined as $G_{po}G_c$. For successful application of this property, the *product* $G_{po}G_c$ has to be square. This means that we specify as many target values as we have outputs. This is *not* the same as requiring G_{po} or G_c to be square, which would require an equal number of control inputs and plant outputs. Regardless of the block diagram used, when Δ is included, the closed-loop transfer function becomes, [33],

$$y = \left(T_{11} + T_{12}\Delta(I - T_{22}\Delta)^{-1}T_{21}\right)w_1$$

Equation 6-13

Note that we have used Equation 6-12 to obtain this result with T_{22} and Δ square, by definition. Because we designed the controller for the nominal plant, we know that the closed-loop transfer functions (T_{11}, T_{12}, T_{21}, T_{22}) are stable in the nominal process. We will assume that process uncertainty, Δ, is itself stable. Therefore, we need look only at the inverse term in Equation 6-13. Note that Δ is used to model those parts of the process that are unknown. If it is allowed to be unstable, and we do not know about it, the closed-loop process will be unstable no matter what controller we design for the known part. Our estimation of full closed-loop stability will depend solely on the interconnection of T_{22} and Δ according to the generalized Nyquist theorem, [33]:

The feedback process can become unstable only if one or more of the characteristic loci of $-T_{22}\Delta$ *encircles the point -1*

If we had a precise definition of Δ, we could conclude stability from just that statement. However, since we don't, we have to use a slightly modified criterion. The spectral radius, ρ, is defined as the maximum modulus of the eigenvalues, [32],

$$\rho(T_{22}\Delta) = \max_i |\lambda_i(T_{22}\Delta)|$$

Equation 6-14

where λ_i is the i[th] eigenvalue. The Nyquist criterion will therefore be satisfied as long as

$$\rho(T_{22}\Delta) < 1$$

Equation 6-15

It can be shown that, [32],

$$\rho(T_{22}\Delta) \le \bar{\sigma}_{sv}(T_{22}\Delta)$$

Equation 6-16

where $\bar{\sigma}_{sv}$ is the maximum singular value. The subscript *sv* is used to differentiate this symbol from the σ which refers to the process standard deviation. Because of Equation 6-16, the generalized Nyquist theorem will still be satisfied if

$$\bar{\sigma}_{sv}(T_{22}\Delta) < 1$$

Equation 6-17

Now, we use the definition of the infinity norm

$$\|G\|_\infty = \sup_\omega \bar{\sigma}_{sv}(G(j\omega))$$

Equation 6-18

where sup is the supremum. The supremum is the least upper bound, defined as the level, L, such that no member of the set $\bar{\sigma}_{sv}(G(j\omega))$ exceeds L. Therefore, we can rewrite Equation 6-17 using Equation 6-18 as

$$\|T_{22}\Delta\|_\infty < 1$$

Equation 6-19

which will satisfy the Nyquist theorem. Although we have a precise definition of T_{22}, we do not have a precise definition of Δ. Since Equation 6-19 relies on their product, and T_{22} and Δ may be matrices, we need another property of the infinity norm:

$$\|T_{22}\Delta\|_\infty \le \|T_{22}\|_\infty \|\Delta\|_\infty$$

Equation 6-20

Now, we can finally state the Small Gain Theorem for stability:

- *If* $\|T_{22}\|_\infty \leq \gamma$
- *Then, a sufficient condition for stability is* $\|\Delta\|_\infty < 1/\gamma$

Restated, the small gain theorem says that if Equation 6-20 is an equality and T_{22} has infinity norm equal to γ, there is *at least one* matrix Δ with infinity norm equal to $1/\gamma$that will cause instability; the infinity norm of Δ must be lower than $1/\gamma$to guarantee robust stability. Note that this means that not every Δ matrix with infinity norm equal to $1/\gamma$will cause instability. This is most likely due to the fact that we have reduced all realizations of Δ matrices to a single value, the infinity norm. Because a Δ that fails the stability bound is not guaranteed to cause instability, we should view the small gain theorem bound as the dividing line between robust stability and non-robust stability rather than between stability and instability. The choice to exceed the robust stability bound should be made based on the cost of going unstable. If all forms of Δ are allowed (unstructured real and complex functions) then the small gain theorem is both necessary and sufficient to determine robust process stability.

6.3 Stability through μ analysis

Thus far, we have allowed the uncertainty matrix Δ to posses any structure and contain both real and imaginary numbers. For this purpose, the small gain theorem is a good starting point for robust stability of MIMO processes. Although it proves to be both necessary and sufficient when all realizations of Δ are possible, it can be overly conservative if the properties of Δ are known. This is shown in Example 6-2.

Example 6-2 Structured Δ

Consider a single input-single output closed-loop process with uncertainty in the pole location, as shown in Figure 6-5.

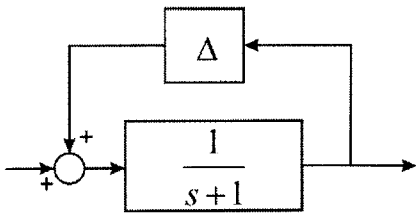


Figure 6-5 Example closed-loop process block diagram.

The calculated infinity norm of the nominal closed-loop plant is 1, which would require that $\|\Delta\|_\infty \leq 1$ for stability if all realizations of Δ are admissible. This can be compared to standard control analysis by performing a little block-diagram algebra to reduce the plant description in Figure 6-5 into a single transfer function:

$$\frac{Output}{Input} = \frac{1}{s+(1-\Delta)}$$

Now assume that it is known that Δ can assume only real values. It can readily be seen that Δ<1 for asymptotic stability, which matches the small gain theorem result. However, notice what happens if we assume that Δ can take on only *negative* real values. Now, with this constraint (which might be interpreted as a phase constraint of -180°), the magnitude of Δ can be made *arbitrarily large* without loosing stability. Thus, the small gain theorem can be shown to be overly conservative when certain properties of Δ are known.

As a means of taking advantage of any knowledge of Δ, we use structured singular values, or μ analysis. This will allow us to narrow the definition of Δ, as we did in Example 6-1 going from Equation 6-6 to Equation 6-7. The structured singular value of a matrix T_{22} is computed with respect to a set Δ of user specified perturbations and is defined as

$$\mu_\Delta(T_{22}) = \frac{1}{\min\{\bar{\sigma}_{sv}(\Delta^*) : \Delta^* \in \Delta, \det(I - T_{22}\Delta^*) = 0\}}$$

Equation 6-21

unless no Δ^* inside the set Δ makes $|I-T_{22}\Delta^*|=0$, then $\mu_\Delta(T_{22})=0$, [32]. Expressed in words, Equation 6-21 says that μ_Δ is equal to the inverse of the smallest maximum singular value

of Δ^*, which is a particular realization of uncertainty belonging to the allowable set Δ, that causes $I\text{-}T_{22}\Delta^*$ to be singular and to fail the Nyquist stability criterion. Note that we must therefore specify the structure of Δ before we can compute the structured singular value and that the structured singular value is a function of frequency, since at least T_{22} is a function of frequency.

We introduced the idea of defining Δ in Section 6.1 where we saw that we could define elements as real or complex and that we could specify the structure by placing 0's in the Δ matrix when the uncertainties were uncoupled. The weighting values, also discussed in Section 6.1, are absorbed into the description of the plant and are not strictly considered a part of Δ. We slightly augment the previous discussion in this section. Towards specifying Δ, we note that some noises can be independent, while others are not. In the case where the relationship between the unknown noises is not known, we are still able to enter 0's in the full Δ matrix, reducing it to a block-diagonal, rather than a pure diagonal, structure. The Δ matrix would thus look like

$$\Delta = \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \Delta_3 \end{bmatrix} = \begin{bmatrix} \delta_1 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 \\ 0 & 0 & \begin{bmatrix} \delta_{1,1} & \delta_{1,2} \\ \delta_{2,1} & \delta_{2,2} \end{bmatrix} \end{bmatrix}$$

Equation 6-22

If their relationship is known, for example if the mass and spring variation were related and the damper variation is independent, we are able to reduce the Δ matrix further to obtain

$$\Delta = \begin{bmatrix} \Delta_{m,k} & 0 \\ 0 & \delta_c \end{bmatrix} = \begin{bmatrix} \delta_{m,k} I & 0 \\ 0 & \delta_c \end{bmatrix}$$

Equation 6-23

Interested readers are referred to Maciejowski, [33], and Zhou and Doyle, [32], for additional details on specifying Δ.

The stability criterion can be re-expressed in terms of the structured singular value. It can be shown that, [32],

$$\rho(T_{22}) \le \mu_{\Delta}(T_{22}) \le \overline{\sigma}_{sv}(T_{22})$$

Equation 6-24

where, as before, ρ is the spectral radius and $\overline{\sigma}_{sv}$ is the largest singular value. Because it is closer to the spectral radius than $\overline{\sigma}_{sv}$, the structured singular value, μ_{Δ}, gives a less conservative robust stability bound. To develop the μ_{Δ} stability criterion, we follow the same development as for the small gain theorem, Equation 6-16 through Equation 6-20. For a given structured uncertainty matrix Δ the new stability criterion becomes, [32, 33],

- *If* $\|T_{22}\|_{\mu} \le \gamma$
- *Then, the condition for stability for all allowable* Δ *is* $\|\delta_i\| < 1/\gamma$, *where* δ_i *is a diagonal entry, block*(Δ_i) *or scalar* (δ_i) *within the* Δ *matrix.*

where $\|T_{22}\|_{\mu} = \sup_{\omega} \mu(T_{22})$. Restated, this new robust stability condition says that given a maximum μ value of γ, each subset δ_i of an allowable Δ must have an infinity norm of less than $1/\gamma$ to guarantee robust stability. Allowable Δ are defined as those Δ which have the specified structure, as did Equation 6-7 in Example 6-1 or the Δ in Example 6-2. As with the small gain theorem, these bounds differentiate between robust stability and non-robust stability rather than between stability and instability.

With this new stability criterion, we may think that we are ready to compute the stability margins of arbitrarily-sized processes. However, no algorithms currently exist for computing the exact values of γ for processes with more than three blocks inside of the Δ matrix. Remember that γ is a function of frequency, Equation 6-21. Because we want to address processes with uncertainty size greater than three, we are forced to calculate the upper and lower bounds of γ, γ_{upper} and γ_{lower}. According to the *MATLAB* reference, [34], given $\mu_{\Delta}(T_{22}) \leq \gamma$ (Section 6.3) the perturbed process is stable for all admissible Δ satisfying

$$\max_{\omega} \overline{\sigma}[\Delta(j\omega)] < \frac{1}{\gamma_{upper}}$$

Equation 6-25

and there exists a particular admissible perturbation matrix Δ satisfying

$$\max_{\omega} \overline{\sigma}[\Delta(j\omega)] = \frac{1}{\gamma_{lower}}$$

Equation 6-26

that causes instability. Here, γ_{upper} and γ_{lower} are the upper and lower bound of γ. Therefore, we determine robust stability of a closed-loop process by calculating the upper and lower bounds of γ as a function of frequency, taking the maximum value of γ_{upper}, and inverting it to obtain the maximum bound on the infinity norm of an allowable Δ. An example of such a computation using the *MATLAB* function "mu" is shown in Figure 6-6. To successfully apply this function, we need to know the controller, plant, and uncertainty matrix and their interconnections; the shown bounds correspond to the nine input-nine output case discussed in Example 6-5. From Figure 6-6 we estimate the maximum value of γ_{upper} to be ~8.5, meaning that the maximum infinity norm of δ_i can be 0.12. In this plot, we also see that the bounds of γ are 0 for the frequency range of 1.5

rad/s to π rad/sec. From Equation 6-21 and the subsequent discussion, we see that this means that no allowable Δ exists in that frequency range that causes the process to go unstable.

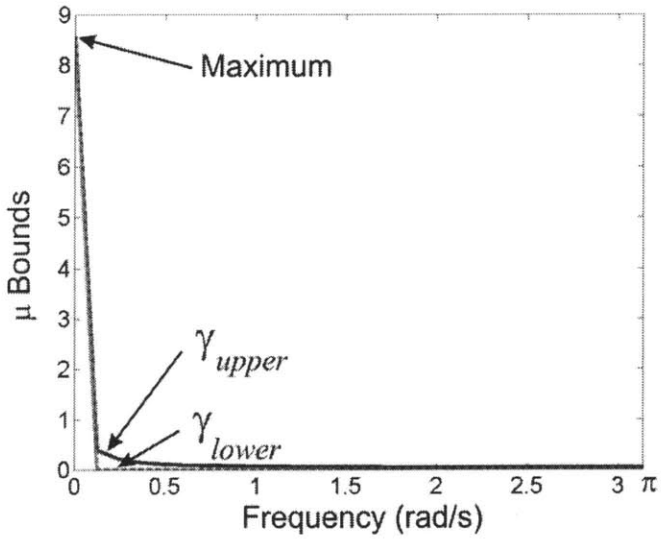


Figure 6-6 Example μ bounds (γ_{upper} and γ_{lower}) as a function of frequency. Example process is the nine input-nine output case in Example 6-5. Since this is a discrete-time process, the needed frequency range is [0,π].

In Example 6-1, we decided to scale each of the uncertainty δ's. This was done because μ analysis will calculate the largest allowable δ that can be in the closed-loop process, given our particular description. The scaling, therefore, allows us to obtain a good sense of allowable uncertainty even when considering quantities of vastly different magnitudes. This is explained through two *very* simple examples.

Example 6-3 The benefit of scaling Δ

Consider an uncoupled CtC process model with the following nominal plant model

$$K_{Po} = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.1 \end{bmatrix}$$

and an integral controller with $K_c=I$. Since this is just two uncoupled SISO plants, we could draw conclusions about allowable variation in each matrix entry without performing μ analysis. This was done in Chapter 2 to show that the asymptotic stability limit is $0<K_pK_c<2$. For the purposes of this example, however, we will analyze the two SISO processes as if they were one MIMO process.

Suppose that we know a little bit about uncertainty such that the real plant gain, K_p is defined by:

$$K_p = K_{po} + K_\Delta \Delta = K_{po} + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}$$

with independent scalar δ's with

$$\|\delta_i\|_\infty < 1 \quad for \quad i = 1,2$$

Note that this means that we have bounded the uncertainty, $K_\Delta\Delta$, at 10% of its nominal value, K_{po}. By observation of SISO stability bounds, we know that the maximum allowed variations are:

$$\begin{bmatrix} -1 & 0 \\ 0 & -0.1 \end{bmatrix} < K_\Delta \Delta < \begin{bmatrix} 1 & 0 \\ 0 & 1.9 \end{bmatrix}$$

if K_Δ is included in the calculation of μ bounds, we obtain:

$$\|\delta_i\|_\infty < 10$$

such that

$$K_P = K_{po} + K_\Delta \Delta = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

if it is left out, we obtain:

$$\|\delta_i\|_\infty < 0.1$$

such that

$$K_P = K_{po} + \Delta = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} + \begin{bmatrix} -0.1 & 0 \\ 0 & -0.1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0 \\ 0 & 0 \end{bmatrix}$$

Both of these limits are at the lower boundary of the (2,2) entry in the plant gain matrix. However, scaling allows the use of the full range of uncertainty in the (1,1) entry; $0<K_{p(1,1)}<2$ for scaled Δ, or $0.9<K_{p(1,1)}<1.1$ for the unscaled case.

Example 6-4 The disadvantage of scaling Δ

This time, consider an uncoupled CTC process model with the following nominal plant model

$$K_{Po} = \begin{bmatrix} 1.3 & 0 \\ 0 & 0.8 \end{bmatrix}$$

and an integral controller with $K_c=I$. As in Example 6-3, the asymptotic stability limit is $0<K_PK_c<2$. Suppose that our knowledge of uncertainty is defined by:

$$K_P = K_{Po} + K_\Delta \Delta = K_{Po} + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix}$$

again, with independent scalar δ's with

$$\|\delta_i\|_\infty < 1 \quad for \quad i = 1,2$$

This represents a 10% uncertainty on the (1,1) entry and a 50% uncertainty in the (2,2) entry. By observation of SISO stability bounds, we know that the maximum allowed variations are:

$$\begin{bmatrix} -1.3 & 0 \\ 0 & -0.8 \end{bmatrix} < K_\Delta \Delta < \begin{bmatrix} 0.7 & 0 \\ 0 & 1.2 \end{bmatrix}$$

if K_Δ is included in the calculation of μ bounds, we obtain:

$$\|\delta_i\|_\infty < 1.6$$

if it is left out, we obtain:

$$\|\delta_i\|_\infty < 0.7$$

If both δs assume the maximum variations at the same time. this translates to the following allowable variations in K_P

$$\begin{bmatrix} 1.14 & 0 \\ 0 & 0 \end{bmatrix} < K_P < \begin{bmatrix} 1.46 & 0 \\ 0 & 1.6 \end{bmatrix}$$

with scaling, or

$$\begin{bmatrix} 0.6 & 0 \\ 0 & 0.1 \end{bmatrix} < K_P < \begin{bmatrix} 2.0 & 0 \\ 0 & 1.5 \end{bmatrix}$$

without scaling. In this example, *not* using our knowledge of uncertainty actually allows us to compute a larger range of uncertainty in the (1,1) parameter, with little degradation in the boundaries for the (2,2) parameter.

Example 6-3 and Example 6-4 showed two cases where using knowledge of uncertainty can help or hurt in the calculation of a μ-based boundary. Of course, we can turn a critical eye on these examples. First, scaling or normalizing δ_i reflects our best engineering guess about the magnitude of uncertainty. As such, it should be included as an intelligent way of describing the process. Second, these were very simple (artificial) cases where we could easily compute stability boundaries through other methods. Most of the time, μ analysis is applied precisely because there does not exist an alternate method. It is in this context that μ analysis really shines.

Since the structure of Δ can make a big difference, a logical question to ask is whether any information exists about the uncertainty structure of the processes that are being addressed in this thesis. This can be answered by looking at a block diagram. Figure 6-7 shows the process with uncertainty in place. Note that we have chosen the multiplicative

form of uncertainty. Alert readers may refer back to Figure 1-5 and raise the question of uncertainty within each of the machine actuators. Although this uncertainty can be included, it is not considered in this thesis. Therefore, the only uncertainty that exists within our class of systems is caused by imperfect knowledge of the plant gain. Remember that our inability to know this plant gain exactly is what motivated the development of the generic plant model. In addition to knowing the specific location of the uncertainty within the block diagram, we know that the uncertainty is limited to *real* numbers, since, by definition of cycle-to-cycle, no dynamics are allowed beyond a simple delay. Thus, we are limited to parametric uncertainty rather than "unmodeled dynamics," which plague most process model simplifications. A first attempt at applying uncertainty to our class of process is shown in Example 6-5.

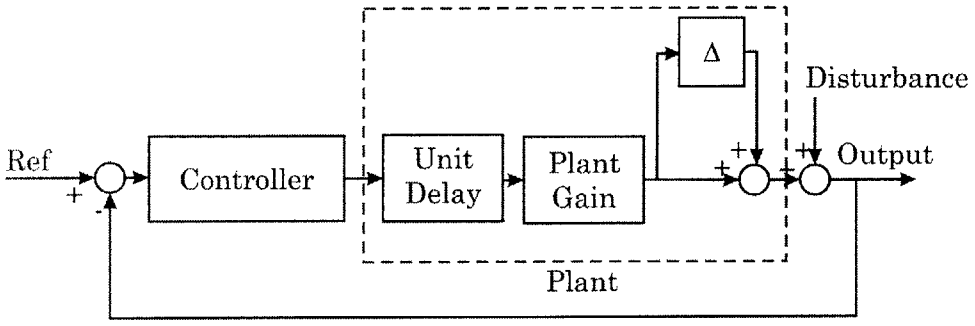


Figure 6-7 CtC block diagram with process uncertainty.

Example 6-5 μ-bounds with fully independent uncertainty

The process (plant) is defined as an *nxn* grid arrangement of inputs and outputs, resulting in n^2 inputs and n^2 outputs. This may be a square bundle of pins and measurement points for our sheet metal forming example or a set of heaters and temperature points for a plate-heating process. We will assume that the coupling is unity in all directions such that $\sigma_x=\sigma_y=1$ for our process model. For these initial trials, we will also leave the model unscaled.

Uncertainty is represented as a percentage of each plant matrix entry, such that the ij^{th} entry in the true process is defined as

$$(K_P)_{ij} = (K_{Po})_{ij}(1+0.1\delta_{ij})$$

where

$$\left\|\delta_{ij}\right\|_\infty < 1$$

and each δ_{ij} is constrained to be a real number. Note that we will limit the precision of the K_{Po} matrix by defining the smallest, non-zero entry as 0.0002, thus some of the entries in the plant matrix will be zero and, based on our assumption, will not have uncertainty associated with them.

The controller is defined to be an integral controller with an uncoupled, unity gain matrix. The controller will thus have the form;

$$K_c = \frac{z}{z-1} I$$

where I is the identity matrix and is sized according to the plant.

Stability bounds are computed through both the small gain theorem and through μ analysis. Results are presented in Table 6-1; the calculation code is available in Appendix F. We can observe two trends. First, as n^2 increases, so does the percentage benefit of computing the stability bounds through μ analysis. Second, the maximum value of the allowable perturbation is shrinking. Because all combinations of allowable uncertainty must be considered when computing the stability bounds, this shrinking is most likely due to our overly-broad definition of uncertainty as independent at each point, i.e. small uncertainty contributions from enough points eventually add-up to a significant value.

# of inputs n^2	# of outputs n^2	# of δs	Small Gain Theorem Bound $\left\|\Delta\right\|_\infty <$	μ Bound $\left\|\delta_{ij}\right\|_\infty <$	% improvement
4	4	16	0.55	0.59	5.9
9	9	81	0.09	0.12	22.7
16	16	256	0.027	0.041	34.1
25	25	621	0.012	0.02	41.2

Table 6-1 Small gain theorem and μ-analysis bound comparison for uncertainty in the plant matrix.

Example 6-5 showed the influence of process size on the stability bounds when each value in the matrix was allowed to vary independently. Although this independence may be a reasonable first-shot assumption, it is not practical. First, even though the percent improvement is growing, the actual size of the allowable gain error is shrinking rapidly; even at the 5x5 process size, the absolute error in gain can only be 0.002 (0.1*0.02=0.002) in a matrix where the maximum entry is 0.159. If this shrinking continues, even at a slower rate, the lab-scale process may have small gain theorem and μ-analysis bounds which are practically indistinguishable and overly conservative.

