

Algorithms for Closed Loop Shape Control

by

Walton A. Norfleet

B.S. Mechanical Engineering, 1995

Purdue University

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE in Mechanical Engineering

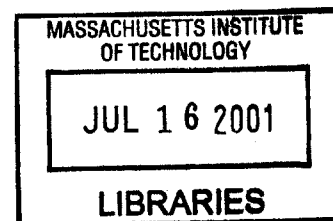
at the

Massachusetts Institute of Technology

June 2001

BARKER

© 2001 Massachusetts Institute of Technology
All rights reserved



Signature of Author.....

Department of Mechanical Engineering
May, 2001

Certified by.....

Professor David E. Hardt
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by.....

Ain A. Sonin
Chairman, Mechanical Engineering Committee on Graduate Students

Algorithms for Closed Loop Shape Control

by

Walton A. Norfleet
B.S. Mechanical Engineering, 1995
Purdue University

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degree of
MASTER OF SCIENCE in Mechanical Engineering

ABSTRACT

The stretch forming process is used to make structural sheet metal parts in the aerospace industry. The development of stretch forming tools has long been plagued by significant challenges. First, the low production volumes within the aerospace industry and the large numbers of stretch formed parts make the process capital intensive. Second, the development of stretch forming tooling has long been more of an art than a science. This results in poorly designed tools, poor quality parts, and lengthy tooling development cycles.

A stretch forming tool capable of rapid reconfiguration was previously designed to address these issues. This tool is used in conjunction with a self-tuning shape control algorithm, which guides the die to the correct shape. There have been many simulations, and lab scale successes with these algorithms, but production scale implementations have experienced difficulties. These problems are related to the method of system identification and process variation. To better understand these issues, analysis and simulation are performed on the various forms of the algorithm. These investigations led to a greater understanding of the algorithms and the synthesis of an improved algorithm.

In conclusion, a greater understanding of previously developed algorithms is presented. The system identification is mapped as a Point Spread Function applied through a cyclic convolution. This view provides insight into how the system identification is applied and allows system coupling to be quantified. Furthermore, through improved understanding a new algorithm is synthesized. This new algorithm offers an implementable solution that is optimized for performance, robustness to variation, and ease of use.

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

Acknowledgments

I thank my advisor, Professor David Hardt for his contributions to my graduate school experience. Much to his credit, I have thoroughly enjoyed the past two years at the Massachusetts Institute of Technology. In particular, I am thankful to him for providing answers to many of my questions, and more importantly for urging me to provide answers as well.

I thank John Papazian and all of his colleagues at Northrop Grumman for the opportunity to contribute to this project. I am proud to have been involved with this program and have enjoyed the camaraderie that typified all of our meetings.

I am extremely thankful for the support provided by Northrop Grumman to my research under the 'Reconfigurable Tooling for Sheet Metal Forming' contract (110B24019AK).

I thank Dr. Simona Socrate for her assistance.

I would like to thank all of the friends I have made along the way.

And I would like to thank my wife Kate, for her support.

Chapter 1 : Introduction	13
Research Background.....	13
Flexible Sheet Forming System	14
Self-Tuning Shape Control Algorithm.....	16
Contribution of This Research	16
Goals.....	16
Thesis Summary	17
Conclusions	18
 Chapter 2 : Stretch Forming Overview	19
Basic stretch forming mechanics.....	19
Surface Geometry Classifications	28
Summary	32
 Chapter 3 : Research Background.....	33
Issue # 1: Excessive Capital Costs for Stretch Forming	33
Issue # 2: Difficulty in Predicting Correct Die Shape.....	36
Issue # 3: Excessive Shape Variation in Stretch Forming	38
Summary	40
 Chapter 4 : The Generic Algorithm Structure.....	41
Stretch forming process model.....	41
Algorithm Model.....	42
Schematic Process Model.....	46
Control system view.....	48
Steady state error	49
Response time	50
Disturbance Rejection	53
Common Implementation Elements.....	55
Geometry Parameters	55
Surface Fitting.....	56
Part Registration.....	57
Error Measurements	58
Summary	61
 Chapter 5 : Discussion of Algorithm #1	63
Basic Structure	63
Implementation.....	64
Historical Experimental Results.....	64
Algorithm Concerns	65
System Coupling	65
System Identification.....	66
Summary	67
 Chapter 6 : Discussion of Algorithm #2	69