Second, the computation time grows geometrically with process size from fractions of a second for a 2x2 process to over two hours for a 5x5 process when calculated on a Pentium 4, 2 GHz computer. This does not bode well for even a lab-scale process which is 10x11. From a practical point of view, although we have a good idea about the location and the form of the uncertainty, real numbers, we do not know the uncertainty matrix structure. For instance, we may assume that there is separate uncertainty in each of the nonzero entries within the process matrix, as we do in Example 6-5, or we can assume that variations in some of the values are actually coupled together and should have only a single δ. To this effect, we will assume that we have gotten the general shape of regional coupling *mostly* correct and assume uncertainty in the scaling matrix, Equation 5-4. Note that the scaling matrix scales and stretches the initial Gaussian shape according to experimental data. We have made this exact assumption in Section 5.3, where we fixed the coupling region and recalculated the scaling factors for different calibration parts. The effects of using this approach are presented in Example 6-6. Because of the good results obtained therein, we will use this description of uncertainty in future tests.

Example 6-6 μ-bounds with scaling matrix uncertainty

Rather than assuming independent uncertainty in every single value of *Kp*, as in Example 6-5, we assume that uncertainty is propagated through the scaling matrix. We will bound our nominal uncertainty at 10% of the scaling matrix entries. This bound is derived from simulation experiments presented in Section 5.3, Figure 5-11. Therefore, the plant gain matrix is

$$K_P = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_{n^2} \end{bmatrix} \left([I] + 0.1 \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \delta_{n^2} \end{bmatrix} \right) \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

Equation 6-27

with

$$\|\delta_i\|_\infty < 1$$

As before, we will use an integral controller with an identity matrix as its controller gain. The outcome of computing the bounds for 2x2 through 5x5 processes is shown in Table 6-2. The corresponding simulation code is available in Appendix G. Stability bound computations took under a second. From the table, we see that the μ bound is computed to be 10 for all cases, translating to allowing each scaling value s_i to vary by ±100%.

# of inputs n^2	# of outputs n^2	# of δs	μ Bound $\|\delta_i\|_\infty <$
4	4	4	10
9	9	9	10
16	16	16	10
25	25	25	10

Table 6-2 μ analysis bound for uncertainty in the plant matrix.

Before we continue with more general theory, let us look at exactly what a Δ matrix means to us. As we have previously stated, Δ is solely used to represent the uncertainty in the real-valued gain matrix describing our plant. We have also selected an uncertainty description that allows the scaling values to vary by a percentage of their nominal values, while the keeping the coupling matrix unchanged. Note that, since the two are multiplied in the plant model, the whole plant model varies. Our uncertainty description, therefore, says that the real plant model is expressed as Equation 6-27, shown further developed as:

$$K_P = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_{n^2} \end{bmatrix} \left([I] + 0.1 \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \delta_{n^2} \end{bmatrix} \right) \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

$$= \begin{bmatrix} s_1 + 0.1\delta_1 s_1 & 0 & 0 & 0 \\ 0 & s_2 + 0.1\delta_2 s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_{n^2} + 0.1\delta_{n^2} s_{n^2} \end{bmatrix} \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

Equation 6-28

In addition to describing the ability to tolerate uncertainty, Δ can be used to determine if we can switch target shapes without having to recalibrate the plant model; if the resulting

change in the scaling values is small enough, the process will not lose stability. Again, the first steps towards this were taken in Section 5.3, where we looked at the variation in the scaling values for cylindrical target shapes with different calibration parts and widely ranging radii; we fixed the shape of coupling and recalculated the scaling factors for different calibration pairs. Figure 5-11 is particularly relevant as it shows the variations in these exact scaling factors. As mentioned in Example 6-6, the level of 10% variation is derived exactly from the scaling factors presented in that figure.

6.4 Performance as stability with μ analysis

Thus far, we have concentrated on uncertainty solely as it affects the stability of the closed-loop process. Although this is our primary concern, we also want good performance. As it turns out, we can analyze robust performance by re-casting the problem as a stability test, [32, 33, 35]. To do this, we follow the development presented in Dahleh et al., [35].

Suppose that we wish to see if a closed-loop process is robust to an output disturbance represented by ξ and filtered by W as in Figure 6-8. Here, $\|\xi\|_2 < 1$ and the filter, W, is designed to capture the frequency characteristics of the disturbance and is scaled according to our performance specification. To achieve robust performance, we must have

$$\left\|(I + G_P G_C)^{-1} W\right\|_\infty < 1$$

Equation 6-29

which is a criterion on the closed-loop transfer function from ξ to y. This looks like the criterion we used in the development of the small gain theorem and, in fact, it is just that.

Thus, we see that our performance requirement is equivalent to the robust stability problem presented in Figure 6-9.

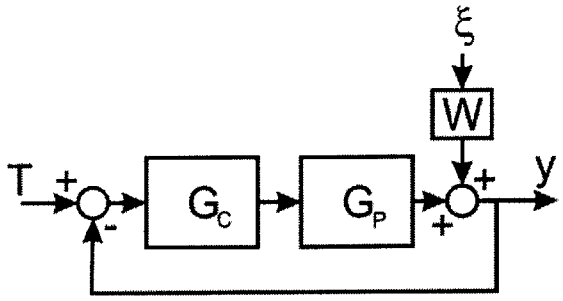


Figure 6-8 Feedback block diagram with output disturbance. Figure reproduced from [35].

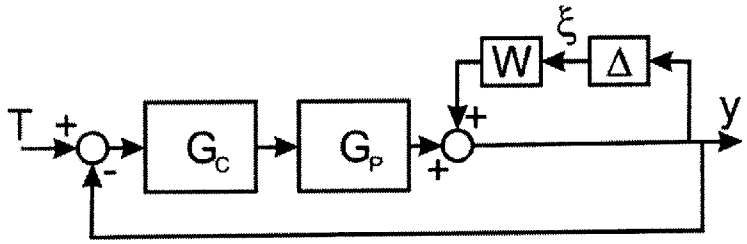


Figure 6-9 Feedback block diagram with performance as stability. Figure reproduced from [35].

To test for robust performance, we must use structured singular values. Given an uncertain plant and performance requirements, we can put together a block diagram as shown in Figure 6-10. Here, we see that Δ_1 and W_1 describe the stability and Δ_2 and W_2 describe the performance. To test whether the nominal plant as well as the perturbed plants meet the performance specification, we define the uncertainty matrix as

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$$

Equation 6-30

where we can see that the uncertainty matrix is block diagonal. Now, we simply compute the singular values as we did for the stability tests. If the computed maximum structured singular value is greater than one, all plants described by Δ_1 and W_1 do not meet the performance requirements.

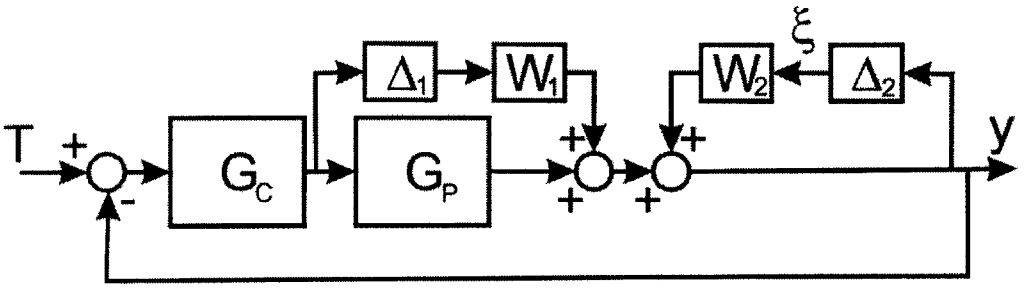


Figure 6-10 Feedback block diagram with mixed stability and performance.

6.5 Robust Stability and Performance Conclusions

In this chapter we have seen how to use the singular values to determine if a process is closed-loop robust to uncertainty. We have seen that the analysis of robustness relies on the use of the generalized Nyquist theorem and that we can get a better estimate of the stability bounds by defining the uncertainty as best we can. Finally, we demonstrated that we can test for robust performance by framing our requirements as a stability test. The tools developed in this chapter will allow us to better design and analyze MIMO controllers.

Before we close this chapter, we take a final look at what we mean by uncertainty and the matrix Δ. Our knowledge of Δ can take the form of magnitude limits, phase constraints, or matrix structure, among others. However, we are limited in the way we can express this knowledge according to the following:

1. We can specify the matrix structure of Δ by inserting zeros into this matrix when the uncertainties are uncoupled. Therefore, in Example 6-1 we were able to define the Δ as

$$\Delta = \begin{bmatrix} \delta_m & 0 & 0 \\ 0 & \delta_k & 0 \\ 0 & 0 & \delta_c \end{bmatrix}$$

instead of a full matrix.

2. We are able to designate each block diagonal section of Δ as real or complex, frequency dependent. For example, in Equation 6-30, Figure 6-10, we could say that the plant only has real, parametric, uncertainty and that the performance "uncertainty" is complex. Then,

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$$

with Δ_1 real and Δ_2 complex.

These two statements encompass all that we can specify about Δ itself. We cannot specify the actual complex function that makes up the block Δ_2. As a partial workaround, we introduced weighting functions connected to Δ. These weights were the subject of Example 6-3 and Example 6-4. These may be real, as was the case in the examples, or complex. Frequency weighting is especially useful if we have a plant that we know well at some frequencies, but not others. Note that, even with these weights, we are simply bounding the magnitude of the Δ matrix, not specifying the functions that make up its element. That's all the control we have, zero or not, real or complex, weighted or not.

We make special note to repeat the conclusion of Section 6.2: just because a matrix Δ fails our infinity norm test, it does not mean that it will necessarily cause the process to go unstable. The infinity norm test just tells us that there exists *at least one* matrix Δ with a failing infinity norm which will cause instability. These statements hold for both unstructured and structured uncertainty. Therefore, the norm-bound is an expression of the dividing line between robust stability and non-robust stability. Our choice of operating within this region should be based on risk-aversion to failure.

CHAPTER

7

CONTROLLER DESIGN AND EXPERIMENTS

At the end of Chapter 5, we saw that the method of designing a controller gain matrix through plant matrix inversion is not practical because of a large condition number for even moderately sized matrices. A large condition number was also noted to be a characteristic of other plant models that describe regional coupling. As a result, we are forced to pursue alternative controller approaches; ones that do not rely on taking the inverse of the plant matrix.

A search through the literature has revealed three candidate design methods the Linear Quadratic Regulator (LQR), the Linear Quadratic Gaussian or H_2, and the H_∞ controller. We select these controller design methods because they are a good starting point for MIMO controller design, just like the integral, proportional, and derivative controllers are good starting points for SISO controller designs. Other controllers may be built off of the groundwork laid in this thesis. However, these could be the subject of another thesis and will not be addressed here. Each controller will be considered in-turn via three sections. First, general controller theory will be addressed. Second, a specific controller will be designed for the discrete-die sheet metal forming process discussed in 1.5.1. Finally, experimental verification of the expected (simulated) response will be carried out on the

discrete-die sheet metal forming process. Because we will try to mimic the experimental procedure in simulation, we take a look at this first.

7.1 Application and Experimental Procedure

As we introduced in Chapter 1, our experimental platform is the discrete-die sheet metal forming process described in Section 1.5.1. We will be using a $10x11$ array of input pins located at the center of the reconfigurable die, see Figure 1-9. Therefore we will have 110 inputs, and we will consider 110 outputs, points on the sheet metal just above the input pin centers. All of our experiments will be carried using a target shape of a 27 cm radius cylinder. Additional experimental parameters are listed in Table 7-1.

Parameter	Value
Material	Al 2024-0, 0.16 cm thick, 14 cm x 49 cm
Preload	6,125 lb
Safety Load	8,500 lb
Target Shape	Cylinder, 27 cm radius
Interpolator	Elvax 360 (1.36 cm thick), covered with two layers of Teflon
Control Mode	Force control: pre-stretch, wrap, no post stretch

Table 7-1 Experiment parameters for the discrete-die sheet metal forming process discussed in Section 1.5.1.

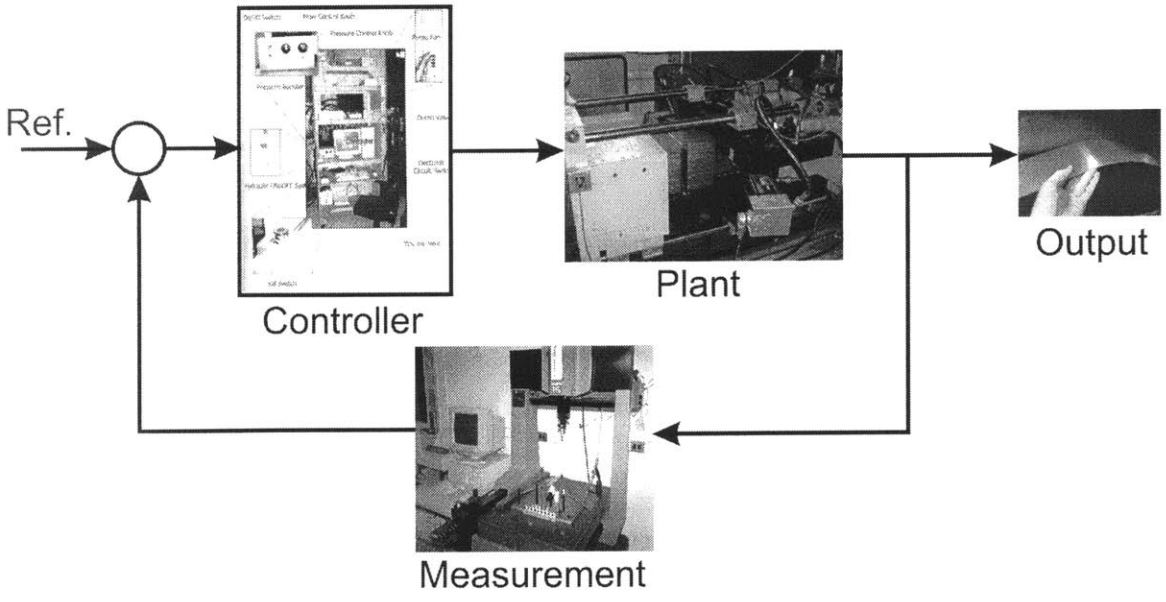


Figure 7-1 Discrete die sheet metal forming block diagram. Figure adapted from Norfleet, [30].

The block diagram for the discrete-die sheet metal forming process is shown in Figure 7-1. We see that the control loop is characterized by three components: the controller, plant, and measurement. The controller is implemented on a computer and it takes-in the error vector ($110x1$) and outputs the die vector (die shape). The controller logic includes a start-up procedure for a new forming run. The procedure is as follows

- First run: since the error vector does not yet exist and the generic process model has not yet been calibrated, the die shape is simply the reference shape.
- Second run: now we have an error vector, but a calibrated plant model still does not exist. Therefore, the error vector is simply added to the first die and this sum is used as the second die shape.
- Third run: now we have both an error vector and the ability to calibrate the plant model. Note that we are assuming a generic unity coupling region and, thus, only need to compute the diagonal scaling vector. First, both the parts and the dies are offset by three standard deviations of their measurements. This is done because some measurements may be close to zero and have a poor signal to noise ratio, [38, 39, 45]. Therefore, the first two parts and dies are redefined as

$$part_i = part_i + 3\sigma_{part_i}$$

Equation 7-1

Then, the i^{th} scaling value is computed as

$$s_i = \frac{\left(part_2 - part_1\right)_i}{\left(\begin{bmatrix} unscaled \\ Gaussian \\ matrix \end{bmatrix}\left(die_2 - die_1\right)\right)_i}$$

Equation 7-2

This may be done in *MATLAB* syntax as

$$scale = (part_2 - part_1)./\left(\begin{bmatrix} unscaled \\ Gaussian \\ matrix \end{bmatrix}(die_2 - die_1)\right)$$

Equation 7-3

Once the Gaussian influence coefficient model is thus calibrated, it is used to design the feedback controller. This, in our case, is the LQR, LQG, or H_∞ controller. Once designed, this controller is used to determine the third die input.

- Fourth and subsequent runs: the error vector is passed to the controller and the feedback controller designed at the third run is used to determine the next die.

The next segment is the stretch forming machine (plant) and it uses local position controllers to set up the discrete die according to the die vector specified by the main controller. An interpolator (elastic medium) is then placed over this assembled die. This is done to spread-out the influence of each pin over a larger area; otherwise, a sheet metal part stretched over the raw die will be formed with significant "dimples" from the input pins, [30, 39, 45]. A sheet metal blank is then hand mounted in the machine and restrained by two gripping jaws, one on each side of the die. Each jaw has two setscrews which need to be hand tightened. To ensure proper alignment of the blank part, small stops are placed within the jaws. However, care must be taken to evenly progress forward and tighten the two setscrews as it is possible to fully engage one side of the jaw, while preventing the engagement of the other side. Engagement is prevented by excessive misalignment. If this happens, the sheet metal is allowed to slip out of the jaw in the

unengaged section and the workpiece is not subjected to the prescribed strain profile. As a result, a "correct" die will not form a "correct" part.

Once the part is mounted, an additional parameter needs to be specified, the stretch distance. This is the distance that the machine moves through such that the metal blank is tangent to the die when forming is finished. This distance is a function of the machine geometry as well as the die shape. A change in the way that this distance is calculated is presented in Appendix D. Once the part is stretch formed, a locating mark is made in the sheet metal and the part is removed from the plant for measurement.

A coordinate measurement machine (CMM) is the last part of our control loop. Care must be taken to properly mount the part in the CMM and to maintain its correct orientation and alignment. The measurement platform and a formed part are shown in Figure 7-2. Two alignment posts are used to give the formed part the same orientation that it had in the forming machine's clamping jaws. The part is supported at four points. Two of these are vacuum cups used to hold the part down and two are support posts placed across the "long side" of the part. Because this is not a kinematic mount, care should be taken with the non-vacuum support posts as it is possible to artificially deform the part and thus get a bad measurement.

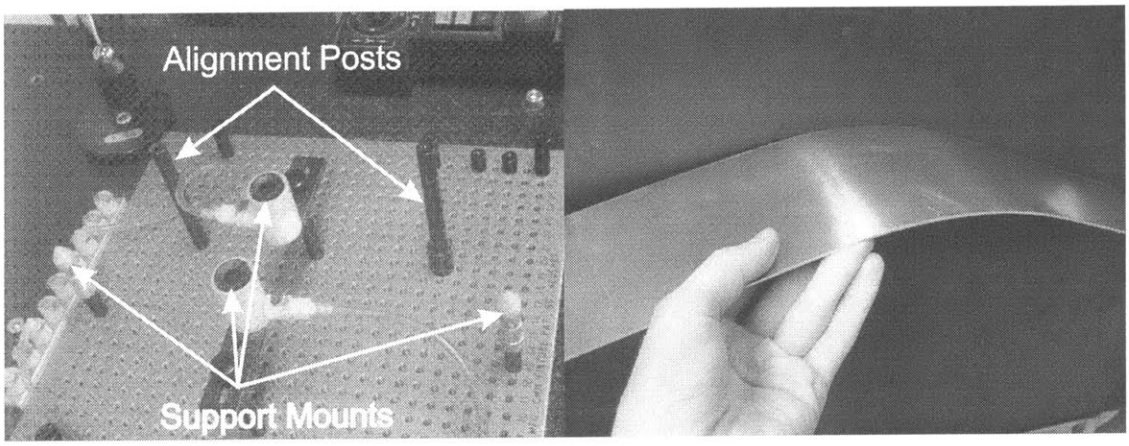


Figure 7-2 Part support for use in coordinate measurement machine (left) and formed part (right).

Once the part is measured, we have to "register" it. The registration process is an attempt to locate the part in 3D space and is necessary because we move the part between different machines, therefore removing natural datum points which are known in any one coordinate frame. At the forming step we introduced an artificial locating mark; we now register the part by subtracting the part height at this mark from all the measured points. Once the part is registered, it is subtracted from the reference vector to form the error signal and close the feedback loop. Additional details on the registration procedure, as well as the whole experimental procedure, may be found in [30, 39, 45]. The approximate time breakdown per cycle is as follows.

1. Automatic discrete-die assembly from specification, 50 minutes
2. Part mounting in forming machine, 5 minutes
3. Part forming, 5 minutes
4. Part removal from forming machine, 3 minutes
5. Part mounting in measurement jig, 5 minutes
6. CMM measurement time, 30 minutes
7. Post CMM data processing and next-die specification, 15 minutes

7.2 Robust Pole Placement

Before we take a detailed look at each of the three controller design methods, we take a final look at pole placement methods. Our initial method of designing the controller was, in effect, pole placement through matrix inversion. As we saw, this guaranteed one

time-step settling and a unique solution for the gain matrix. One time-step settling meant that all the poles had to be located at the origin of the z-plane. In Chapter 5 however, we saw that the sensitivity of the matrix inverse to noise and numerical round-off prevented us from obtaining good results. A cursory look into traditional pole placement techniques reveals that they are not applicable to CtC MIMO processes since they require the inversion of the Gaussian matrix (as it may be contained in the Ackermann's formula or in the controllability matrix) [32, 36]. As a solution to this, a method for "robust" pole placement is described by Kautsky *et al.*, [37].

Kautsky *et al.*, [37], exploit the underdetermined nature of the pole-placement problem to select the best condition numbers out of a pool of solutions. Recall that an *nxn* system will have n poles, but requires the determination of n^2 entries to fill the controller matrix completely. A user specifies the desired poles (eigenvalues) and the algorithm chooses the best conditioned corresponding eigenvector set. "The assigned poles are then as insensitive to perturbations as possible and the resulting feedback matrix is as reasonably bounded as may be expected, given the original system" [37].

The Kautsky *et al.*, [37], method may be applied through the *MATLAB* function "place." The test process is a $10x10$ array of inputs, thus 100 inputs and 100 corresponding outputs are assumed. Since we have the ability to place poles arbitrarily, let us place all poles at the origin which will give one time step settling. The obtained poles and calculated gains are shown in Figure 7-3. The displayed gains correspond to the 45^{th} column of the gain matrix. The 45^{th} input occurs near the center of the input bundle; the inputs are numbered row-wise such that the entries in the first row correspond to numbers 1 through n and the first entry in the second row is number n+1. Immediately,

one notices that the control gains are too high for practical control to be implemented. To put it in perspective, the controller would make a 20 cm correction to a 1 mm error.

A second case, one in which we are not asking for matrix inversion, is to place all the poles within a certain band. The size of this band will therefore describe the performance of the closed-loop process according to Equation 2-21. The results of placing all the poles within a band of ± 0.3 are shown in Figure 7-4. Again, the control gains are too high for practical implementation. These results lead us to abandon direct pole placement methods.