Basic Structure	69
Linear Systems Background.....	73
Implementation.....	92
Historical Experimental Results.....	95
Algorithm Concerns	97
Superposition.....	98
System Identification.....	98
Summary	100
Chapter 7 : Presentation and Discussion of Algorithm #3	103
Basic Structure	103
Implementation.....	105
System Identification.....	105
Part “Referencing”	106
Algorithm Concerns	115
Variation Treated Differently Across the Die	115
Coupling Ignored.....	116
System Identification Ambiguity	116
Summary	116
Chapter 8 : Simulation Methods.....	117
Full Part Forming Simulation.....	117
Accuracy of Full Part forming simulation.....	123
Simulating the Effects of Variation on FEA Formed Components	124
Summary	127
Chapter 9 : Simulation Results.....	129
Comparison with Experimental Results.....	129
Algorithm Comparisons with Full Forming Simulations.....	132
Variation Investigations, Noise included in Identification.....	135
Variation Investigations, Noise Excluded from Identification.....	139
Summary	140
Chapter 10 : Conclusions - Guidelines for Future Work	143
Conclusions	143
Future Work	145
References	147

Figure 1-1: Applications of stretch formed aerospace components	13
Figure 1-2: Flexible sheet forming system.....	15
Figure 1-3: Shape control algorithm differences.....	17
Figure 2-1: Cyril Bath stretch forming machine in process	20
Figure 2-2: Schematic of stretch forming process	21
Figure 2-3: Description of sheet geometry.....	22
Figure 2-4: Strains & Stresses after pre-stretch phase	22
Figure 2-5: Power law, strain hardening curve for Aluminum 2024-O	23
Figure 2-6: Representation of sheet bending.....	24
Figure 2-7: Strains & Stresses after bending phase	25
Figure 2-8: Strains & Stresses after post stretch phase	25
Figure 2-9: Strain and Stress state after sheet release	26
Figure 2-10: Springback of sheet on a cylindrical die	27
Figure 2-11: Representation of surface curvatures	28
Figure 2-12: Shape classifications based on mean & Gaussian curvature	30
Figure 3-1: Typical rigid stretch forming tool	33
Figure 3-2: Forming surface of MIT, lab scale, re-configurable tool [Valjavec, 1999] ...	34
Figure 3-3: Forming surface of Northrop Grumman / MIT, production scale, re- configurable tool	34
Figure 3-4: Interface between die, interpolator and sheet metal	35
Figure 3-5: Rework Procedure	36
Figure 3-6: sample output of ABAQUS FEA	37
Figure 3-7: Variation levels associated with different process control strategies.....	38
Figure 3-8: Stress – Strain curve sensitivities	40
Figure 4-1: Stretch forming process model.....	41
Figure 4-2: Some stretch forming process parameters.....	41
Figure 4-3: Stretch forming process model with feedback	42
Figure 4-4: Block diagram of generic algorithm structure.....	43
Figure 4-5: Schematic model of algorithm	47
Figure 4-6: Control system view of algorithm	48
Figure 4-7: Root locus of die configuration algorithm	51
Figure 4-8: discrete, system responses	52
Figure 4-9: Control schematic for evaluating disturbance effects	53
Figure 4-10: Part and Die shape parameters	56
Figure 4-11: Lab Scale Coordinate Measuring Machine	57
Figure 4-12: Reference mark used with MIT stretch forming press	58
Figure 4-13: Error surface	59
Figure 4-14: Altering the average error ($\bar{\epsilon}$) value by translating along the z-axis.....	60
Figure 5-1: Algorithm #1 control schematic	63
Figure 5-2: Algorithm #1 results from Webb [1987, p. 59] on match die tooling	65
Figure 5-3: Depiction of system coupling.....	66
Figure 5-4: graphical interpretation of system identification.....	67
Figure 6-1: Algorithm #2 control system schematic (Note: all variables are Fourier transforms of particular part or die shapes)	70
Figure 6-2: Linear, continuous-time system	73