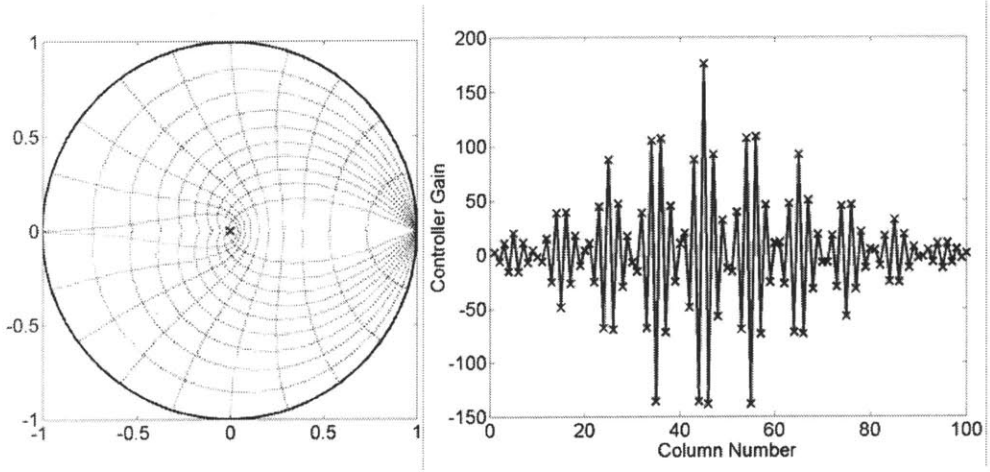


Figure 7-3 Obtained pole locations and control gains calculated via Kautsky *et al.* method. A 10*x*10 bundle of inputs (totaling 100) is assumed and the 45th input (occurring near the center of the bundle) is displayed. All poles are located at the origin.

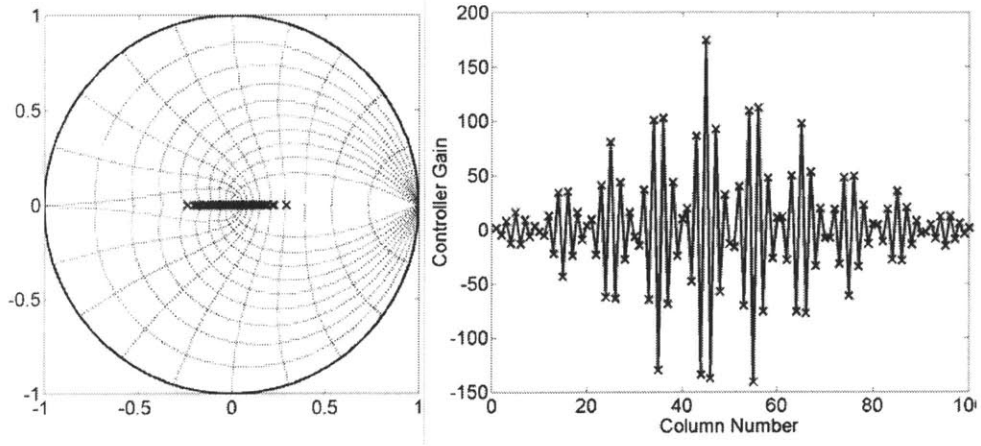


Figure 7-4 Obtained pole locations and control gains calculated via Kautsky *et al.* method. A 10*x*10 bundle of inputs (totaling 100) is assumed and the 45th input (occurring near the center of the bundle) is displayed. All poles are located within a ± 0.3 band on the z-plane.

7.3 LQR Controller

A possible solution to designing the controller is the linear quadratic regulator (LQR). The controller design is done in state-space as discussed in Chapter 3. The LQR design provides an optimal control gain with a control law of the form

$$\underline{u} = K_c(-\underline{x})$$

Equation 7-4

where K_c is a gain matrix. Note that the control law assumes full knowledge of all states. The optimal solution is the result of an optimization problem to minimize the quadratic cost function, [33]:

$$J = \frac{1}{2}\sum_{k=0}^{N}\left[\underline{x}^T(k)Q\underline{x}(k) + \underline{u}^T(k)R\underline{u}(k)\right]$$

Equation 7-5

subject to the original state-space equation:

$$0 = -\underline{x}(k+1) + A\underline{x}(k) + B\underline{u}(k)$$

Equation 7-6

where Q and R are user-defined weighting matrices with $Q=Q^T$ and $R=R^T$. The controller gain matrix is given by

$$K_c = R^{-1} B^T P_c$$

Equation 7-7

where P_c satisfies the algebraic Riccati equation

$$A^T P_c + P_c A - P_c B R^{-1} B^T P_c + Q = 0$$

Equation 7-8

and

$$P_c = P_c^T \geq 0$$

Equation 7-9

Although Equation 7-8 often has many solutions, only one of them satisfies Equation 7-9, i.e. is positive semi-definite, [33].

In addition to the optimality of the solution, Zhou and Doyle, [32], show that the LQR design guarantees a minimum of 60 degrees of phase margin and 6 dB of gain margin to an additive disturbance when $R=\alpha I$, where α is a scalar. It should be noted, however, that there is no guarantee on the sensitivity of the solution. Thus, the gain margins may not be sufficient if the solution has a high sensitivity to model errors.

Before we apply this method to the cycle-to-cycle plant, we stress the fact that the controller is simply a gain and is, thus, equivalent to the proportional controller analyzed in Section 3.1.1. To overcome this limitation, and achieve our goal of fast response and guaranteed steady-state error, we embed the desired controller dynamics into the description of the plant and then use the LQR method to design the controller gain matrix. The dynamics augmented plant is therefore

$$G_P = \left(\frac{z}{z-1}\right)\left(\frac{1}{z} K_P\right) = \frac{1}{z-1} K_P$$

Equation 7-10

which is then expressed in state-space as

$$z\underline{x} = I\underline{x} + K_P\underline{u}$$
$$\underline{y} = I\underline{x} + [0]\underline{u}$$

Equation 7-11

This plant description is both controllable and observable with our GIC model and is used in subsequent LQR designs.

The weighting matrices, Q and R, provide a different approach to controller design. The designer chooses the relative weights of inputs and outputs, perhaps based on cost of control, rather than performing specific pole-placement. The drawback of this approach seems to be a lack of ability to set the performance, e.g. settling time, of the closed-loop process. This often leads to a bit of "trial-and-error" with the weighting matrices in order to obtain the desired performance.

In choosing the weighing matrices, it is assumed that all outputs are equally important and that the same holds true for all inputs. Therefore, the matrices will assume the following form:

$$Q = mI \qquad R = I$$

Equation 7-12

where m is a relative weighting scalar. Note that this form complies with that required for guaranteed phase and gain margins. It should be noted and understood that choosing such forms for the weighting matrices limits the available solution space; the multivariable problem has been reduced to a single variable one, that of choosing m.

7.3.1 LQR Simulation

Looking into the future, our experimental testbed is the sheet metal forming process described in Section 1.5.1. Thus, we will use a similarly sized, 110 input-110 output process for simulation. As in the physical process, the inputs are bundled into 10 rows by

11 columns. The process model is assumed to have a generic, unity coupling spread. However, the scaling matrix is determined according to real, experimental data obtained from Vaughan, [39]. The target shape is a 27 cm radius cylinder. Noise, in noisy simulations, will be modeled as normal with an assumed standard deviation of 0.001 cm.

The discrete-time LQR controller may be designed with the *MATLAB* function "dlqr." The results of designing an integral controller gain matrix with m=100 are shown in Figure 7-5. Both a noiseless simulation (left) and one with noise (right) are shown. Note that an error is introduced at run 0 and the first closed-loop part is the second part, run number 1. The initial part, run 0, is an experimentally formed part obtained from Vaughan, [39], all subsequent parts are simulated in *MATLAB*. The registration process is included and is modeled by taking the average of two points occurring near the center and subtracting it from all part heights. The displayed mean error points are the mean of the 110 errors which make up each run's target-part difference. Noise takes the form of an additive disturbance on each of the 110 outputs. We can readily see that the process has a quick response and maintains the ability to fully reject a step disturbance, as is expected with an integral controller. Therefore, the process distribution will be centered on target. We also consider the spread of the output distribution. A simulation of 1,000 closed-loop runs reveals that the ratio of closed-loop variance to open-loop variance is 1.9. For further analysis, we will draw an analogy to a SISO process. Since we are using uncorrelated noise, we can use Equation 2-38 to estimate the loop gain as 0.95. Note that this equation is exact only for SISO processes. We can use this estimate of loop gain along with Equation 2-21 to calculate the expected settling time as 2 cycles. As with

SISO, this is an estimate for MIMO processes. As we can see, our estimate of loop gain is consistent with the observed performance.

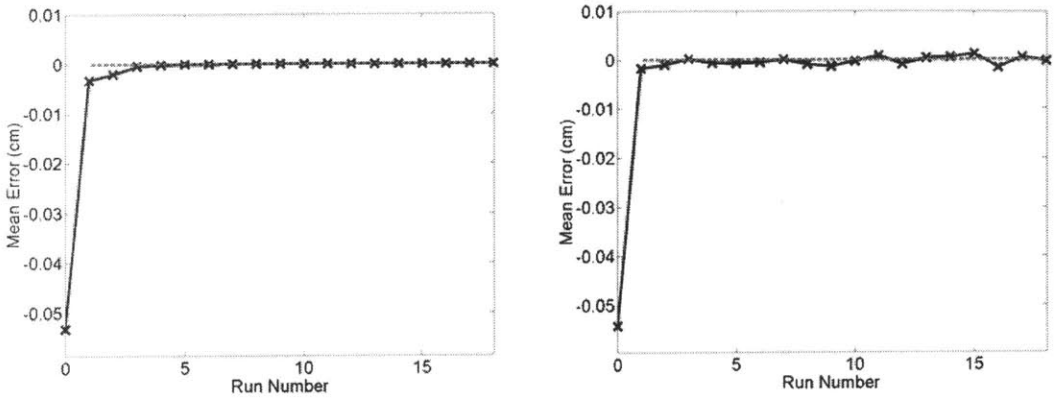


Figure 7-5 Simulated closed-loop runs of a 110 input, 110 output process with LQR designed integral controller gain matrix. Noiseless simulation (left) and Noisy simulation (right). m=100.

In addition to time response, we look at the robustness properties of this controller-plant combination. As in Example 6-6, we estimate stability robustness based on allowable variations in the scaling matrix values, i.e. we assume that a whole row varies together. Again, we bound our nominal uncertainty at 10% of the scaling value. Therefore, the "true" process gain matrix is described according to

$$K_P = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_{n^2} \end{bmatrix} \left([I] + 0.1 \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \delta_{n^2} \end{bmatrix} \right) \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

Equation 7-13

with $\|\delta_i\|_\infty < 1$, just as it was in Example 6-6. The results of estimating the μ bounds, γ_{upper} and γ_{lower}, are shown in Figure 7-6. Note, that since we are in pure discrete time, we only need to compute the bounds on the interval [0,π]. The maximum γ_{upper} is 0.14, and through Equation 6-25 this translates to an allowable infinity norm of δ_i of 7.1. Therefore,

since we have a 0.1 weighting value that we assumed for uncertainty, each scaling value may vary by +-71% without upsetting the stability of the closed-loop process.

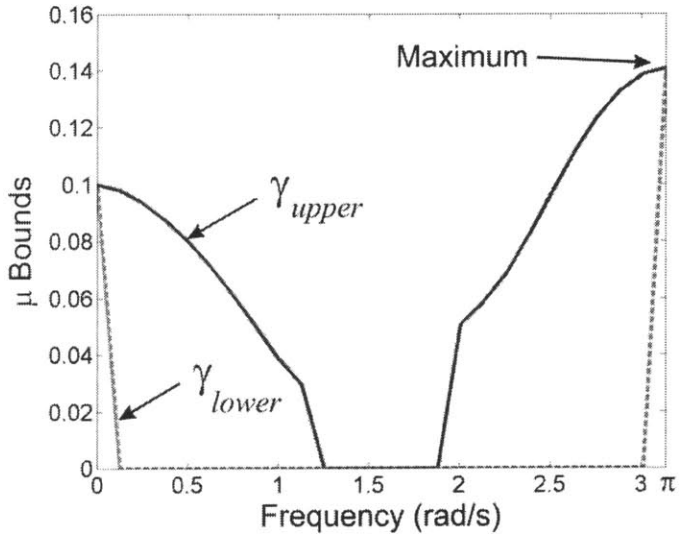


Figure 7-6 μ_Δ bounds as a function of frequency for a closed loop process with an LQR designed integral controller gain matrix. m=100.

Although we have obtained good performance from our initial weight of m=100, we want to see if we can get better performance by weighting the states even more. As a second design, we use m=1,000. All other simulation parameters remain unchanged. A simulation of the closed-loop runs is shown in Figure 7-7. As we can see, we get marginally better performance in terms of speed of response while maintaining the ability to reject step disturbances. The closed-loop to open-loop variance ratio is 1.8 from 1,000 simulated cycles. Again, we estimate the loop gain using Equation 2-38. The estimate is 0.89. Now, we use Equation 2-21 to estimate the expected settling time as 2 cycles. This is very close to the results seen in Figure 7-7.

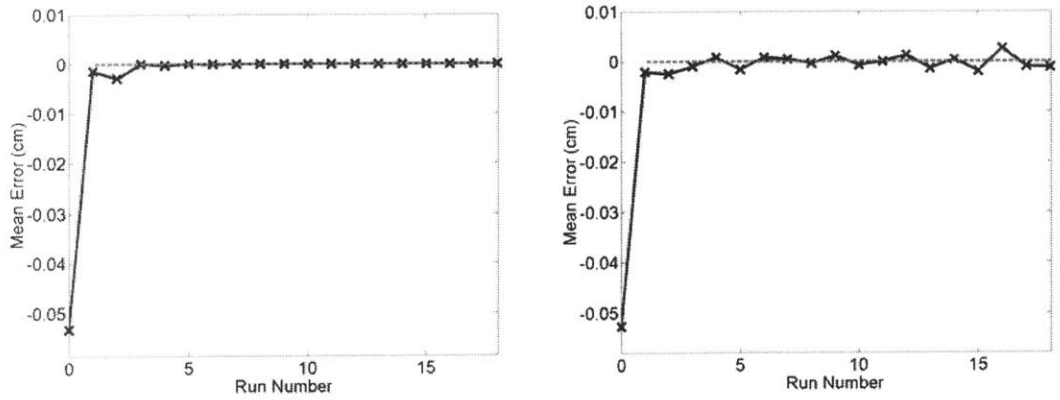


Figure 7-7 Simulated closed-loop runs of a 110 input, 110 output process with LQR designed integral controller gain matrix. Noiseless simulation (left) and Noisy simulation (right). m=1,000.

To see the cost of obtaining marginally better performance when using the higher value of m, we look at the structured singular values. Again, we assume that uncertainty is sufficiently described as a variation of the non-zero values of the scaling matrix. Estimates of the upper and lower bounds are shown in Figure 7-8. The maximum value of γ_{upper} is approximately 0.22, and through Equation 6-25 translates to an allowable infinity norm of δ_i of 4.5. With the higher weighting value, the allowable variation in the scaling factors has gone from 71% down to 45%.

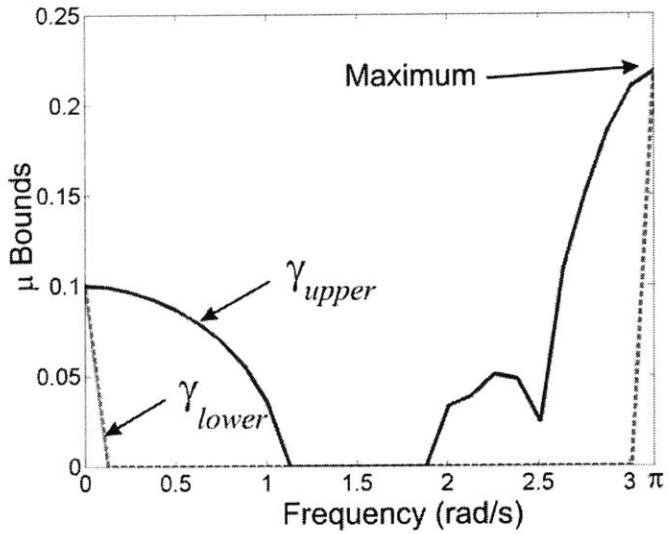


Figure 7-8 μ_Δ bounds as a function of frequency for a closed loop process with an LQR designed integral controller gain matrix. m=1,000.

7.3.2 LQR Experimental Validation

Experimental validation of the LQR designed controllers is carried out on the sheet metal forming process described in Section 1.5.1. The experimental procedure was previously outlined by Webb and Hardt, [38], Vaughan, [39], and was briefly described in this thesis in Section 7.1. According to this procedure, the first two runs are open loop and are used to calibrate the plant model. Then, the integral controller is used for the rest of the cycles. Note that we have made a small change in the experimental procedure, as described in Appendix D. As in simulation, a reference input of a 27cm radius cylinder is specified. An example part is shown in Figure 7-9.

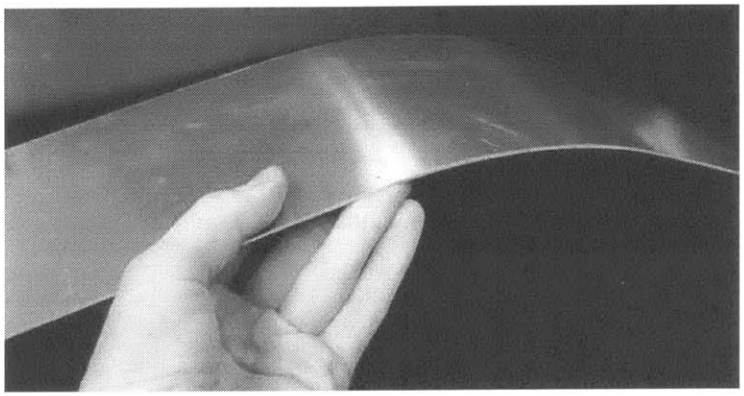


Figure 7-9 Experimentally obtained section of a 27cm radius cylinder.

Figure 7-10 shows the results of using the integral controller gain matrix designed with m=100. The data for the first two, open loop, runs was obtained from Vaughan, [39]. All experimental data is listed in Appendix E. As in simulation, each point is the spatial mean of the error between the 110 reference and part points for each cycle. The figure experimentally confirms that the MIMO closed-loop process will reject a step disturbance. The final three runs, those that appear to be settled, are used to compute performance statistics. The average of the mean errors is calculated using the t distribution, [4]. This value is 0.0007 cm with 95% confidence bounds of -0.004 cm and

0.003 cm; the mean is essentially zero. Next, we calculate the variance of the mean errors using the χ^2 distribution, [4]. The nominal variance is 2.1×10^{-6} cm^2 with 95% confidence bounds of 5.8×10^{-7} cm^2 and 8.4×10^{-5} cm^2. This closed-loop variance is used with a set of eight open-loop runs performed by Vaughan, [39], to establish a closed-loop/open-loop variance ratio. This ratio is 0.4 with lower and upper bounds of 0.06 and 2.2 at 95% confidence, [2, 39]. The F statistic was used to compute this ratio, [4]. Note that we expect a ratio of 1.9 from our simulated runs. Even though we cannot assess the exact variance ratio, we conclude that LQR cycle-to-cycle control still possesses the desired properties of process centering and bounded variance amplification.

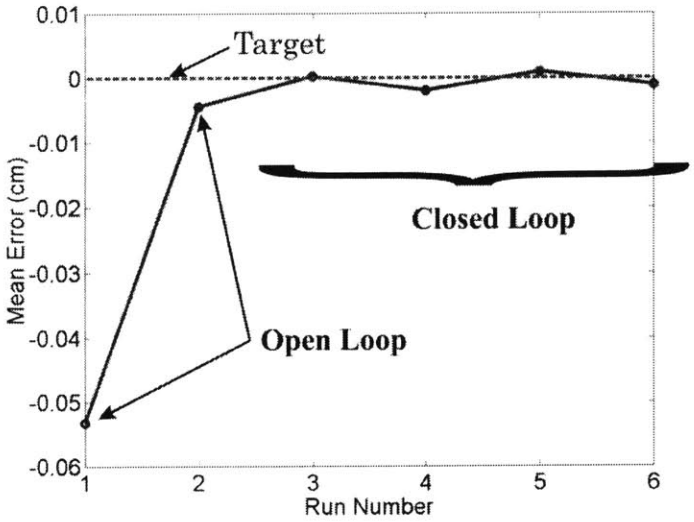


Figure 7-10 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. Mean error shown, LQR designed integral controller gain matrix, m=100.

Since we are examining a MIMO process, it is possible for one region of positive errors to mask another region of negative errors, resulting in a part which appears good "on average." To combat this, we display both the maximum and root-mean-squared (RMS) errors in Figure 7-11. The maximum error is computed based on absolute magnitude with the sign reapplied post-computation. Note that the maximum error does

not necessarily appear at the same output on subsequent runs. We see that these plots are consistent with Figure 7-10 as both the maximum and RMS errors appear to settle by run number four. Note that we do not expect the maximum and RMS errors to go to zero, even with infinite time. This is entirely consistent with our assumption that any one output will go to zero *on average*, but will not necessarily be zero at any one step. Therefore, since both the maximum and RMS error are looking across the full 110 outputs and any one of these outputs can be nonzero at any time step, the maximum and RMS error will not be zero. For the "settled" criterion, we only look for a random distribution (maximum error) or a leveling out of noise (RMS).

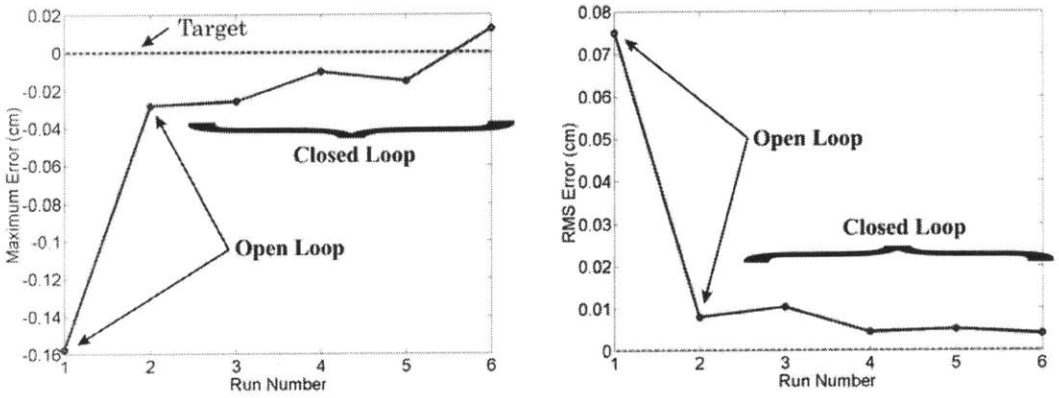


Figure 7-11 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. Maximum (based on magnitude) and RMS error shown, LQR designed integral controller gain matrix, m=100.

Another set of experiments is run to test the controller gain matrix designed in Section 7.3.1, with m=1,000. Remember that this controller slightly improved the time response at the cost of a smaller stability margin. The results of experimental verification are shown in Figure 7-12 and Figure 7-13. Note that we again use the first two calibrating runs from Vaughan, [39]. We can clearly see that this controller goes unstable, since the outputs are deviating away from the target input. Experiments had to be halted after the 6^{th} cycle since the 7^{th} die was not realizable. Because we have no way of knowing the

exact reason for this instability, we are forced to theorize as to its cause. The cycle-to-cycle process is made up of simple components: the controller, an output additive disturbance, and a pure delay-gain plant, Figure 6-7. Since we design the controller, it is known exactly and, thus, is not the cause of the instability. We can eliminate the additive disturbance, as additive signals injected anywhere within the loop cannot cause instability. The plant or, more precisely, uncertainty in the plant gain matrix must therefore be the cause of the instability. We are forced to conclude that the error between our plant model and the physical process exceeds the stability bound computed through μ analysis.

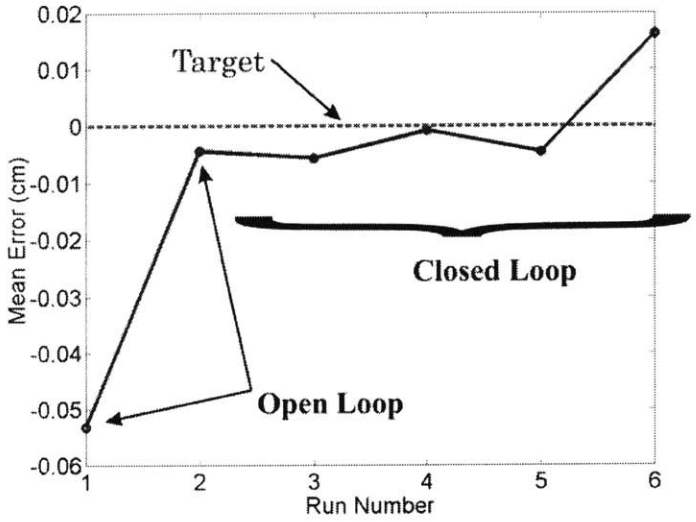


Figure 7-12 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. LQR designed integral controller gain matrix, m=1,000.

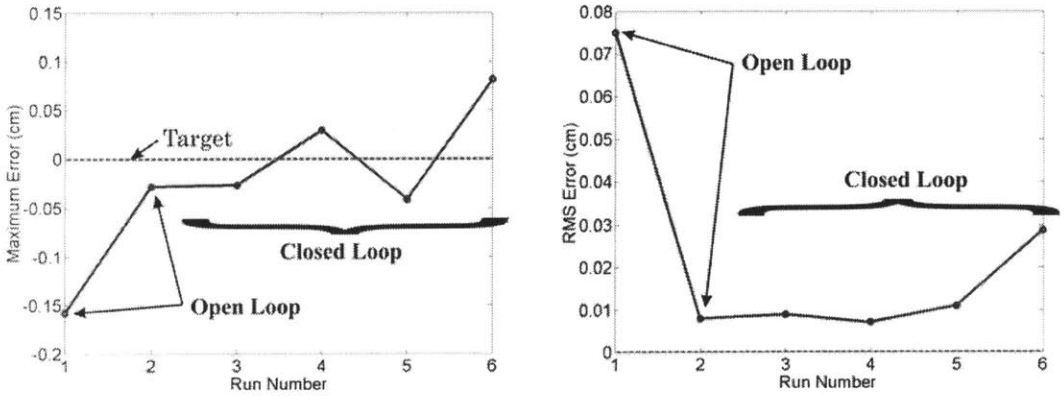


Figure 7-13 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. Maximum (based on magnitude) and RMS error shown, LQR designed integral controller gain matrix, m=1000.

7.4 LQG – H_2 Controller

Our next approach to designing the controller is the linear quadratic Gaussian (LQG) or H_2 controller. This controller is actually the full state feedback LQR controller coupled with a Kalman optimal observer. This controller is especially useful in environments where accurate knowledge of the states is not available because of excessive measurement noise. Note that although measurement noise will not affect the part being measured, it will affect subsequent parts through the use of feedback.

The general LQG problem is posed as follows. Assume a plant in state-space such that

$$z\underline{x} = A\underline{x} + B\underline{u} + \Gamma\underline{w}$$
$$\underline{y} = C\underline{x} + \underline{v}$$

Equation 7-14

where $\underline{w}$ and $\underline{v}$ are white noise terms; zero-mean Gaussian processes with no correlation in time. The covariance of the noise terms is

$$E\left\{\underline{w}\underline{w}^T\right\} = W \geq 0 \qquad E\left\{\underline{v}\underline{v}^T\right\} = V > 0$$

Equation 7-15

and we assume that the cross covariance $E\{\underline{w}\underline{v}\}=0$. The problem, then, is to minimize the cost function:

$$J = E\left\{\frac{1}{2}\sum_{k=0}^{N}\left[\underline{x}^T(k)Q\underline{x}(k)+\underline{u}^T(k)R\underline{u}(k)\right]\right\}$$

Equation 7-16

This cost function is very similar to that used for LQR, Equation 7-5. However, in LQR, we assumed full knowledge of the states. The solution to the LQG problem is through the separation principle, [33]. First, we design an optimal observer such that we minimize

$$E\left\{(\underline{x}-\underline{\hat{x}})^T(\underline{x}-\underline{\hat{x}})\right\}$$

Equation 7-17

Then, we use our estimate as if it were an exact measurement to solve the LQR state feedback problem.

To obtain the optimal observer gain, we solve a problem that is the dual to the LQR optimal controller gain problem. The optimal Kalman filter gain matrix is

$$K_f = P_f C^T V^{-1}$$

Equation 7-18

where P_f satisfies

$$P_f A^T + AP_f - P_f C^T V^{-1} C P_f + \Gamma W \Gamma^T = 0$$

Equation 7-19

and $P_f = P_f^T \geq 0$. As in Equation 7-8, Equation 7-19 often has many solutions, only one of which is positive semi-definite, [33]. The equations of the Kalman filter are, [33, 40],

$$\hat{x}_{k+1|k} = (A - K_f C)\hat{x}_{k|k-1} + Bu_k + K_f y_k$$
$$\begin{bmatrix}\hat{y}_{k|k}\\ \hat{x}_{k|k}\end{bmatrix} = \begin{bmatrix}C(I-MC)\\ I-MC\end{bmatrix}\hat{x}_{k|k-1} + \begin{bmatrix}CM\\ M\end{bmatrix}y_k$$

Equation 7-20

where M is called the innovations gain and it obtained as $M = K_f C^T$. As we can see, the Kalman filter gives the most recent estimate of both the states and the process outputs, given the current noisy output y_k. Note that the control signal, u_k, is used to update the observer states. The final control scheme, therefore, is the combination of Equation 7-20

and Equation 7-4. Unlike the LQR controller, the LQG controller does not have any guaranteed stability margins, [32]. Thus, we must use our singular value methods to estimate the bounds for every design.

The H_2 problem may also be alternately expressed as a robust control problem. A standard representation of a feedback problem with uncertainty is shown in Figure 7-14. Here, as before, y are the measured output variables and u are the control signals. However, we introduce a generalized error, z, and a generalized disturbance, w. These quantities may be used to represent signals that the designer wants to keep small and input quantities which may disturb the closed-loop process, respectively. P(z) is an augmented description of the plant that accounts for all the inputs and outputs. The augmented plant thus takes the form

$$\begin{bmatrix} zx \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

Equation 7-21

The H_2 problem may thus be posed as

$$\min\left[\frac{1}{2\pi}\int_{-\infty}^{\infty}\left|T_{wz}(j\omega)\right|^2 d\omega\right]^{1/2} = \min\left\|T_{wz}(j\omega)\right\|_2$$

Equation 7-22

which means that we are minimizing the H_2 norm of the closed-loop transfer function matrix from w to z. The cost function, Equation 7-16, may be equivalently expressed as, [41],

$$J = E\left\{\frac{1}{2}\sum_{k=0}^{N}\begin{bmatrix}\underline{x}^T(k) & \underline{u}^T(k)\end{bmatrix}\begin{bmatrix}Q & N_c\\ N_c & R\end{bmatrix}\begin{bmatrix}\underline{x}(k)\\ \underline{u}(k)\end{bmatrix}\right\}$$
$$= E\left\{\frac{1}{2}\sum_{k=0}^{N}\begin{bmatrix}\underline{x}^T(k) & \underline{u}^T(k)\end{bmatrix}\begin{bmatrix}C_1^T\\ D_{12}^T\end{bmatrix}\begin{bmatrix}C_1 & D_{12}\end{bmatrix}\begin{bmatrix}\underline{x}(k)\\ \underline{u}(k)\end{bmatrix}\right\}$$

Equation 7-23

where in Equation 7-16 we assumed that the cost N_C=0.

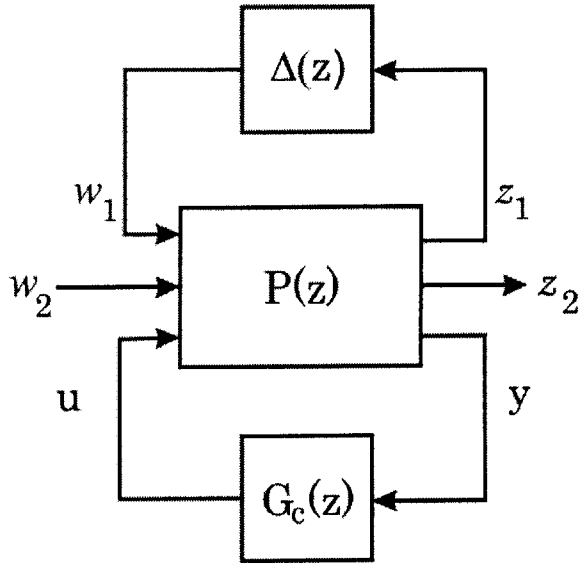


Figure 7-14 Representation of uncertainty in feedback control, [33].

H_2 controller synthesis requires, [32],

1. (A,B_2) is stabilizable and (C_2,A) is detectable
2. $D^*_{12}D_{12}>0$ and $D_{21}D^*_{21}>0$
3. $\begin{bmatrix}A-j\omega I & B_2\\ C_1 & D_{12}\end{bmatrix}$ has full column rank for all ω
4. $\begin{bmatrix}A-j\omega I & B_1\\ C_2 & D_{21}\end{bmatrix}$ has full row rank for all ω

where the A, B, C, and D matrices were defined in Equation 7-21.

7.4.1 LQG Simulation

To obtain a design, we will take advantage of the separation principle and design the LQG controller in two steps, the Kalman filter and the LQR regulator. The associated *MATLAB* functions are "dkalman" and "dlqr." For our design, we will be using the experimentally calibrated plant model from Section 7.3.1. As in the LQR designs, we embed the desired integrator dynamics into the description of the plant. For the Kalman filter, we assume that the state disturbance signal, $\underline{w}$, enters the augmented plant at the same spot as the control input and that the measurement noise affects the augmented plant output. This is shown in Figure 7-15. To obtain the covariance matrices, we will model $\underline{w}$ and $\underline{v}$ as

$$\underline{w} = k_{w1} a \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} + k_{w2} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \qquad \underline{v} = k_v \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

Equation 7-24

where a_i are NIDI signals with zero mean and unity standard deviation. We use the k_{w1} term to model random variations in fixturing the part within the gripping jaws of the forming press. To obtain the scaling factors, we look back at Pi, [42], and Vaughan, [39]. Pi lists the standard deviation of the measurement process (fixturing, measuring, and registration) as 0.0038 cm, and the total standard deviation for the whole forming process as 0.0076 cm. This means that there is an even uncertainty contribution from both the forming and the measurement process. Vaughan quantifies the forming process standard deviation as 0.0025 cm; about 2/3 of Pi's level. Because we will are using Vaughan's forming procedure, we will use his number while assuming an equal contribution from the measurement process, which was observed by Pi (Vaughan did not experimentally quantify the measurement process noise). Under these guidelines, the measurement noise

scaling factor, k_v, should be 0.0025 cm. Next, we will assume that the forming noise is dominated by things other than part fixturing, and use a 20%-80% split for the weighting factors k_{w1} and k_{w2}. This means that these factors are k_{w1}=0.0005 cm and k_{w2}=0.002 cm. These scaling factors are repeated in Table 7-2. The LQR regulator is then designed as in Section 7.3.1 with m=4. Higher values of m were found to result in an oscillatory response when coupled with the Kalman observer.

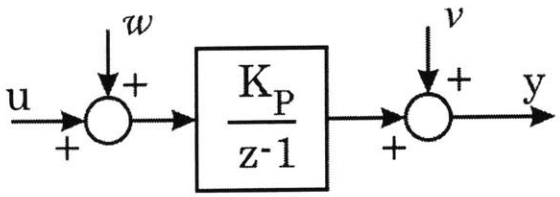


Figure 7-15 Plant description used for Kalman filter design.

k_{w1}	k_{w2}	k_v
0.0005	0.002	0.0025

Table 7-2 Kalman filter design standard deviation scaling factors.

For an initial simulation, we will assume that all states are zero and an error is introduced at run number 0. The first closed-loop part is run number 1 and all parts are simulated in *MATLAB*. For this simulation, we do not include the effect of the registration procedure. The results obtained with the H_2 controller are shown in Figure 7-16. Both a noisy and a noiseless simulation are shown. A generic Gaussian noise with a standard deviation of 0.001cm is used. We see that the process settles quickly and that the integral action needed to give zero steady-state error is retained with this observer-regulator pair. 1,000 simulation cycles are used to compute a closed-loop to open-loop ratio of 1.6. Again, we draw an analog with a SISO process and use Equation 2-38 to compute the equivalent loop gain. The loop equivalent SISO loop gain is 0.75. The expected settling time for this loop gain is 3 cycles. This is consistent with Figure 7-16.

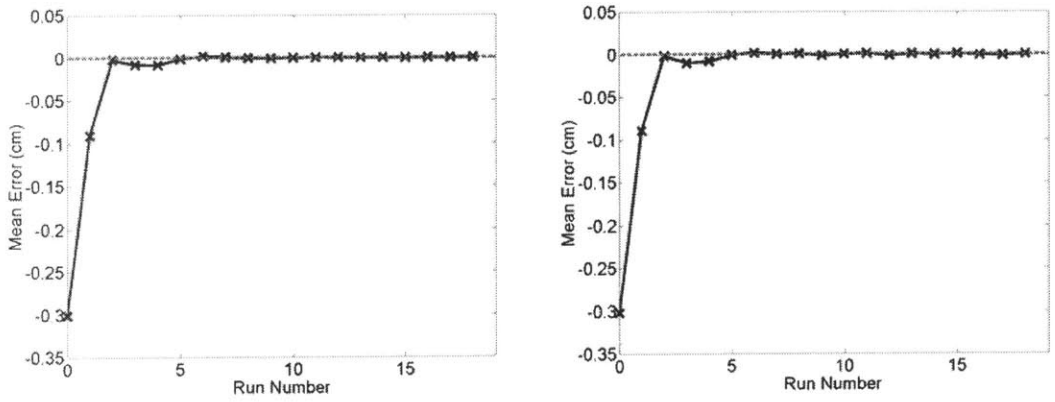


Figure 7-16 Simulated closed-loop runs of a 110 input, 110 output process with H_2 designed controller. Noiseless simulation (left) and Noisy simulation (right).

To analyze stability robustness, we again assume a 10% uncertainty in the entries of the plant scaling matrix, Equation 7-13. The results of computing the structured singular values are shown in Figure 7-17. The maximum γ_{upper} is 0.1, meaning that the allowable infinity norm of δ_i is 10, see Equation 6-25. This translates to an allowable variation of ±100% in each of the scaling values.

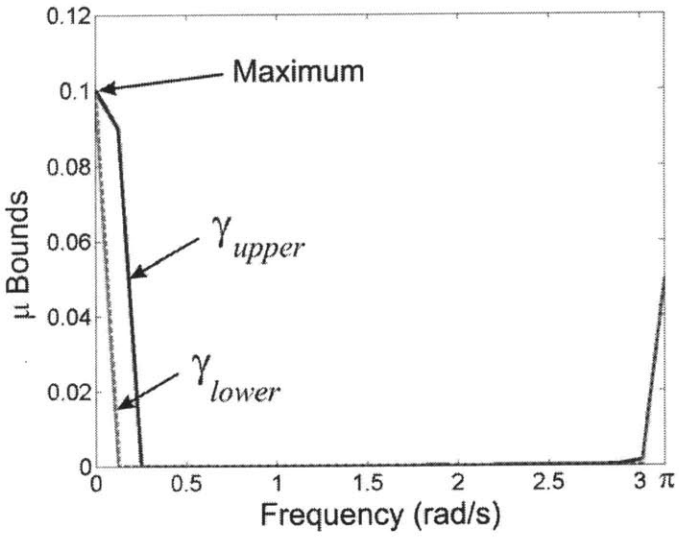


Figure 7-17 μ_Δ bounds as a function of frequency for a closed loop process with an H_2 designed controller.

Before we begin our experiments, we look at the problem of initializing the controller states. To create Figure 7-16, we initialized all plant and controller states to zero before introducing an error at run number zero. Although this is appropriate for a simulation, our experimental process will already have formed two parts, for model calibration, and thus will be closer to the desired output. We can take advantage of this knowledge to initialize the observer states. Note that the observer operates on inputs y and $(u_i\text{-}u_{i\text{-}1})$, Equation 7-20, and that the LQR part of the controller contains no states of its own. We therefore slightly change the simulation procedure and the run-numbering format by admitting the first *two* open-loop parts. The observer will be initialized as all zero before making the first part, i<-1. The first observer states are initialized based on the first outputs $y_{\text{-}1}$ and the whole first input $u_{\text{-}1}$. The second and subsequent states are then computed based on the i^{th} output y_i and the change in input $u_i\text{-}u_{i\text{-}1}$. Note that, with this procedure change, the first controller-made part is actually the third part, still named run number 1.

For comparison with experiments, we will use the first two open loop runs from Vaughan, [39]. However, we are still not including the registration procedure. The results of this approach to initializing are shown in Figure 7-18 where we have included both the Kalman observer estimate of the output and the actual simulated output. Comparison of Figure 7-18 with Figure 7-16 shows that the initial two runs can be used to our advantage to reduce the error when the first closed-loop part is made at run number one.

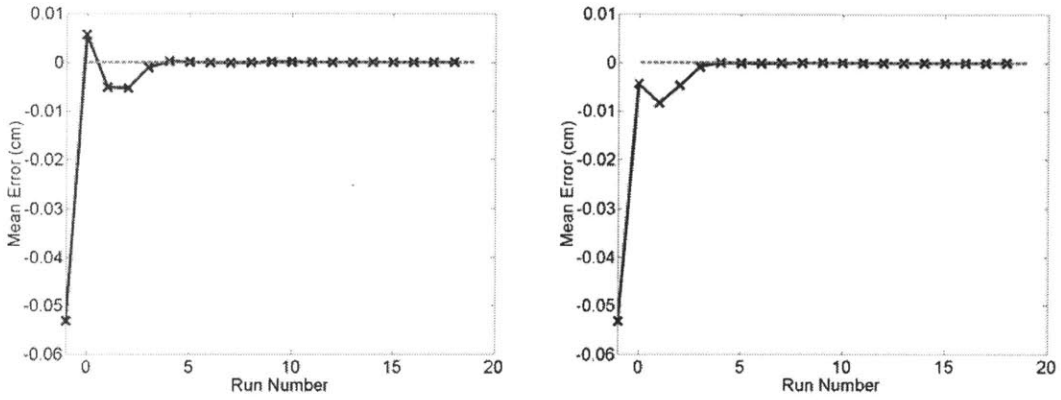


Figure 7-18 H_2 simulation, Kalman observer estimated output (left) and actual time response (right) with non-zero initialized observer states.

Finally, we look at the application-specific step of registering the part. During the physical experiments, our part will be made on a stretch forming machine, removed, and measured on a coordinate measuring machine (CMM). Since the part travels between different machines, there exists no natural datum point that is considered "known" in absolute space. Additional details on this may be found in [30]. As a remedy, an artificial datum point is assumed and is taken as the average of the two center points (in a $10x11$ bundle). This average is always considered to be at absolute height 0 and its height measurement is thus removed from all point measurements. A part with the datum height removed from all height measurements is called "registered." In the preceding simulations, we have not registered the simulated outputs. The results of including this step in simulation are shown in Figure 7-19. Note that we have included both the Kalman observer estimate of the output (left) and the actual output (right). As we can see, the act of registering, which is required in the experimental process, significantly changes the time response. This change in the output was not observed with the pure LQR procedure. It is thus believed that the observer is to blame for this difference in behavior; our active

offsetting of the measured output interferes with the Kalman observer estimation of the current output and upsets the controller.

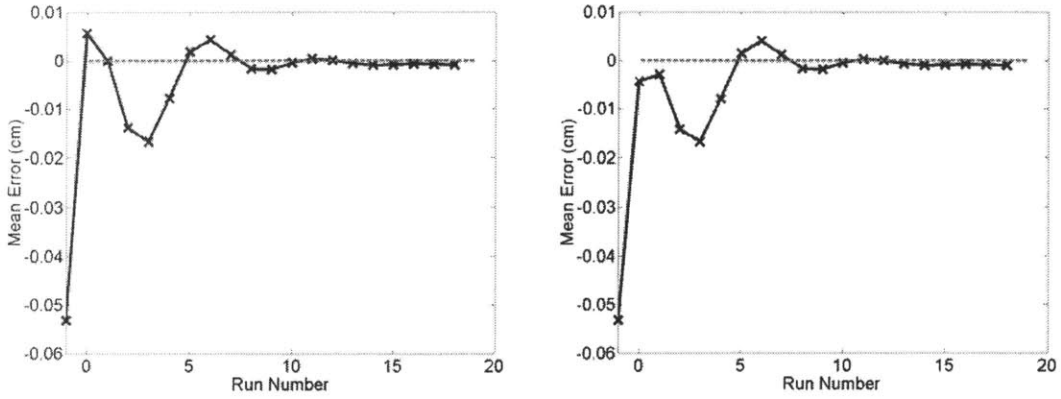


Figure 7-19 H_2 simulation with registering. Kalman estimate of the output (left) and actual simulated output (right). A noiseless simulation is used.