Figure 6-3: System input	73
Figure 6-4: Discrete time system	74
Figure 6-5: graphical depiction of convolution.....	77
Figure 6-6: Assumption of periodicity.....	78
Figure 6-7: Convolution of a cyclic input	79
Figure 6-8: Cyclic convolution	80
Figure 6-9: Causal and A-Causal Systems.....	81
Figure 6-10: Centering a 1D Impulse response for easier viewing.....	81
Figure 6-11: Two-dimensional, cyclic, spatial system.....	82
Figure 6-12: Shifting of the 2-D, periodic PSF to assist viewing	84
Figure 6-13: Schematic of 2-D, PSF shifting.....	84
Figure 6-14: PSF functions displaying coupling.....	85
Figure 6-15: Linear system computation methods.....	87
Figure 6-16: Convolution & De-convolution.....	88
Figure 6-17: De-convolution & convolution.....	89
Figure 6-18: Truncated system responses	91
Figure 6-19: Windowing Function	92
Figure 6-20: Windowed Functions.....	92
Figure 6-21: Common Parameters for Valjavec's [1999] tests of algorithm #2.....	95
Figure 6-22: Reference shapes for Valjavec's [1999] experiments	96
Figure 6-23: Valjavec's [1999] experimental results.....	97
Figure 6-24: system identification performed with calibration die shapes far apart.....	99
Figure 6-25: System identification performed with close calibration dies and noise	100
Figure 7-1: Shape control algorithms.....	103
Figure 7-2: Algorithm #3 control schematic (Note: all variables are spatial coordinates)	104
Figure 7-3: Offset z-axis coordinates.....	106
Figure 7-4: Stretch forming die with and without added constant ' α '	107
Figure 7-5: Sensitivity near 'zero'	111
Figure 7-6: Production scale example of sensitivity near zero	112
Figure 7-7: Heuristic for Algorithm #3	113
Figure 7-8: Reducing sensitivity near 'zero'	114
Figure 8-1: MIT, lab scale stretch press.....	117
Figure 8-2: Full forming simulation process flow	118
Figure 8-3: Pin position and die interaction positions.....	119
Figure 8-4: Equivalent, rigid die surface.....	119
Figure 8-5: Representation of material blank and die in ABAQUS	120
Figure 8-6: Boundary conditions.....	121
Figure 8-7: Explanation of softened contact	122
Figure 8-8: Dies and Parts for simulation shortcut	125
Figure 8-9: Introducing additive noise.....	126
Figure 8-10: Introduction of additive noise to reference part	127
Figure 9-1: Equations used to define reference parts and initial dies	130
Figure 9-2: Replication of Valjavec's [1999] cylinder experiment	130
Figure 9-3: Replication of Valjavec's [1999] saddle experiment	131
Figure 9-4: Replication of Valjavec's [1999] spheroid experiment.....	131

Figure 9-5: Summary of full simulation results	133
Figure 9-6: Simulations of cylinder forming trial	133
Figure 9-7: Simulations of saddle forming trials	134
Figure 9-8: Simulations of spherical forming trials	134
Figure 9-9: Variation amplification, noise included in identification	136
Figure 9-10: Error convergence during variation investigation simulation, noise included in identification	137
Figure 9-11: Die errors for algorithm #2 (no window) and algorithm #3	138
Figure 9-12: Variation amplification, noise excluded from identification.....	139
Figure 9-13: Error convergence during variation investigation simulation, noise excluded from identification.....	140
Figure 10-1: Algorithm Framework.....	143

Chapter 1 : Introduction

Stretch forming is a manufacturing process in which metal sheets or extrusions are wrapped around a die to form parts into simple or complex shapes of mild curvature. This technique is often employed within the aerospace industry to build structural skin components including nacelles, pressure domes, and wing leading edges, which are shown in Figure 1-1. Aluminum is the most common material used for such applications, however other metals are also used including steel and titanium.



Figure 1-1: Applications of stretch formed aerospace components

Research Background

The stretch forming process used to make structural sheet metal parts in the aerospace industry and the development of stretch forming tools have long been plagued three significant challenges. First, the low production volumes within the aerospace industry and the large numbers of stretch formed parts make the process extremely capital intensive. Second, the development of a stretch forming tooling has long been more of an art than a science. This often results in poorly designed tools and poor quality parts, or lengthy, iterative tooling development cycles. Third, poor stretch forming process control in most aerospace manufacturing settings causes excessive variation in the stretch formed components.