7.4.2 LQG Experimental Validation

As before, we experimentally validate our controller design on the discrete-die sheet metal forming process. The target value is still a 27 cm radius cylinder, which allows us to use the first two, open-loop runs from the LQR experiments. The observer states are initialized based on these open-loop runs.

The results of the experiments are shown in Figure 7-20 and Figure 7-21. As we can see, the observed response is very close to the expected response, Figure 7-19. We prominently see the oscillation behavior resulting from the registration of the experimental workpieces. Because of this long lasting oscillation, we use only the last four runs of Figure 7-20 to estimate process statistics. Using the *t* statistic, the mean value is computed to be -0.001 cm with 95% confidence bounds of -0.003 cm and 0.0006 cm. Next, the χ^2 distribution is used to compute the variance of the process. The nominal variance is $1.5\text{x}10^{-6}$ cm^2 with 95% confidence bounds of $1.8\text{x}10^{-7}$ cm^2 and $2.1\text{x}10^{-5}$ cm^2. This closed-loop variance is used with a set of eight open-loop runs performed by

Vaughan, [39], to establish a closed-loop/open-loop variance ratio. The computed ratio of variance is 0.3 with bounds of 0.02 and 1.5 at 95% confidence. The variance ratio is calculated using the F statistic. The simulation obtained variance ratio of 1.6 appears within the bounds at 96% confidence. Overall, the H_2/LQG controller behaves as predicted, with the designed performance significantly degraded by the process-specific, necessary output registration procedure.

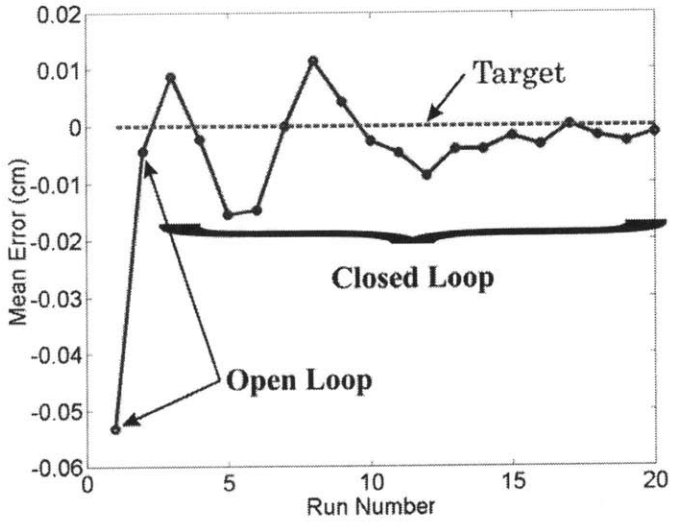


Figure 7-20 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. H_2 (LQG) designed controller.

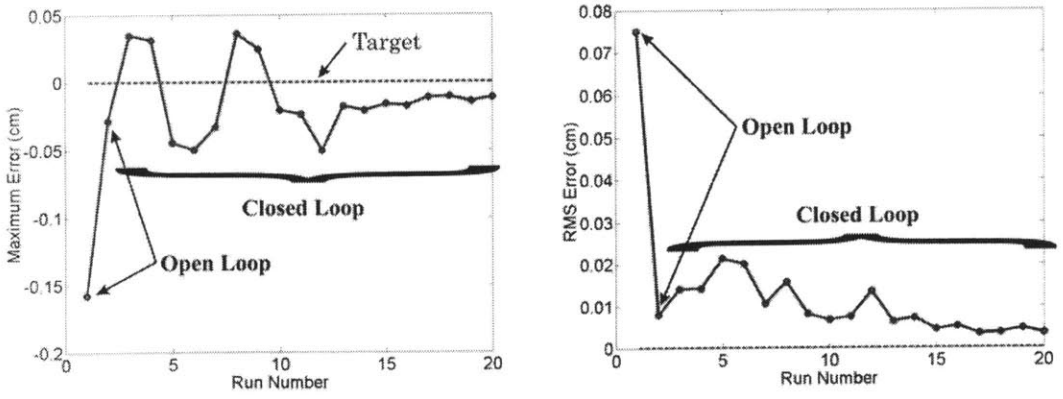


Figure 7-21 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. Maximum (based on magnitude) and RMS error shown, H_2 (LQG) designed controller.

7.5 H_∞ Controller

The solution to the H_∞ problem will contain the same order controller as the plant, which is identical to the H_2 problem. However, the H_∞ solution is the result of a different criterion function. The cost to minimize is expressed as

$$J = \sup_{\omega} \bar{\sigma}(T_{wz}(j\omega)) = \|T_{wz}(j\omega)\|_\infty$$

Equation 7-25

This means that we minimize the H_∞ norm of the transfer function from w to z, as opposed to the H_2 norm for LQG. It is interesting to note the difference between the 2 norm and the infinity norm. The 2 norm, Equation 7-22, is the integral over all frequencies. Minimizing it, therefore, will give the lowest average magnitude frequency response. The infinity norm, Equation 7-25, on the other hand, is concerned only with minimizing the largest singular value. Therefore, this criterion will peak the lowest "peak" in the frequency response, regardless of what happens in the rest of the frequency spectrum. The difference between the two norms or, more precisely, between their respective choices for a solution, is shown in Figure 7-22.

The selection of an H_2 or H_∞ controller is largely determined by the operating range and the type of disturbance affecting a process. Note that both the H_2 cost function, Equation 7-22, and the H_∞ cost function, Equation 7-25, are minimizing the closed-loop transfer functions from generalized disturbances to generalized errors. In this approach, the reference input is considered as a generalized disturbance, since it is an influence external to the process. Therefore, the cost functions do not discriminate between the operating input and disturbance input frequencies. When the closed-loop process has to operate over a wide range of frequencies or is subject to broadband disturbances, it is logical to select the H_2 controller to achieve a best overall response. If, however, only a

small operating range is required or the disturbance has limited frequency content, H_∞ is the natural choice for controller synthesis. As a simple example, consider a process which is asked to operate in a limited band near its natural frequency. Here, minimizing the peak of the frequency response is more important than achieving good performance over all frequencies, most of which will not influence the process. Therefore, H_∞ is the natural choice.

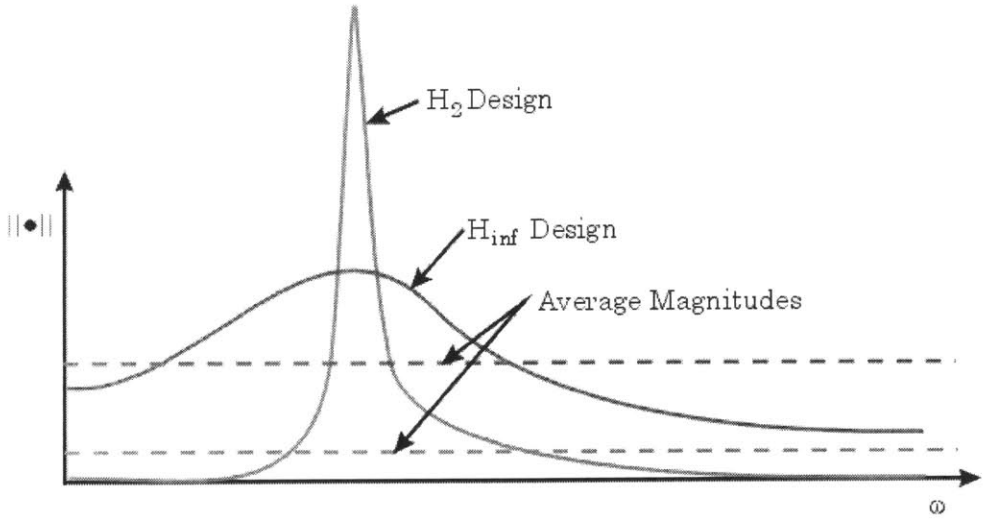


Figure 7-22 Schematic view of two frequency response functions and the difference between the optimal solution for the H_2 and H_∞ problem.

Like the H_2 controller, the H_∞ controller synthesis requires, [32],

1. (A,B_2) is stabilizable and (C_2,A) is detectable
2. $D^*_{12}D_{12}>0$ and $D_{21}D^*_{21}>0$
3. $\begin{bmatrix} A-j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank for all ω
4. $\begin{bmatrix} A-j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ has full row rank for all ω

where A, B, C, and D are defined in Equation 7-21.

7.5.1 H_∞ Simulation

To design the H_∞ controller, we specify a block diagram and use the *MATLAB* function "dhfsyn" to obtain the solution. Our block diagram is shown in Figure 7-23 with the scaling variable values shown in Table 7-3. Note that we uniformly scale the 110 signal vector such that the control cost $z_{control}$ is scaled by $K_{z2}*I$ where I is a $110x110$ identity matrix. As is shown, we have four generalized disturbance inputs and three generalized error outputs. Input u and output e are used by *MATLAB* as the signals into and out-of the controller and are thus not considered in the cost function, Equation 7-25.

The input-output pair w_{delta} and z_{delta} is used to describe the uncertainty in the plant matrix. As before, we describe uncertainty as 10% of the scaling matrix, Equation 7-13. Note that to be strictly proper, we should place the uncertainty between the unscaled Gaussian matrix and the scaling matrix. However, we take advantage of the diagonal structure of both the description of uncertainty and the scaling matrix to keep the plant matrix together. For us then, the plant matrix with uncertainty can be described as

$$K_P = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_{n^2} \end{bmatrix} \left([I] + 0.1 \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \delta_{n^2} \end{bmatrix} \right) \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

$$= \left([I] + 0.1 \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \delta_{n^2} \end{bmatrix} \right) \begin{bmatrix} s_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & s_{n^2} \end{bmatrix} \begin{bmatrix} Unscaled \\ Gaussian \\ Matrix \end{bmatrix}$$

Equation 7-26

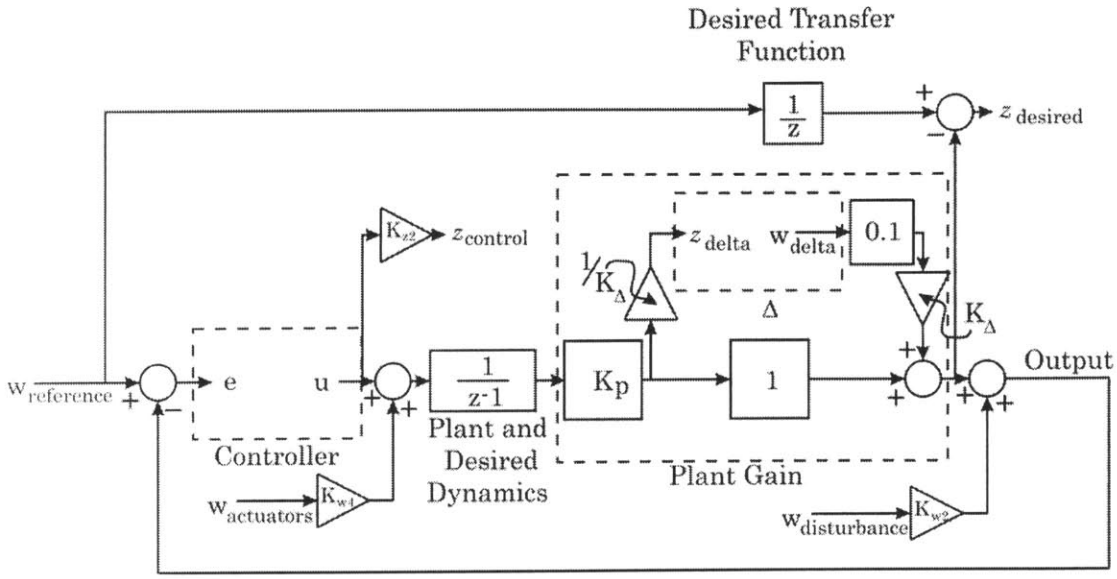


Figure 7-23 H_∞ design block diagram.

K_{z2}	K_{w2}	K_{w4}	K_Δ
0.01	0.0001	0.1	0.1

Table 7-3 H_∞ design parameters.

The remaining inputs and outputs require little additional explanation. Input $w_{disturbance}$ is the modeled disturbance input, $w_{reference}$ is the reference input, and $w_{actuators}$ is an actuator disturbance input. Note that input $w_{actuators}$ is also required to satisfy requirement number 4 for H_∞ synthesis. Output $z_{control}$ is the control cost and output $z_{desired}$ is a penalty for deviation from the desired transfer function of a one time step delay.

The results of a simulation with the above controller are shown in Figure 7-24. Both a noisy and a noiseless simulation are shown. A generic level of Gaussian noise with a standard deviation of 0.001cm is used. Because this controller has its own states, as did the H_2, we have chosen to initialize them according to the procedure described in Section 7.4.1. However, we have chosen to leave-out the registration procedure. As before, the first closed-loop part is run number one. The first two parts, -1 and 0, are experimentally formed parts from Vaughan, [39]. The choice of weights, Table 7-3, has resulted in a

quick response for the H_∞ controller and we see that the integral action needed to give zero steady-state error is retained. A simulation of 1,000 cycles of this process reveals a closed-loop to open-loop variance ratio of 1.95. When we use this in Equation 2-38, we estimate the loop gain as 0.97. This is then used in Equation 2-21 to estimate the expected settling time as 2 cycles. This is consistent with what we see in Figure 7-24.

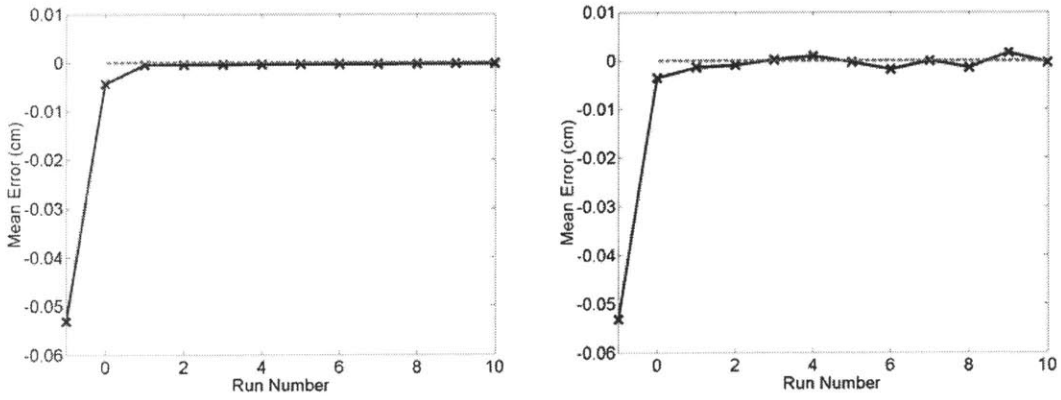


Figure 7-24 Simulated closed-loop runs of a 110 input, 110 output process with H_∞ designed controller. Noiseless simulation (left) and Noisy simulation (right).

The results of stability analysis are shown in Figure 7-25. We keep our familiar Δ form of ten percent uncertainty in each of the plant scaling values. As with the LQG design, we see that the maximum value of γ_{upper} is 0.1, which means that each of the scaling values can vary by ±100%. We also look at the response of the process when part registration is included. This had caused a significant deviation from the designed response with the H_2 controller. The results of including registration are shown in Figure 7-26. We can see that including the registration causes no visible deviation from the designed response, Figure 7-24.

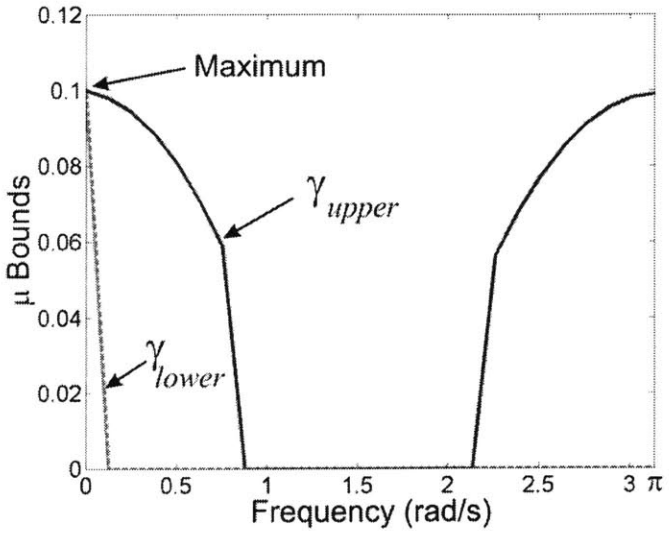


Figure 7-25 μ_Δ bounds as a function of frequency for a closed loop process with an H_∞ designed controller.

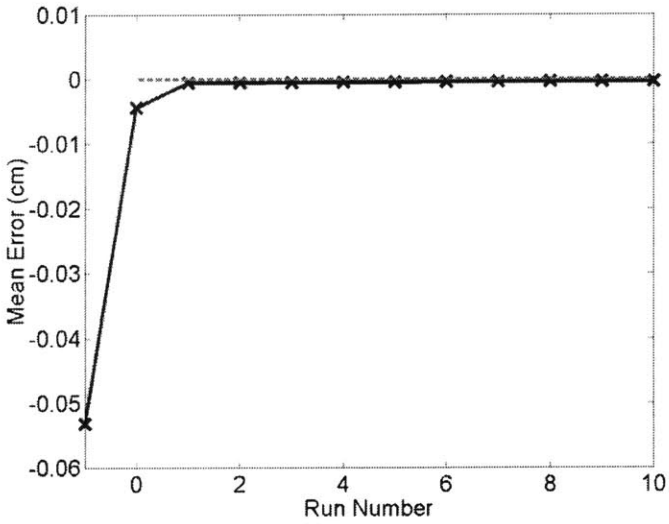


Figure 7-26 Closed-loop, noiseless response of the H_∞ controller with registration included.

7.5.2 H_∞ Experimental Validation

As with the previous controller designs, we validate our H_∞ controller design on the discrete-die sheet metal forming process described in 1.5.1. We use a 27 cm radius cylinder as the target part and again re-use the first two open-loop runs from Vaughan, [39]. As with the H_2 controller, we use these two runs to initialize the controller states.

The experimental results are shown in Figure 7-27 and Figure 7-28. We observe an initial overshoot on the first closed-loop part with no observed oscillation in subsequent runs. These remaining runs, which appear to be a slow oscillation in Figure 7-27, are within the observed noise level of the process. We can use runs four through six to estimate the process statistics. Using the *t* statistic, the mean of the mean error plot is estimated at -0.002 cm with 95% confidence bounds of -0.005 cm and 0.002 cm. Next, we calculate the variance of the mean errors using the χ^2 distribution. The nominal variance is 1.9×10^{-6} cm^2 with 95% confidence bounds of 5.1×10^{-7} cm^2 and 7.4×10^{-5} cm^2. This closed-loop variance is used with a set of eight open-loop runs performed by Vaughan, [39], to establish a closed-loop/open-loop variance ratio. This variance ratio is calculated using the *F* statistic, and is 0.3 with bounds of 0.008 and 2.1 at 95% confidence. These results lead us to conclude that our H_∞ controller design experimentally retains the property of zero steady-state error with a bounded increase in variance.

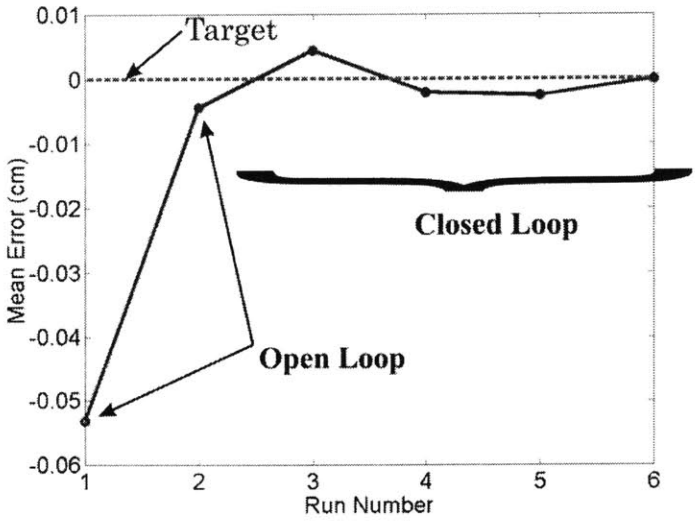


Figure 7-27 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. H_∞ designed controller.

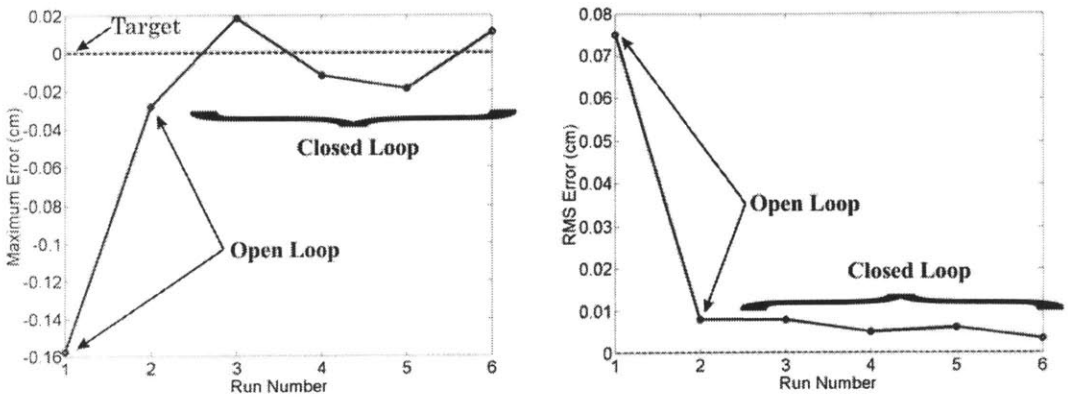


Figure 7-28 Experimental closed-loop runs of the sheet metal forming process described in Section 1.5.1. Maximum (based on magnitude) and RMS error shown, H_∞ designed controller.

7.6 Controller Conclusions

The work in prior chapters has led us to search for controllers that could be successfully and robustly implemented given our generic, poorly-conditioned plant gain matrix. In fact, the impracticality of inverting our plant gain matrix is what forced us to abandon our deadbeat analog controller design method presented in Section 3.2.1. Although many such controller design methods exist and more are created every year, we have settled on three which form a good starting set: LQR, LQG, and H_∞. Note that all designs had the desirable dynamics of the controller, integral action, embedded into the description of the plant prior to actual controller design. This ensured that all controller designs will have zero steady-state error. Because of the good agreement between the simulation and experimental results, we conclude that we have shown the ability to design well behaving controllers with predictable output mean and variance behavior.

CHAPTER

8

CONCLUSIONS AND FUTURE WORK

Having developed and experimentally validated our controller designs in the previous chapter, we look at two additional designs which exist in literature. First, our designs come on the heels of a controller which has been used for many years in MIT's Manufacturing Process Control Laboratory. We have introduced this controller as the spatial coordinate algorithm (SCA) in Section 5.2. The distinguishing characteristic of this controller is that it assumes no coupling among the outputs. The controller gain matrix is therefore nonzero only on the diagonal. In Section 5.2, we have seen that the SCA model was, by a slim margin, the worst predictor of outputs when the inputs are given. However, as stated in Section 1.1, control can compensate for imperfections in the model. This serves to explain why the SCA controller has been used successfully to stretch form parts. It may also be the case that sufficiently small changes are made between subsequent dies that the effect of coupling is minimal in this process.

A sample run of parts experimentally formed using the SCA algorithm is included in Figure 8-1, [2, 39]. We can see that this algorithm maintains the ability to center the process. Note that, although the SCA algorithm refers to designing the controller gain matrix by inverting the uncoupled plant gain matrix, desirable integrator dynamics are embedded into the final form of the controller. The calculated process mean is 0.0001cm

and the variance ratio of closed-loop and open-loop runs is computed to be 3.5 with 95% confidence bounds of 0.6 and 16.6, [2, 39].

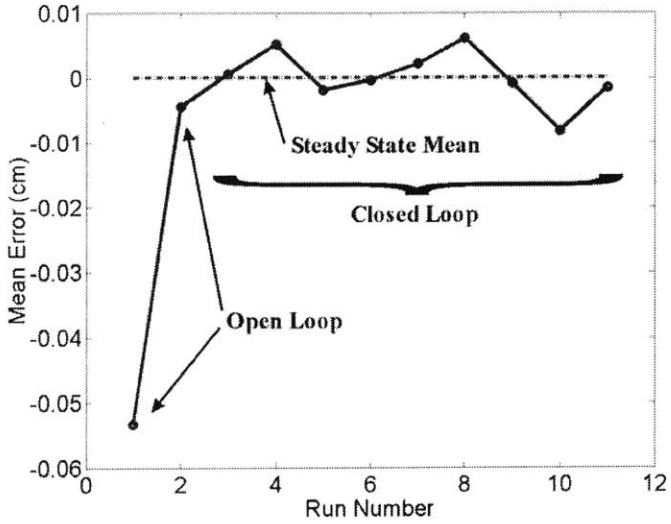


Figure 8-1 SCA algorithm experimental application results, [2].

The desire to be able to analyze the SCA controller was one of the motivating factors for the research in this thesis. We can analyze this controller with the tools developed in Chapter 6. To do this, we assume our familiar form of uncertainty, Equation 7-13. The μ_Δ bounds are shown in Figure 8-2. As we can see, the largest value of γ_{upper} is 0.1, which translates to a maximum allowed variation of 100% in each of the scaling factors. This maximum variation matches the stability bounds achieved with both the H_2 and H_∞ controllers and is greater than the stability region achieved with either of the LQR controllers. Because of the multi-year success with this controller plant pair, this result is not too surprising. Overall, because of its good performance and ease of implementation, we consider this controller to be a good candidate design for weakly coupled processes.

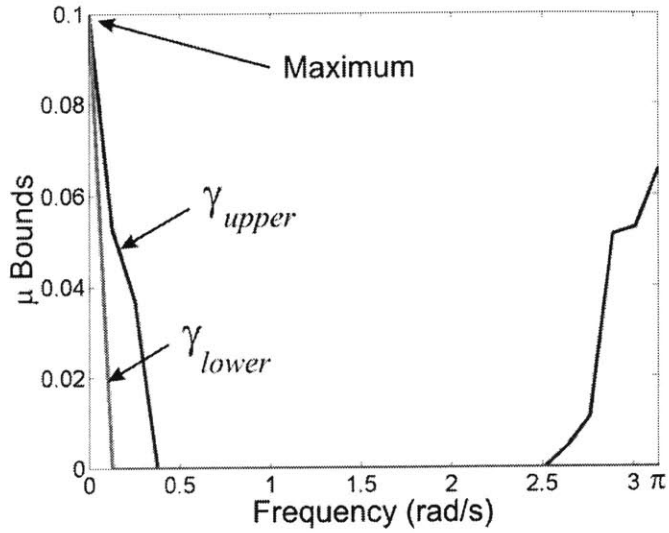


Figure 8-2 μ_Δ bounds as a function of frequency for a closed loop process with an SCA designed controller.

A second design for controllers comes from Rajagopal and Del Castillo, [43], who look at MIMO controller designs for non-square plants. This is an extension of their work done for square plants, [44]. Rajagopal and Del Castillo do not confine themselves to our particular class of process, and instead choose to address a general class of MIMO processes. The work presented in either publication relies on taking the pseudo inverse of the plant gain matrix. This is problematic for our class of process since the pseudo inverse of a matrix A is equal to the inverse of that matrix when it is square and non-singular. As such, the pseudo inverse will suffer from the same error propagation properties associated with taking the inverse of a matrix with a large condition number, making it impractical for our processes. Note that taking the inverse of the plant matrix was the method of designing the controller gain matrix that we obtained from the ideal MIMO extension presented in Chapter 3. To this date, we have not found publications by other authors discussing the problem of applying run-by-run control to our class of spatially coupled processes.

With our controller designs in hand, we can also test a statement we made back in Section 6.2, that the size of Δ could be used to describe our ability to use a single plant model for different target outputs. To test this statement, we use historical data for a saddle shaped target part from Valjavec, [45]. Our testing procedure will be to calibrate a plant model to the saddle shape, and use this plant matrix in a *MATLAB* simulation with the controllers in Chapter 7. Note that the controllers were designed for a cylindrical target shape.

The target shape is shown in Figure 8-3 and is described by radii of 81.4cm and 114.3cm in the two directions. Because we have access to Valjavec's data files, we use his first two runs to establish a GIC matrix that is calibrated for this saddle shape, Equation 5-4. Note that, to fit within our definition of uncertainty we keep a coupling distance of σ=1, as in Chapter 7, and recalculate the scaling matrix. The scaling values are shown in Figure 8-4 according to their position in a 10x11 grid of inputs. Note that we can use this representation since the scaling matrix is diagonal, i.e. we only need to show the non-zero entries. We also calculate the deviation from the nominal, cylinder-configured, scaling values according to

$$\% = \left(\frac{s_{cylinder} - s_{saddle}}{s_{cylinder}} \right) * 100$$

Equation 8-1

for each pair of the 110 scaling values. The percent deviation is shown in Figure 8-5. Immediately, we see that the deviation in scaling values is significant. The maximum percentage of deviation is 75%. This value exceeds the guaranteed stability margins of both LQR controllers, but remains within the boundaries of both the LQG and H_∞ controllers.

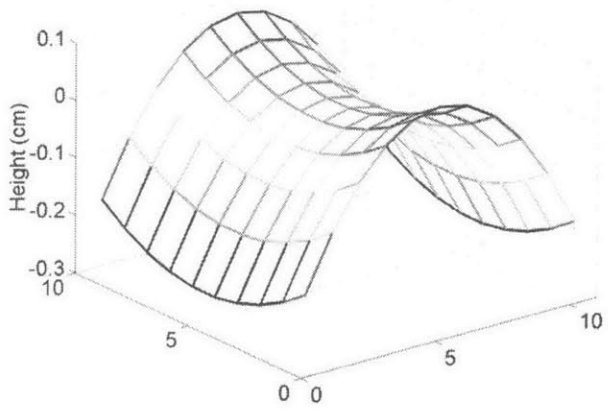


Figure 8-3 Saddle target shape from Valjavec, [45].

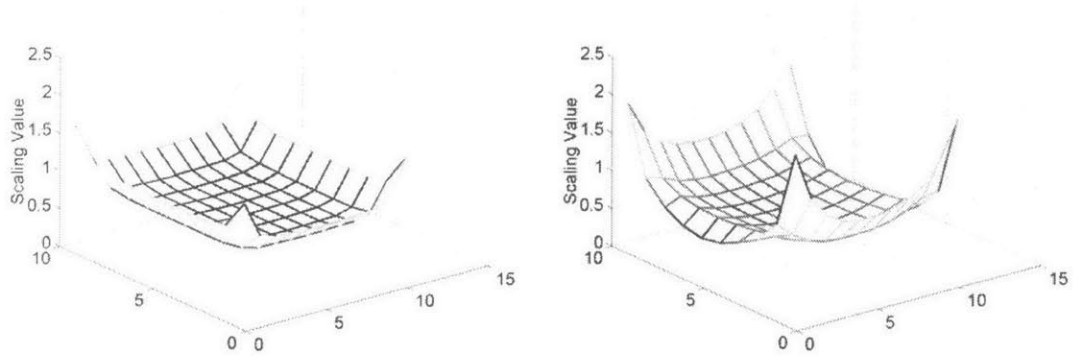


Figure 8-4 Diagonal scaling matrix entries shown according to their position in a 10*x*11 grid of inputs. Cylinder scaling (left), saddle scaling (right).

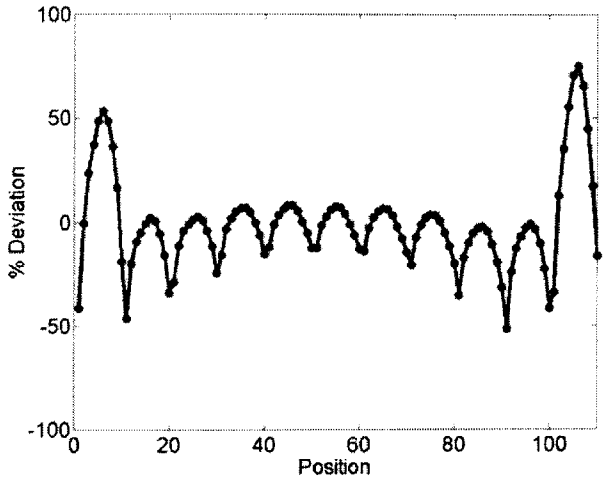


Figure 8-5 Percent deviation of each scaling value between the cylinder- and saddle-calibrated plant matrices.

Now, we will test our controllers by defining a new plant matrix with these scaling values without redesigning the controllers from Chapter 7. The tests are carried out using a *MATLAB* simulation in a noise-free environment but with the effects of the registration included. The results for all controllers are shown in Figure 8-6. We see that all controllers remain stable with some performance degradation in each case. It is interesting to note that the H_2 controller, the worst performing at the nominal setpoint, also suffered the least performance degradation due to the change in the process matrix. The fact that the LQR controllers remained stable even though this variation is outside the guaranteed stability bounds reiterates our closing statements in Chapter 6: the difference between being within the μ_Δ bound and outside it does not delineate the difference between stability and instability; it delineates the difference between robust stability and non-robust stability. Therefore, even though we have a case that is outside the robust stability bounds of the LQR controllers, we are not looking at one of the matrices that cause these closed-loop processes to go unstable.

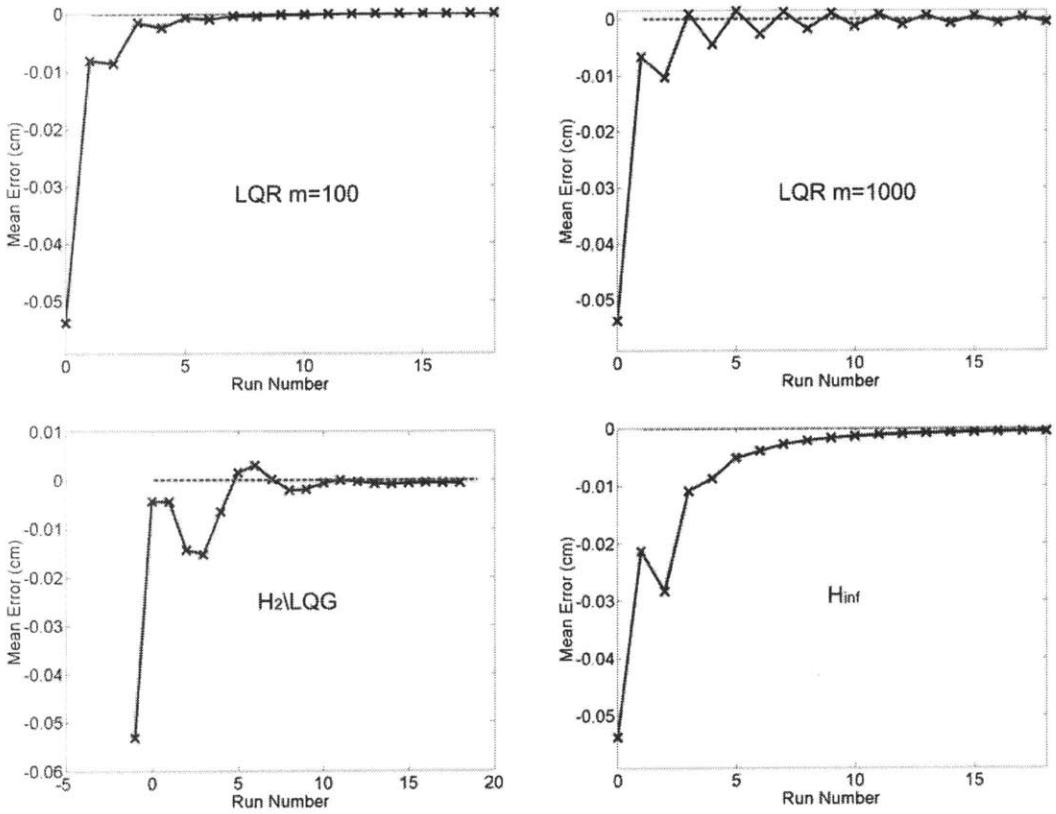


Figure 8-6 Process response plots with a saddle-configured plant matrix and cylinder-configured controller matrices.

8.1 Conclusions

The work presented in this thesis was performed to further the state of the art in cycle-to-cycle and run-by-run control of manufacturing processes. In Chapter 1, we first motivated the use of control stating that it would allow us to use imprecise models while still getting good performance. Control could also be used to reject or reduce the effects of disturbances. Next, we outlined the specific area which we would address, the control of spatially coupled MIMO processes, and finally listed a number of candidates that could benefit from this work.

Before proceeding towards addressing MIMO processes, we used Chapter 2 to lend a control engineering perspective to the design of both the single EWMA and double EWMA controllers. We first tackled the EWMA and integral controller, redeveloping it from both the filtering and discrete control point of view. It was interesting to note that the two different approaches yielded nearly the same controller. The only difference between them was in their response to a change in the target input. This is also the only difference between cycle-to-cycle and run-by-run control. Next, we laid out a number of controller design factors including time response, speed and shape, and variance change. We then moved on to examine the double EWMA controller. We showed how the controls perspective reduced the scope of the problem to a minimum set and showed how the root-locus technique could be used to visualize quickly all possible design combinations. We also discovered that overshoot could be a problem when applying this controller. We believe that this chapter helps to make CtC and RbR control more appealing to industry because it clearly lays out the development and design procedure needed to achieve the desired time response and variance change characteristics.

In Chapter 3, we extended the results obtained with single input-single output processes to multiple input-multiple output ones. We saw that the integral controller was still needed to reject step disturbances and that simulation trials with the smallest MIMO process, a $2x2$, revealed a still bounded change in variance. We also saw that the MIMO integral controller problem is underconstrained with pole-placement. To solve this problem, we used a deadbeat-analog design procedure which requires us to invert the plant gain. With this approach, we would decouple the MIMO process and achieve one

time step settling. We concluded by stating that all we would need to design this controller was knowledge of the plant.

Chapter 4 outlined our attempts at identifying the plant. We used an *ABAQUS* simulation, [27], to try to discover the nature of coupling within the process of sheet metal stretch forming. Unfortunately, the results of these experiments were inconclusive. We settled on trying to discover the tolerance of a MIMO processes to uncertainty through *MATLAB* simulations with a 2x2 process. These simulations revealed that our candidate process was tolerant to significant variation without much degradation in performance. Encouraged by these results, we used Chapter 5 to develop and test a generic, easily calibrated plant model, Gaussian influence coefficients. We saw how this plant model did the best job of predicting process outputs, given the inputs, when compared with two previously developed and tested process models. Unfortunately, we also discovered that our plant model was severely ill-conditioned, preventing us from being able to reliably take its inverse. Large condition numbers were found to be a problem with all models that described limited regional coupling. We finished the chapter by saying that an alternate method needed to be found for controller design.

Before looking at alternate controllers, we decided to take an educated look at establishing the tolerance to uncertainty of MIMO processes (Note that, prior to Chapter 6, we had used simulation for this purpose and that this approach is neither rigorous nor practical for large MIMO processes). We first introduced the small gain theorem and its limitations. Then, we looked at how applying structured singular values allows us to obtain tighter estimates of the stability bounds by using our knowledge of the nature of

uncertainty. We also saw how we could include performance analysis by re-posing it as a stability problem.

Finally, in Chapter 7, we started to look at controller design methods different from plant inversion. Note that we also had to be careful to avoid those methods that relied on plant inversion in any of their solution steps. After examining and dismissing a robust pole placement approach, we designed an LQR controller. This was done because our experimental test bed is not subject to a significant amount of noise and we have easy access to the process states. In fact, we are able to measure the states directly. The LQR controller is a gain-only controller. Therefore, no additional dynamics were introduced into the loop. The obtained results, Figure 7-10 and Figure 7-11, show that the controller is stable and provides good performance.

Next, we looked at the H_2 (LQG) controller. We had done this because the process states may not be directly accessible in all processes which can benefit from the work in this thesis. We have seen that the LQG controller is just a combination of the LQR controller and a Kalman filter which is used to give optimal estimates of the process output and states. Theoretically, we would not expect significant degradation of performance from the inclusion of the Kalman filter. However, experimentally we saw that the registration procedure which was needed for our experiments caused significant oscillations to occur, Figure 7-20 and Figure 7-21. Because of this, we recommend that an observer not be used unless it is actually necessary.

Finally, we designed an H_∞ controller and built-in, at the controller design stage, tolerance to uncertainty. Both our simulation and experimental results yielded very good

performance with quick settling, Figure 7-27 and Figure 7-28. The good results obtained with this controller were comparable to those obtained with the successful LQR case.

We think that the work presented in this thesis presents a significant addition to the knowledge base for CtC and RbR control of MIMO processes. We believe that we have made a number of contributions. First, the development and use of a general, intuitive, and easy to calibrate plant model for spatially coupled plants. Second, the application of LQR, LQG, and H_∞ theory to large, multivariable, coupled processes belonging to our class of CtC, or RbR, processes and, third, the experimental validation of these algorithms on our class of process, which we have not previously seen in literature. Finally, we note that, although we have tended to use square plants, ones with an equal number of inputs and outputs, LQR, LQG, and H_∞ do not require this condition for their successful application.

8.2 Future Work

As we have made our contributions, we have also left work undone. First, we looked at only a preliminary set of controllers. We picked these because we believe them to be a good starting set. An in depth review of more controllers may reveal that other controller forms are superior to the three discussed in Chapter 7. As an example, consider an adaptive controller for on-line estimation of the process gain. Such a controller may be unnecessarily complicated for processes with step disturbances and may suffer from poor estimation if the signal to noise ratio is unfavorable. However, it may be very useful for processes that are subject to long runs and significant drift disturbances. In Chapter 2 we showed that a double EWMA controller can be used to achieve zero steady state error under these conditions. However, one of the underlying assumptions of using the double

EWMA controller is a small change in the process gain. If this change is large, this controller will either perform poorly or go unstable. An adaptive controller could compensate for large process gain changes to maintain good performance and stability over a wide range. This can be seen through a single input-single output example. Consider two nonlinear processes with input-output response characteristics shown in Figure 8-7 with an initial input operating range of (0,2). We can estimate the process gains, k_p, as 0.3 for the process on the left and 1 for the process on the right. Since we want good performance, we will select the controller according to the deadbeat analog gain selection method making our gains, k_c, 3.3 and 1, respectively. Now, suppose that the drift disturbance is significant enough and we run long enough to move us away from that range. We see that the process on the left tends to have its gain go to zero. By looking at Figure 2-5 and applying Equation 2-21, we see that the settling time for new target inputs or to recover from disturbances goes to infinity. In contrast, the gain for the process on the right tends to increase and rise above 2 when the input is above 6. We can check Table 2-1 to see that above 6, the process will go unstable. This effect is seen in Rajagopal and Del Castillo, [43], when their simulated nonlinear process goes unstable after a large number of runs.

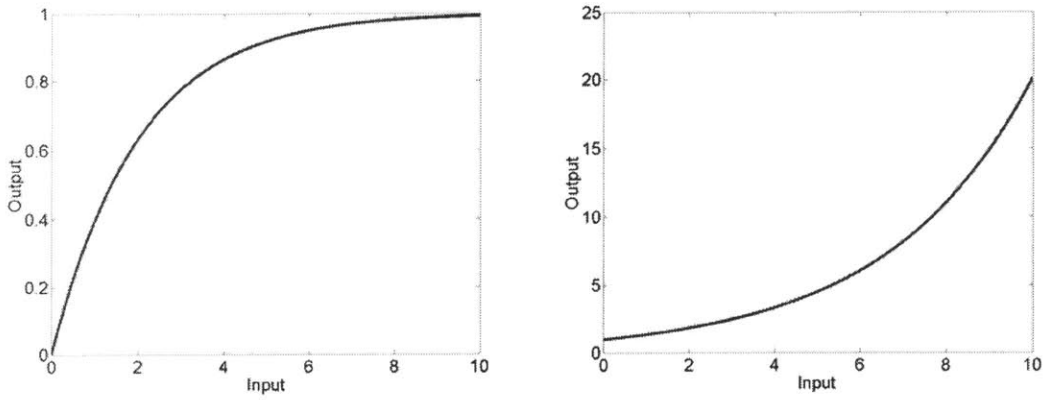


Figure 8-7 Example input-output response characteristics used to illustrate adaptive control.

In addition to different controllers, we note that our choice of controllers need not be restricted to our class of problem. We present it as a separate research problem to look at the effects of using the LQR, LQG, and H_∞ controller for run-by-run control of arbitrarily coupled MIMO processes. An example of such a process is injection molding where the outputs (critical dimensions) are not co-located with the inputs (pressure, material temperature, hold time, etc.). As we have mentioned in Section 8.1, our controller methods do not require the candidate process to have an equal number of inputs and outputs.