The large numbers of tools found within a stretch forming manufacturing process are costly to develop and often require large warehouses for storage. As a result, time and labor associated with changing from one tool to another are also excessive. Furthermore, the designers of stretch forming tools have historically approached tool design by making their best guess at the die shape that correctly compensates for material springback and then improving the die iteratively through difficult rework. This approach is expensive, time consuming, and may result in poor quality parts. In recent years, some finite element analysis techniques (FEA) have aided engineers and designers in choosing appropriate die shapes. However, such techniques require excessive engineering time. These techniques also require engineers to make accurate assumptions regarding material properties and process parameters.

The conventional methods used to control the stretch forming process are flawed with excessive variation. Generally, the trajectory of the stretch forming process is controlled by monitoring hydraulic cylinder pressure, displacement of the hydraulic cylinders, or even through the machine operator's rough, visual estimates of the process. The inherent variation associated with these control methods exacerbates the variation seen in parts formed under such control schemes.

Flexible Sheet Forming System

To address these issues, researchers at Massachusetts Institute of Technology (MIT) and Northrop Grumman have developed a flexible manufacturing system that consists of a reconfigurable tool and a self-tuning shape control algorithm. An overview of this system is shown in Figure 1-2.

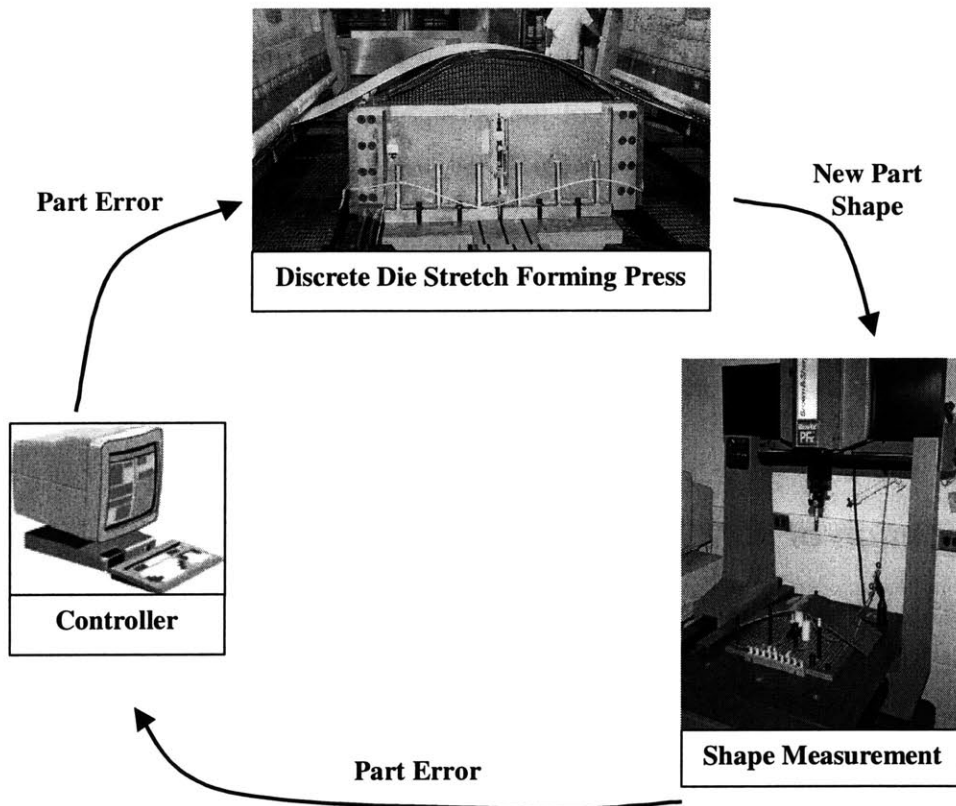
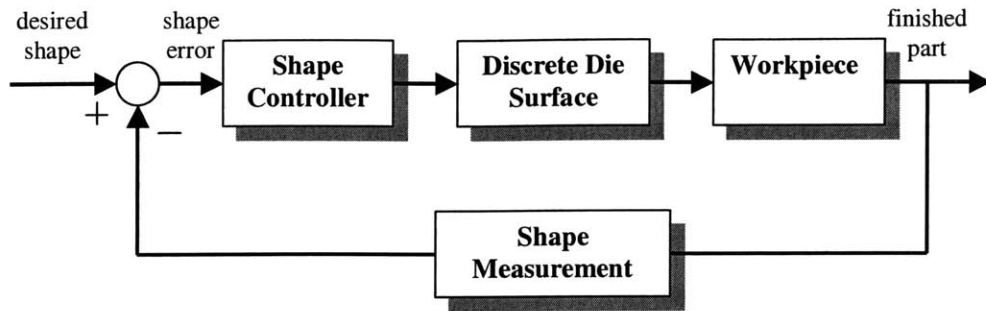


Figure 1-2: Flexible sheet forming system

The reconfigurable tool can rapidly change shape. It can thus replace the majority of the fixed structure dies that exist within the manufacturing process. The reconfigurable tool also improves the die development process by allowing changes to dies shape to be made in-process. A self-tuning shape control algorithm is used to predict a die shape that is correctly compensated for springback.

Self-Tuning Shape Control Algorithm

Over the past years, several different shape control algorithms have been used in different incarnations of the flexible forming system. The first algorithm (called algorithm #1 in this thesis) was implemented by Webb [1981] on a lab scale stretch forming system. This algorithm produced positive results but did not converge quickly enough. It did not have a means to model coupling nor did it use system identification. A later version of the algorithm (called algorithm #2 here) was implemented by Webb [1987] on an axis-symmetric metal stamping operation. Algorithm #2 added coupling to the system by using the Fourier transform of the shape data. It also contained additional system steps that identified nature of the nature of the springback that occurs during formation and then used this information to improve its predictions. Algorithm #2 was also implemented by Osterhout [1991] on a matched die sheet forming system, Valjavec [1999] on a stretch forming system and more recently by Northrop Grumman on production scale stretch forming equipment. This algorithm improved greatly on the performance of algorithm #1, but at the cost of increased complexity and sensitivity to noise.

Contribution of This Research

Goals

The work in this thesis addresses questions that have been asked about algorithms #1 and #2. These question include the following:

- How does algorithm #2 work?
- Can it be understood more fundamentally?
- Can it be improved?

The need to answer these questions has recently been emphasized by issues raised from recent productions scale implementations of the flexible sheet forming system at Northrop Grumman. Initial forming trials have uncovered the following problems and corresponding questions.

- System identification takes two iterations of the forming system. Is it necessary?
- Algorithm #2 is not robust to variability. Can its robustness be improved?
- The quality of the system identification is difficult to evaluate. Can it be understood better?

Thesis Summary

Investigations performed through analysis and simulation have led to a greater understanding of the mathematical mechanisms associated with algorithm #2. The system identification of algorithm #2 is now understood to be an estimate of the system Point Spread Function (PSF), which is applied through a two-dimensional, cyclic convolution. This perspective quantifies the coupling inherent in algorithm #2. It also makes measuring the quality of the system identification much easier.

Investigations of algorithm #1 and #2 led to the development of a new form of the algorithm that uses system identification without the convolution interval to introduce system coupling. The new ‘algorithm #3’ is also charted with algorithms #1 and #2 in Figure 1-3.

No Coupling	<u>Algorithm #1</u> - No System ID - No Coupling	<u>Algorithm #3</u> - System ID - No Coupling
Coupling		<u>Algorithm #2</u> “Deformation Transfer Function” - System ID - Coupling via Convolution
	No Identification	Identification

Figure 1-3: Shape control algorithm differences

Several useful simulations were developed to allow deterministic and controlled stochastic investigations of the algorithms. These simulations have been archived for use by future researchers.

The above mentioned simulations were also used to rate the performance of algorithms #1, #2, and #3, with respect to the following criteria.

- Ease of implementation and use
- Part error reduction rate
- Robustness to process variation

Conclusions

Through new perspectives of previously developed algorithms, an algorithm has been developed with performance comparable to algorithm #2. However, this algorithm is however more robust to variation and simpler to implement.

A greater understanding of the previously developed algorithms has also been attained through this new perspective. This understanding makes troubleshooting the prior algorithms much easier.