Finally, we suggest a further investigation into determining the variance change/loop gain trade-off for MIMO processes. We have used simulations with a small MIMO matrix to compare MIMO variance change to SISO theoretical limits. Our comparisons, Figure 3-6 and Figure 3-7, showed that the MIMO processes are still bounded by these limits and, many times, are actually lower than them. Our own experimental estimates of process variance change, although crude showed that the closed-loop process appeared to behave better than expected. It would be interesting to know if a particular controller formulation can improve the variance change while maintaining the same transient response form as another one.

REFERENCES

1 DeVor, R.E., Chang, T., Sutherland, J.W., "Statistical Quality Design and Control, Contemporary Concepts and Methods." *Upper Saddle River, NJ: Prentice Hall*, 1992.

2 Rzepniewski, A.K., Hardt, D.E., Vaughan, C.D., "Cycle-to-Cycle Control of Multivariable Manufacturing Processes with Process Model Uncertainty." *IMECE200, 2004 ASME International Mechanical Engineering Congress and Exposition*, Anaheim, CA, Nov 2004.

3 Shewhart, W.A. "Economic Control of Quality of Manufactured Product." *American Society for Quality*, Dec. 1980.

4 Montgomery, D.C., "Introduction to Statistical Quality Control." *John Wiley & Sons, Inc.*, 2001.

5 Sachs, E., Hu, A., Ingolfsson, A., "Run by Run Process Control: Combining SPC and Feedback Control." *IEEE Transactions on Semiconductor Manufacturing*, 1995, vol. 8, no. 1, pp. 26-43.

6 Muth, J.F., "Optimal Properties of Exponentially Weighted Forecasts." *Journal of the American Statistical Association*, Jun. 1960, vol. 55, Issue 290, pp. 299-306.

7 Box, G.E., Jenkins, G.M., "Time Series Analysis – Forecasting and Control." *Oakland, CA: Holden-Day*, 1976.

8 Box, G., Luceno, A., "Discrete Proportional-Integral Adjustment and Statistical Process Control." *Journal of Quality Technology*, Jul. 1997, vol. 29, no. 3, pp. 248-260.

9 Hardt, D.E., Siu, T.-S., "Cycle to Cycle Manufacturing Process Control." *First Annual SMA Symposium*, Singapore, Jan. 2002.

10 Kuo, B. C., "Automatic Control Systems." *Englewood Cliffs: Prentice-Hall*, 1995.

11 Valjavec, M. and Hardt, D.E., "Closed-loop Shape Control of the Stretch Forming Process over a Reconfigurable Tool: Precision Airframe Skin Fabrication." *Proceedings of the ASME Symposium on Advances in Metal Forming*, Nashville, Nov. 1999.

12 Papazian, J., "Tools of Change", *Mechanical Engineering*, Feb. 2002, pp. 31-40

13 De Mello, A., "Plastic Fantastic?" *Journal of the Royal Society of Chemistry*, 2002, pp. 31N-36N.

14 Kalpakjian, S., "Manufacturing Engineering and Technology." *Reading: Addison-Wesley*, 1995.

15 Moyne, J., Del Castillo, E., Hurwitz, A.M., "Run-to-Run Control in Semiconductor Manufacturing." *New York: CRC Press*, 2001.

16 Box, G., Luceno, A., "Statistical Control by Monitoring and Feedback Adjustment." *New York: Wiley-Interscience*, 1997.

17 Siu, G. T.-S., "Cycle to Cycle Feedback Control of Manufacturing Processes." *SM Thesis in Mechanical Engineering, MIT*, Feb. 2001.

18 Rzepniewski, A.K., Hardt, D.E., "Multiple Input-Multiple Output Cycle-to-Cycle Control for Manufacturing Processes." *Second Annual SMA Symposium*, Singapore, Jan. 2003.

19 Chen, A., Guo, R.S., "Age-Based Double EWMA Controller and Its Application to CMP Processes." *IEEE Transactions on Semiconductor Manufacturing*, Feb. 2001, vol. 12, no. 1, pp. 11-19.

20 Del Castillo, E., "Long Run and Transient Analysis of a Double EWMA Feedback Controller." *IIE Transactions*, 1999, vol. 31, no. 12, pp. 1157-1169.

21 Garcia, C.E., Morari, M., "Internal Model Control 2, Design Procedure for Multivariable Systems." I*ndustrial Engineering, Chemical Design and Development*, 24, pp. 472-484.

22 Friedland, B., "Control System Design: an Introduction to State-Space Methods." *Boston: McGraw-Hill*, 1986.
23 Kosut, R. L., de Roover, D., Emami-Naeini, A., Ebert, J. L., "Run-to-Run Control of Static Systems." *Proc. of the 37th IEEE Conference on Decision and Control*, 1998, pp. 695-700.
24 Dore, S., Goussard, Y., "Experimental Determination of CT Point Spread Function Anisotropy and Shift-Variance." *Proceedings of the 19th Annual International Conference of the IEEE*, 1997, vol. 2, pp. 788 -791.
25 Hebert, T., Murphy, P., Moore, W., Dhekne, R., Wendt, R., Blust, M., "Experimentally Determining a Parametric Model for the Point Source Response of a Gamma Camera." *IEEE Transactions on Nuclear Science*, 1993, vol. 40, issue 4, part 1-2 , pp. 967 -971.
26 Wan, H., Aarsvold, J., O'Donnell, M., Cain, C., "Thermal Dose Optimization for Ultrasound Tissue Ablation." *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 1999, vol. 46, Issue 4 , pp. 913 -928.
27 Socrate, S., and Boyce, M.C., "A Finite Element Based Die Design Algorithm for Sheet Metal Forming on Reconfigurable Tools." *Proceedings of the ASME Symposium on Advances in Metal Forming*, Orlando, Nov. 2000.
28 Hardt, D.E., Norfleet, N.A., Valentin, V.M, and Parris, A., "In-Process Control of Strain in a Stretch Forming Process." *Proceedings of the ASME Symposium on Advances in Metal Forming*, Orlando, Nov. 2000.
29 Rzepniewski, A.K., Hardt, D.E., "Gaussian Distribution Approximation for Localized Effects of Input Parameters." *Second Annual SMA Symposium*, Singapore, Jan. 2003.
30 Norfleet, W., "Algorithms for Closed-Loop Shape Control." *SM Thesis in Mechanical Engineering, MIT*, June 2001.
31 Strang, G., "Introduction to Linear Algebra." *Wellesley, MA: Wellesley-Cambridge Press*, 1998.
32 Zhou, K., Doyle, J.C., *Essentials of Robust Control*, Prentice Hall, New Jersey, 1998.
33 Maciejowski, J.M., "Multivariable Feedback Design." *Reading, MA: Addison-Wesley Publishers Ltd*, 1989.
34 *MATLAB* μ-Analysis and Synthesis Toolbox, July, 1993.
35 Massachusetts Institue of Technology Open Courseware class notes, course 6.241: Dynamic Systems and Control.
36 Franklin, G.F., Powell, J.D., "Digital Control of Dynamic Systems." *Reading, MA: Addison-Wesley Publishers Ltd*, 1980.
37 Kautsky, J., Nichols, N.K., Van Dooren, P., "Robust Pole Assignment in Linear State Feedback," *International Journal of Control*, 1985, vol. 41, no. 5, 1129-1155
38 Webb, R.D., Hardt, D.E., "A Transfer Function Description of Sheet Metal Forming for Process Control," *ASME J. of Engineering for Industry*, 1990.
39 Vaughan, C.D, "Cycle-to-Cycle Control of Reconfigurable Die Sheet Metal Forming." *SM Thesis in Mechanical Engineering, MIT*, June 2004.
40 *MATLAB* Control System Toolbox User's Guide, July 1992.
41 *MATLAB* Robust Control Toolbox Reference.
42 Pi, A., "Effects of Uncertainty on Closed Loop Shape Control in Stretch Forming." *SM Thesis in Mechanical Engineering*, MIT, September 2002.
43 Rajagopal, R., Del Castillo, E., "An Analysis and MIMO Extension of a Double EWMA Run-to-Run Controller for Non-Squared Systems." *International Journal of Reliability, Quality and Safety Engineering*, 2003, vol. 10, no. 4, pp. 417-428.
44 Del Castillo, E., Rajagopal, R., "A Multivariate Double EWMA Process Adjustment Scheme for Drifting Processes." *IIE Transactions*, 2002, vol. 34, pp. 1055-1068
45 Valjavec, M., "A Closed-Loop Shape Control Methodology for Flexible Stretch Forming Over a Reconfigurable Tool." *Ph.D. Thesis in Mechanical Engineering, MIT*, February, 1999.

APPENDIX

A

MINIMUM MEAN SQUARED ERROR AND EWMA

Muth, [6], and Box and Jenkins, [7], have shown that the EWMA controller results in minimum mean squared error at the output if the disturbance can be modeled as an integrated moving average, IMA(1,1). Here, we review their development.

We start with the control equation, Equation 2-16 relating the control input, u, to the error, $\tilde{e}$, and disturbance, d,

$$u_t - u_{t-1} = -\frac{w}{g}\tilde{e}_t = -\frac{w}{g}\left(d_t - \tilde{d}_{t-1}\right)$$

Equation A-1

Where w is the EWMA control weight and g is the plant gain. We write it for two consecutive cycles to get

$$\tilde{e}_t - \tilde{e}_{t-1} = d_t - d_{t-1} - \tilde{d}_{t-1} + \tilde{d}_{t-2}$$

Equation A-2

we use the EWMA estimate of the future disturbance, Equation 2-14,

$$gu_t = -\tilde{d}_t$$

Equation A-3

and substitute back into Equation A-2 to obtain

$$\tilde{e}_t - \tilde{e}_{t-1} = d_t - d_{t-1} + gu_{t-1} - gu_{t-2}$$

Equation A-4

Now we can use Equation A-1 again

$$\tilde{e}_t - \tilde{e}_{t-1} = d_t - d_{t-1} - w\tilde{e}_{t-1}$$

Equation A-5

Bringing all common terms to one side, we get

$$d_t - d_{t-1} = \tilde{e}_t - (1-w)\tilde{e}_{t-1}$$

Equation A-6

This result can be compared the IMA(1,1) disturbance model is given by

$$d_t - d_{t-1} = a_t - \theta a_{t-1}$$

Equation A-7

where a are normal, identically distributed, independent (NIDI) "shocks" and θ is a smoothing constant. We immediately see that the process error, $\tilde{e}$, must therefore be the same as the NIDI shocks a, proving the MMSE property.

APPENDIX

B

FREQUENCY LIMITED PARTS

The discussion of model precision in Section 5.2 leads to a question of the limit of a discrete-die machine itself. Because a Fourier series can be used to approximate any shape, a "wavelength" approach is taken to determining the precision limit of the machine's forming ability. The thought behind this approach is that any wavelength below the minimum controllable wavelength is due to noise effects (such as measurement noise) and should thus be filtered out.

During Cycle to Cycle operation, parts need to be removed from the machine, transported to a measurement device, and measured to "close the control loop" and provide feedback. This lack of centralization of the process increases the likelihood of extra noise sources within the control loop. A discrete frequency decomposition approach, discrete Fourier series, is taken to filter out "known and uncontrollable" noise effects. The two-dimensional Fourier transform takes the form:

$$F_{xy} = \sum_{i=0}^{n/2} \sum_{j=0}^{m/2} C_{ij} \cos\left(\phi_{ij} + \frac{2\pi i}{\tau_x} x + \frac{2\pi j}{\tau_y} y \right)$$

Equation B-1

where n and m are the number of points in the X and Y direction, respectively, and x and y are points on the surface to be approximated. In order to apply the above equation, it is necessary to know when to end the given series. Figure B-1 shows the minimum

controllable wavelength as a function of pin spacing; which is equal to two pin spaces. Note that this implies that the series will be terminated before $i=n/2$ or $j=m/2$ if those wavelengths are too small.

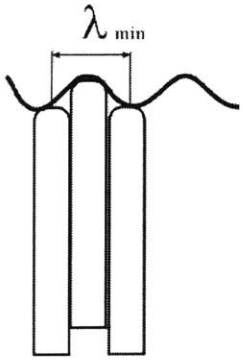


Figure B-1 Controllable wavelength limit for a discrete-die forming machine.

The frequency limiting approach is applied to a cylindrical part in order to observe the forming limit of the current forming machine. Note that the target part has a 6.65 in radius and the forming pins are 0.5 in square. The target part and the frequency limited part are shown in Figure B-2. The difference between the two parts is shown in Figure B-3. Note that the maximum error magnitude is approximately 0.6 mm. This is in opposition to the stated objective of a finished part having a maximum error magnitude of 0.25 mm. The discrepancy is caused by interpolating effects of material placed between the metal sheet to be formed and the pins themselves. Although this interpolator does not change the number of control points, it does change the pressure points between the pins and the metal, effectively changing the spacing of the pins' influence. This change in location of the pressure points means that some regions are able to control wavelengths that are smaller than the minimum predicted, while others are able to control only those which are larger than the minimum predicted.

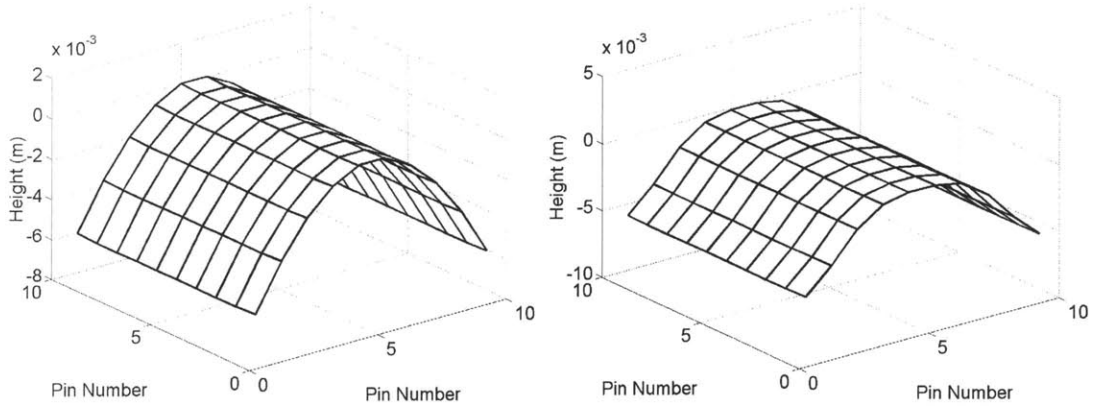


Figure B-2 Target parts. Full frequency (left) Limited frequency (right).

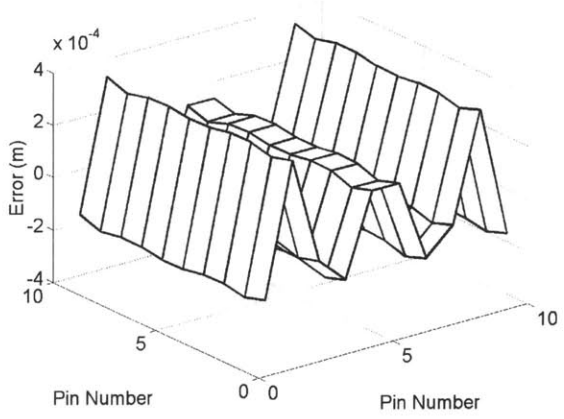


Figure B-3 Difference between the full-frequency and frequency-limited parts.

Even though we have discovered a flaw in the way we think about frequency limiting the input parts, it is still worthwhile to note how the maximum error magnitude changes as a function of the input radius. The target radii in Section 5.2 are arranged so that the 6.65in radius part is number one. The results of this investigation are shown in Figure B-4. As one would expect, larger radii parts are less affected by high frequency components and are thus better approximated by the frequency limited series.

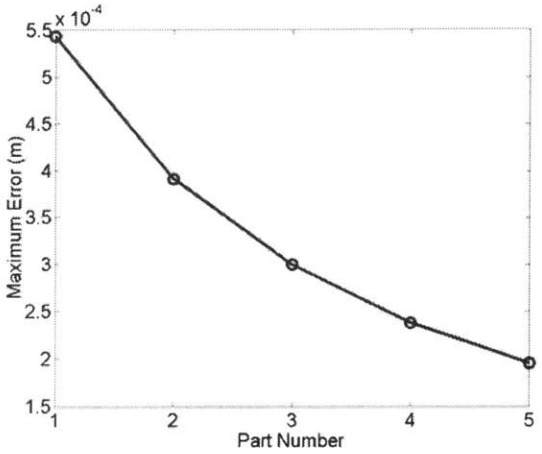


Figure B-4 Maximum error magnitude between a full-frequency and frequency-limited part. Parts ranging from 6.65in (#1) to 14.65in (#5) radius, in 2in increments.

APPENDIX

C

LQR CONTROLLER STRUCTURE

Since we have decided on a physically intuitive structure for the plant matrix, a natural question to ask is about the form of the controller gain matrix solution, i.e., we are curious as to the controller matrix structure. To answer this, we will look at the answers provided by the LQR algorithm. A relative weight of m=100 is assumed. To visualize the gains, we assume a *10x10* bundle of inputs (thus 100 total inputs) and we assume 100 outputs. The visualization will be performed by looking at the 45th row of the gain matrix, which correlates all errors to one of the four center pins in the *10x10* bundle, i.e.

$$\underline{u}^* = G_c^* \underline{e} \Leftrightarrow \begin{bmatrix} \vdots \\ u_i^* \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ Row \quad i \\ \vdots \end{bmatrix} \begin{bmatrix} \\ \underline{e} \\ \\ \end{bmatrix}$$

Equation C-1

Note that, in Equation 7-4 we have assumed that our reference was zero. It should also be noted that the controller gain matrix has been found to be symmetric.

Figure C-1 shows both the plant and controller gain matrix entries as a function of the column number for different values of the coupling spread, σ. We can see that for σ<1.23 there is a banded structure to the controller to correspond with the banded structure of the plant. We can attempt to interpret this result such that a region of errors will affect a single input much like a region of inputs affects a single plant output. Equivalently, by

symmetry of both the controller and plant matrices, it shows how a single error affects a region of inputs much like a single input affects a region of plant outputs.

Figure C-1 also shows that the banded structure of the controller matrix disappears once the matrix exceeds a coupling spread σ=1.22. Although it is not physically appealing, this structure is expected mathematically when we wish to decouple a matrix. It has been shown that to fully decouple the system, the controller matrix must be the inverse of the plant. It is well known, [31], that the inverse of a banded matrix is full of nonzeros. Thus, although the LQR method is not taking a direct inverse of the plant, we can accept a non-banded structure of the controller matrix.

As shown in Figure C-2 and Figure C-3, the solutions obtained by simulating the closed-loop behavior of the process with varying σ do not show any drastic effects on the process outputs due to the change in the controller matrix structure. Even so, we wonder why the change happens. One theory is that higher coupling values simply exceed the dimension of the process under considerations, i.e. the 3σ bounds are greater than the size of the bundle under consideration. To evaluate this theory, a *15x15* input bundle is chosen. Figure C-4 shows the plant and controller gain matrix entries for the 105^{th} input, the equivalent of the 45^{th} input presented in Figure C-1. As we can see, even though the full 3σ spread is easily captured by the process under consideration, the controller gain matrix structure remains non-banded. In our research, we have not found a satisfactory answer to the question of controller structure and present it here as a curiosity and as a subject of future work.

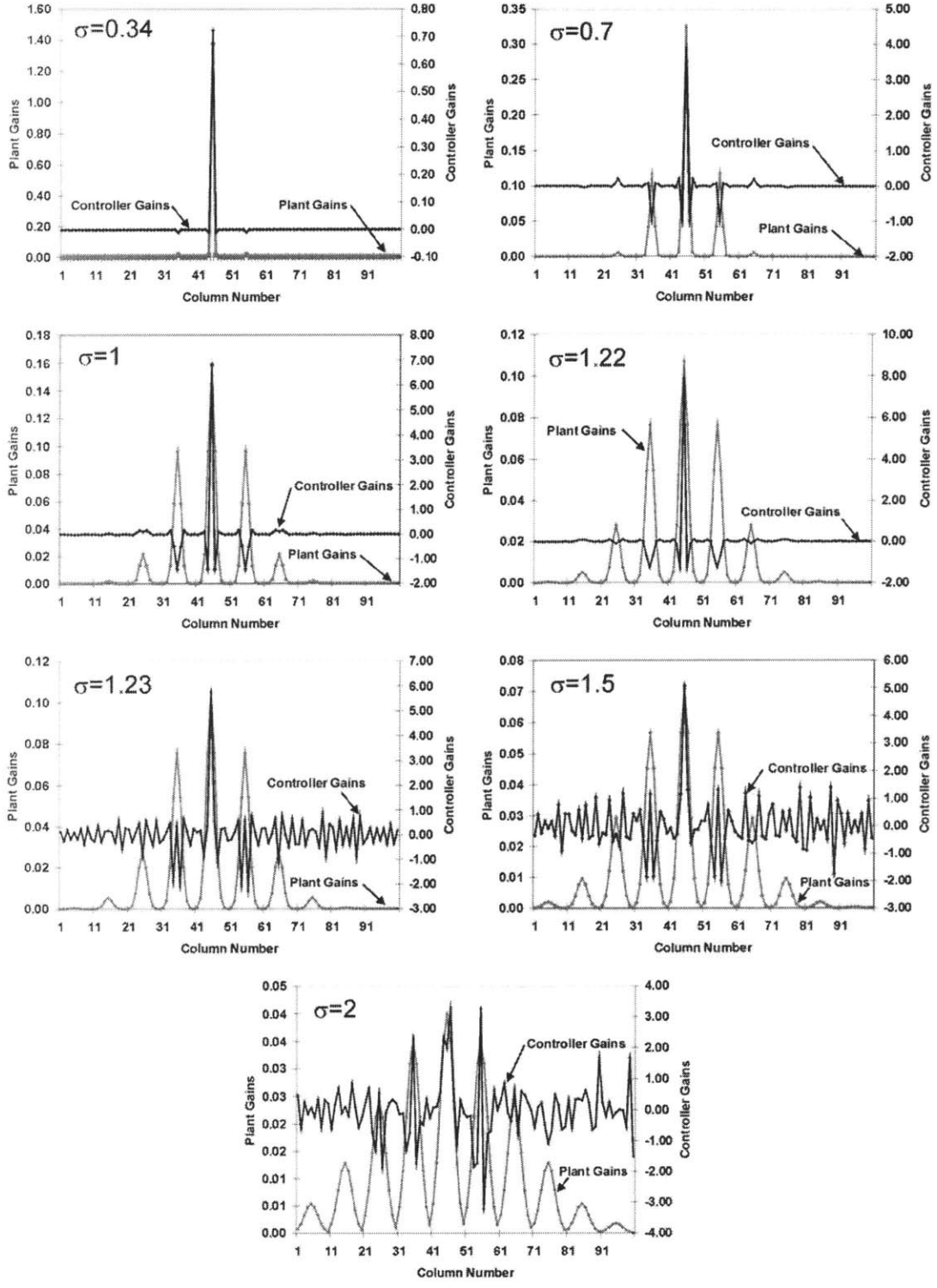


Figure C-1 Plant and LQR control gains for different values of the coupling spread σ. A *10x10* bundle of inputs (totaling 100) is assumed and the 45th input (occurring near the center of the bundle) is displayed. A relative weight of m=100 is chosen.

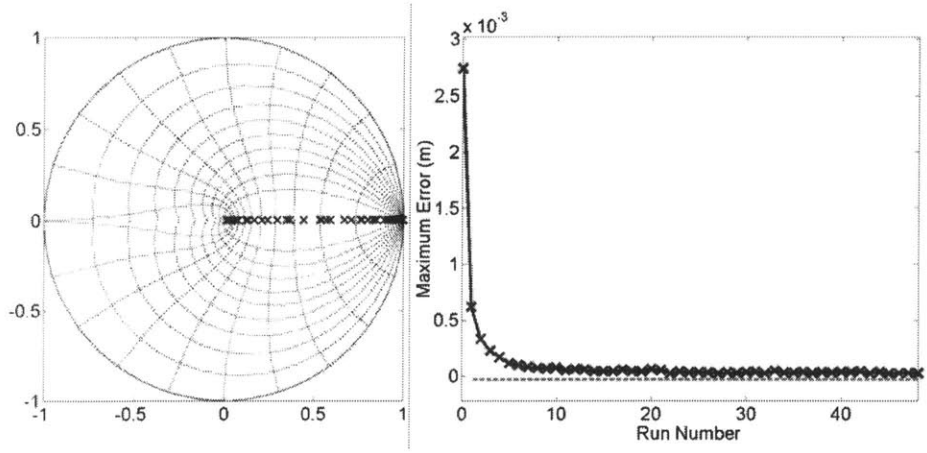


Figure C-2 Poles and maximum error for an LQR design controller. 10x10 bundle of inputs and a spread of σ=1.22 are assumed. A relative weight of m=100 is chosen.

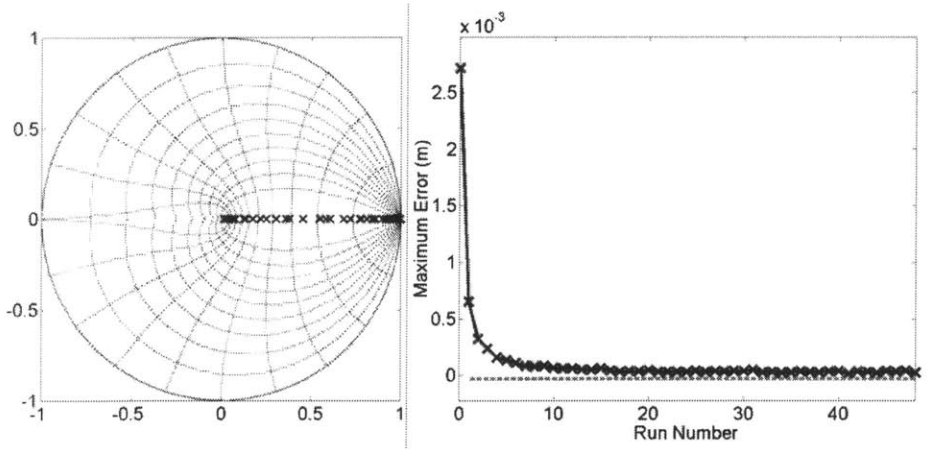


Figure C-3 Poles and maximum error for an LQR design controller. 10x10 bundle of inputs and a spread of σ=1.23 are assumed. A relative weight of m=100 is chosen.

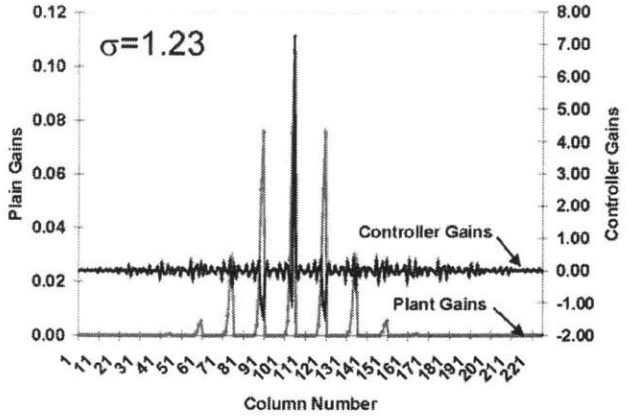


Figure C-4 Plant and LQR control gains. A *15x15* bundle of inputs (totaling 225) is assumed and the 105th row is displayed (equivalent center position of the 45th input presented in Figure C-1). A relative weight of m=100 is chosen.

APPENDIX

D

CHANGING THE PROCEDURE – STRETCH DISTANCE

Experimental results using a Linear Quadratic Regulator (LQR) procedure reveal a sharp change in curvature of the physically made, not simulated, part at the edges of the active die region. This change in curvature is traced to a computation for stretch distance. Stretch distance is defined as the distance the machine must move in order for the metal restraining jaws to become tangent to the prescribed curvature at the edge of the active region. The previously existing algorithm takes an 11 data point column (the edge of the 10 pin by 11 pin active region) and interpolates that data to 50,001 points using a cubic spline. The two rightmost points of the new, fine mesh are used to compute the tangent slope and, thus, the stretch distance. An alternative algorithm is suggested which does not interpolate the data from the 11 points which are originally specified. Taking two neighboring points, traveling left to right, it is possible to compute ten tangential slopes. It is proposed that the average stretch distance be used in the experimental procedure. The results of the existing and new algorithms are shown in Figure D-1 along with an average slope for the alternative stretch distance computation. The figure shows the two initial dies which are computed independently of the closed-loop method along with a

third die as computed by the SCA method (plot 3) and GIC method (plot 4). Note that the third die is the first true closed-loop die.

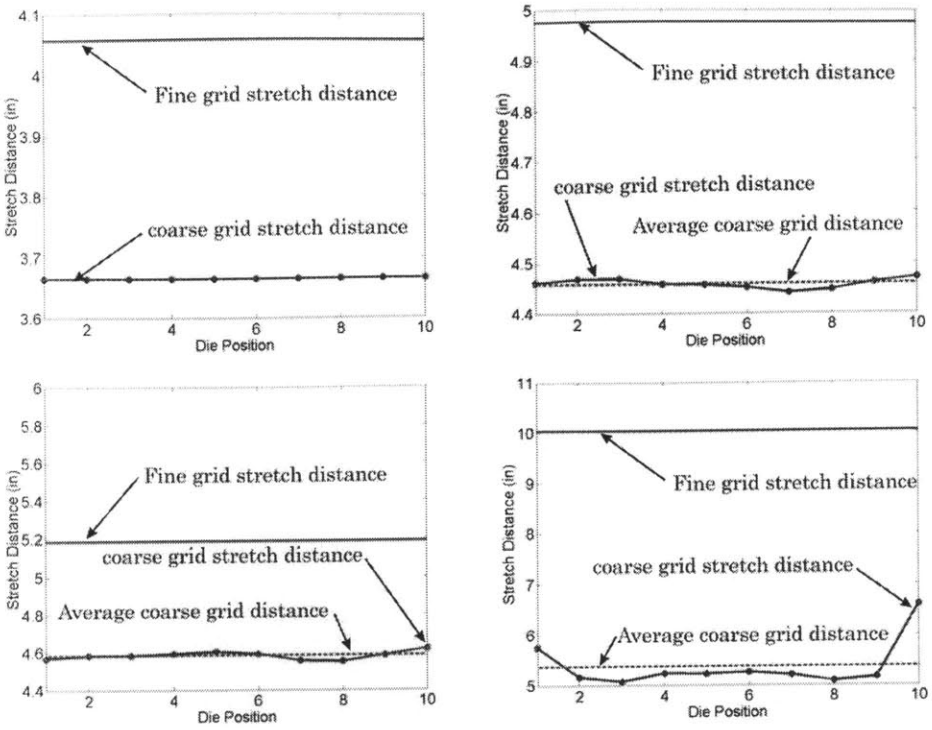


Figure D-1 Experimental process stretch distance computation. Included are results from a fine grid (computed from only the rightmost two points) and coarse grid (computed at each of the ten pin positions) estimation. (1) First die, (2) Second die, (3) Third die, SCA algorithm, (4) Third die, GIC algorithm.

Examination of all four plots of Figure D-1 reveals that the fine grid estimation of the stretch distance does not match any single coarse grid distance or the average coarse grid distance. However, the mismatch between the fine and coarse grid stretch distance for the first two dies and the third SCA algorithm die are much smaller than the mismatch for the third GIC algorithm die. The small size of the mismatch is characteristic of SCA algorithm computed dies. Note that for the GIC algorithm die, the stretch distance error is nearly a hundred percent of the average coarse grid distance.

The sharp change in curvature resulting from the overestimation of the stretch distance is part of the measured region and is sufficiently close to the active region to influence both the part registration routine and the computed part error. This phenomenon can be seen in Figure D-2 where the part error (reference minus part) for parts made with the fine grid suggested stretch distance (left) and coarse grid suggested stretch distance (right) is shown. Because of the significant influence of stretch distance, which is not part of the process modeled for control, it is recommended that its impact be made small and that the average coarse grid stretch distance be used.

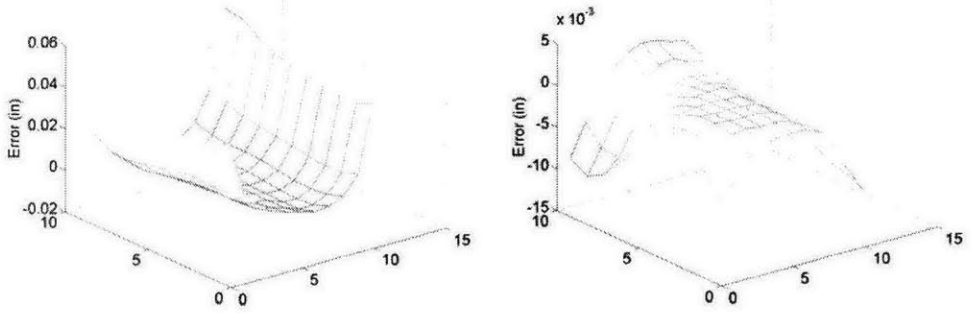


Figure D-2 Part Error for third die with fine grid suggested stretch distance (left) and coarse grid suggested stretch distance (right).

APPENDIX

E

EXPERIMENTAL DATA

LQR Controller, m=100						
Run	1	2	3	4	5	6
Mean Error (cm)	-5.32E-02	-4.41E-03	2.42E-04	-1.91E-03	9.29E-04	-1.06E-03
Maximum Error (cm)	-1.58E-01	-2.81E-02	-2.58E-02	-1.01E-02	-1.51E-02	1.26E-02
RMS Error (cm)	7.51E-02	7.99E-03	1.04E-02	4.43E-03	5.16E-03	4.01E-03

Table E-1 LQR controller (m=100) experimental data.

LQR Controller, m=1000						
Run	1	2	3	4	5	6
Mean Error (cm)	-5.32E-02	-4.41E-03	-5.63E-03	-8.00E-04	-4.55E-03	1.63E-02
Maximum Error (cm)	-1.58E-01	-2.81E-02	-2.67E-02	2.96E-02	-4.14E-02	8.14E-02
RMS Error (cm)	7.51E-02	7.99E-03	9.01E-03	7.06E-03	1.11E-02	2.88E-02

Table E-2 LQR controller (m=1000) experimental data.

LQG/H_2 Controller						
Run	1	2	3	4	5	6
Mean Error (cm)	-5.32E-02	-4.41E-03	8.78E-03	-2.28E-03	-1.56E-02	-1.48E-02
Maximum Error (cm)	-1.58E-01	-2.81E-02	3.51E-02	3.16E-02	-4.43E-02	-4.95E-02
RMS Error (cm)	7.51E-02	7.99E-03	1.41E-02	1.42E-02	2.14E-02	2.01E-02
7	8	9	10	11	12	13
-1.69E-05	1.15E-02	4.27E-03	-2.66E-03	-4.67E-03	-8.75E-03	-4.09E-03
-3.27E-02	3.62E-02	2.49E-02	-2.03E-02	-2.35E-02	-5.02E-02	-1.75E-02
1.06E-02	1.57E-02	8.18E-03	6.77E-03	7.50E-03	1.35E-02	6.33E-03
14	15	16	17	18	19	20
-4.06E-03	-1.77E-03	-3.33E-03	2.00E-04	-1.70E-03	-2.74E-03	-1.17E-03
-2.06E-02	-1.61E-02	-1.72E-02	-1.10E-02	-1.06E-02	-1.43E-02	-1.12E-02
7.24E-03	4.43E-03	5.26E-03	3.48E-03	3.69E-03	4.71E-03	3.59E-03

Table E-3 LQG/H2 controller experimental data.

H_∞ Controller						
Run	1	2	3	4	5	6
Mean Error (cm)	-5.32E-02	-4.41E-03	4.40E-03	-2.10E-03	-2.56E-03	5.52E-06
Maximum Error (cm)	-1.58E-01	-2.81E-02	1.86E-02	-1.20E-02	-1.90E-02	1.12E-02
RMS Error (cm)	7.51E-02	7.99E-03	7.88E-03	4.99E-03	6.11E-03	3.52E-03

Table E-4 H_∞ controller experimental data.

APPENDIX

F

MATLAB CODE: EXAMPLE 6-5

```
clear;


%*****************************************************************%
%plant definition
sigma_x=1.0; %x spread
sigma_y=1.0; %y spread
tol=0.0002; %tolerance for plant matrix entries;
rows=3; %number of input (pin) rows
columns=3; %number of input (pin) columns
raw_Gp = makegp(sigma_x, sigma_y, rows, columns,tol);
Kp_o=raw_Gp; %nominal plant matrix
%*****************************************************************%
clear raw_Gp; %memory savings

n=rows*columns; %this is the dimension of all inputs and outputs


%Plant Gain with Uncertainty (Kp=Kp_o+Kp_del*Delta)
Kp_del=0.1.*Kp_o; %uncertainty in each entry of Kp matrix

for i=1:n
   M12(n*(i-1)+1:n*i,:)=eye(n);%Keep this constant
   M21(i,n*(i-1)+1:n*i)=Kp_del(i,:); % Enter Dels into this matrix!!
end

%*****************************************************************%
%Here we see if any delta's are defined as 0 and eliminate them
del_count=n^2;
for i=n:-1:1; %These loops have to be done BACKWARDS count!!!
   for j=n:-1:1;
      if Kp_del(i,j)<1e-5
         M12(n*(i-1)+j,:)=[];
         M21(:,n*(i-1)+j)=[];
         del_count=del_count-1; %total cound of del's in the plant
matrix
      end
   end
end
%*****************************************************************%

%this is the Fu(M,Delta) structure
```

```
PLANT=[zeros(del_count) M12; M21 Kp_o];
clear Kp_del M12 M21; %memory savings

%Dynamics Block 1/(z-1)
A=eye(n);
B=eye(n);
C=eye(n);
D=zeros(n);
DYN=pck(A,B,C,D);
clear A B C D; %memeory

%Controller Block=> Kc
Kc=eye(size(Kp_o));
CONTROLLER=Kc;
clear Kc Kp_o; %memory savings


systemnames='PLANT DYN CONTROLLER';
sysoutname='G';
inputvar=['[w_del{',num2str(del_count),'};w_ref{',num2str(n),'}]'];
input_to_CONTROLLER=['[w_ref-
PLANT(',num2str(del_count+1),':',num2str(del_count+n),')]'];
input_to_DYN='[CONTROLLER]';
input_to_PLANT='[w_del;DYN]';
outputvar=['[PLANT(1:',num2str(del_count),');w_ref-
PLANT(',num2str(del_count+1),':',num2str(del_count+n),');PLANT(',num2st
r(del_count+1),':',num2str(del_count+n),')]'];
sysic

clear DYN PLANT CONTROLLER; %memeory savings

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Small Gain Theorem calculation
G22=sel(G,1:del_count,1:del_count);
out=dhfnorm(G22);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mu -Robust STABILITY Analysis
% 1. siso-smf has uncertainty between the second input and second
output
% 2. the analysis is run on only a part of the block structure that has
the
%    uncertainty!
n_omega=50; %frequency evaluation points!!
omega=linspace(0,2*pi*(n_omega-
1)/n_omega,n_omega);%[0:pi/100:2*pi*99/100];
G_g=frsp(G,omega,1);
G_uncert=sel(G_g,1:del_count,1:del_count);
deltaset=[-1.*ones(del_count,1),ones(del_count,1)]; %real value
(parametric) uncertainty
[bnds,rowd,sens,rowp,rowg] = mu(G_uncert,deltaset);
mu_high=pkvnorm(sel(bnds,1,1))
[pklow,omegapklow]=pkvnorm(sel(bnds,1,2));
```

```
pert=dypert(rowp,deltaset,bnds); %constructs the smallest destabilizing
perturbation

pertG=starp(pert,G);

disp(['G22 norm is between ',num2str(out(1)),' and ',num2str(out(2)),'
at ',num2str(out(3))]);
disp(['Mu norm is between ',num2str(pklow),' and ', num2str(mu_high),'
at ',num2str(omegapklow)]);
disp(['Improvement = ',num2str((out(1)-pklow)/pklow*100),' % ']);
disp(['Process Poles are']);
proc_poles = spoles(pertG)


figure(1)
clf
vplot(bnds); %plots the bounds
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX

G

MATLAB CODE: EXAMPLE 6-6

```
clear;
%*****************************************************************%
%plant definition
sigma_x=1.0; %x spread
sigma_y=1.0; %y spread
tol=0.0002; %tolerance for plant matrix entries;
rows=5; %number of input (pin) rows
columns=5; %number of input (pin) columns
raw_Gp = makegp(sigma_x, sigma_y, rows, columns,tol);
Kp_o=raw_Gp; %nominal plant matrix
%*****************************************************************%


n=rows*columns; %this is the dimension of all inputs and outputs


%Plant Gain with Uncertainty (Kp=Kp_o+Kp_del*Delta)
Kp_del=0.1.*Kp_o; %uncertainty in each entry of Kp matrix

M12=eye(n);
M21=Kp_del;

%this is the Fu(M,Delta) structure
PLANT=[zeros(n) M12; M21 Kp_o];

%Dynamics Block 1/(z-1)
A=eye(n);
B=eye(n);
C=eye(n);
D=zeros(n);
DYN=pck(A,B,C,D);

%Controller Block=> Kc
Kc=eye(size(Kp_o));
CONTROLLER=Kc;


systemnames='PLANT DYN CONTROLLER';
sysoutname='G';
inputvar=['[w_del{',num2str(n),'};w_ref{',num2str(n),'}]'];
input_to_CONTROLLER=['[w_ref-
PLANT(',num2str(n+1),':',num2str(n+n),')]'];
```

```
input_to_DYN='[CONTROLLER]';
input_to_PLANT='[w_del;DYN]';
outputvar=['[PLANT(1:',num2str(n),');w_ref-
PLANT(',num2str(n+1),':',num2str(n+n),');PLANT(',num2str(n+1),':',num2s
tr(n+n),')]'];
sysic


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Small Gain Theorem calculation
G22=sel(G,1:n,1:n);
out=dhfnorm(G22);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mu -Robust STABILITY Analysis
% 1. siso-smf has uncertainty between the second input and second
output
% 2. the analysis is run on only a part of the block structure that has
the
%    uncertainty!
n_omega=50; %frequency evaluation points!!
omega=linspace(0,2*pi*(n_omega-
1)/n_omega,n_omega);%[0:pi/100:2*pi*99/100];
G_g=frsp(G,omega,1);
G_uncert=sel(G_g,1:n,1:n);
deltaset=[-1.*ones(n,1),ones(n,1)]; %real value (parametric)
uncertainty
[bnds,rowd,sens,rowp,rowg] = mu(G_uncert,deltaset);
mu_high=pkvnorm(sel(bnds,1,1))
[pklow,omegapklow]=pkvnorm(sel(bnds,1,2));

pert=dypert(rowp,deltaset,bnds); %constructs the smallest destabilizing

pertG=starp(pert,G);

disp(['G22 norm is between ',num2str(out(1)),' and ',num2str(out(2)),'
at ',num2str(out(3))]);
disp(['Mu norm is between ',num2str(pklow),' and ', num2str(mu_high),'
at ',num2str(omegapklow)]);
disp(['Improvement = ',num2str((out(2)-pklow)/out(2)*100),' % ']);
disp(['Process Poles are']);
spoles(pertG)
```

APPENDIX

H

MATLAB CODE: SUBROUTINE MAKEGP.M

```
function raw_Gp = makegp(sigma_x, sigma_y, rows, columns,tol);

%this program assembles a prospective Gp.
pi = 3.1416;

raw_Gp = zeros(rows*columns,rows*columns);

die_number = linspace(1,rows*columns,rows*columns);
die_number_map = reshape(die_number,rows,columns);

for i=1:(rows*columns);
   %this locates the pin under consideration
   [pin_row,pin_column]=find(die_number_map==i);
   row_number =0;
   for j=1:columns;
      for k=1:rows;
         row_number=row_number+1;
         %since (row column give a uniform grid position, we can
         % use them to determine the "distance" between pins and
         % points on the surface
         %this gets the distance from pin to point on surface
         %distance = sqrt( (xpos(pin_row,pin_column)-
xpos(k,j))^2+(ypos(pin_row,pin_column)-ypos(k,j))^2 );
         distance_x = sqrt( (pin_column-j)^2 );
         distance_y = sqrt( (pin_row-k)^2 );

         %below is the Gaussian response
         %mean is centered at the pin
         response = 1/(2*pi*sigma_x*sigma_y)*exp(-0.5*(
(distance_x/sigma_x)^2+(distance_y/sigma_y)^2) );
         if (distance_x > 3*sigma_x) & (distance_y > 3*sigma_y)
            response =0;
         end

         if response < tol
            response=0;
         end

         raw_Gp(row_number,i) = response;
      end
   end
end
```