Chapter 2 : Stretch Forming Overview

In this chapter, an overview of the stretch forming process is presented. First, the machinery used to stretch form components and the steps of the stretch forming process are defined. Next, the mechanics of sheet metal deformation are analyzed for bending and stretching a sheet about a die of constant curvature. Finally, shape classifications are defined to help understand the limitation of such analysis techniques.

Basic stretch forming mechanics

The stretch forming of sheet metal parts involves imposing both bending and stretching strains to achieve the desired part shapes and to strain-harden the material. The stretching strains minimize the curvature relaxation, or shape springback, that occurs after the deformation process. They also insure that regions of compression will not buckle during the forming process. The stretch forming process is comprised of several stages, including the pre-stretch phase, the forming phase and the post-stretch phase. It should be noted that either the pre-stretch or post-stretch phase may be omitted from the process, or they may be used in conjunction with one another. The process is called “drape forming” when the pre-stretch phase is omitted. The objective of this process is to achieve a part of a desired contour. As with any deformation process, the challenge is to compensate for the curvature springback of the material and to do so consistently, even when material and machine variations occur.

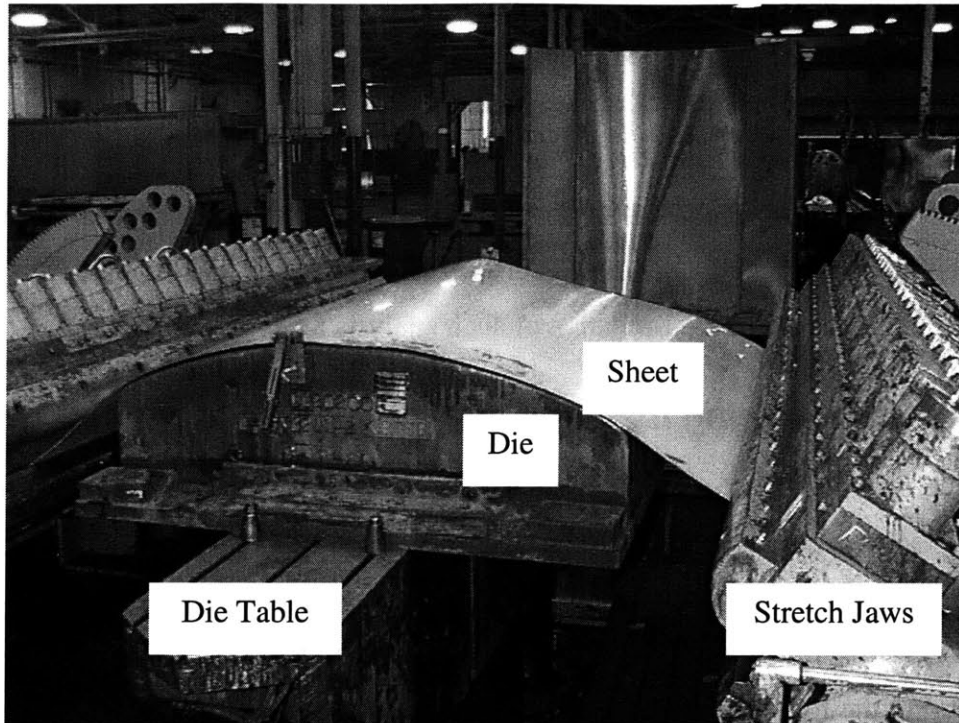


Figure 2-1: Cyril Bath stretch forming machine in process

A stretch forming press is shown in the process of manufacturing an aerospace skin in Figure 2-1. The schematics shown in Figure 2-2 provide a depiction of the phases of the stretch forming process. After the sheet metal blank has been loaded into the machine, an initial strain (ϵ_{Pre}), is imparted to the material in the pre-stretch phase. As previously noted, the trajectory of the machine and thus the level of strain in the part can be controlled by one of many different methods. In practice, stretch cylinder forces or cylinder displacements are often the control variables. Next, the die is pressed into the sheet metal during the wrap phase of the process. Generally, the net level of strain in the sheet is kept constant during this phase, although it does not need to be. During the next phase, an additional strain (ϵ_{post}) is induced into the sheet metal by extending the stretch cylinders further.

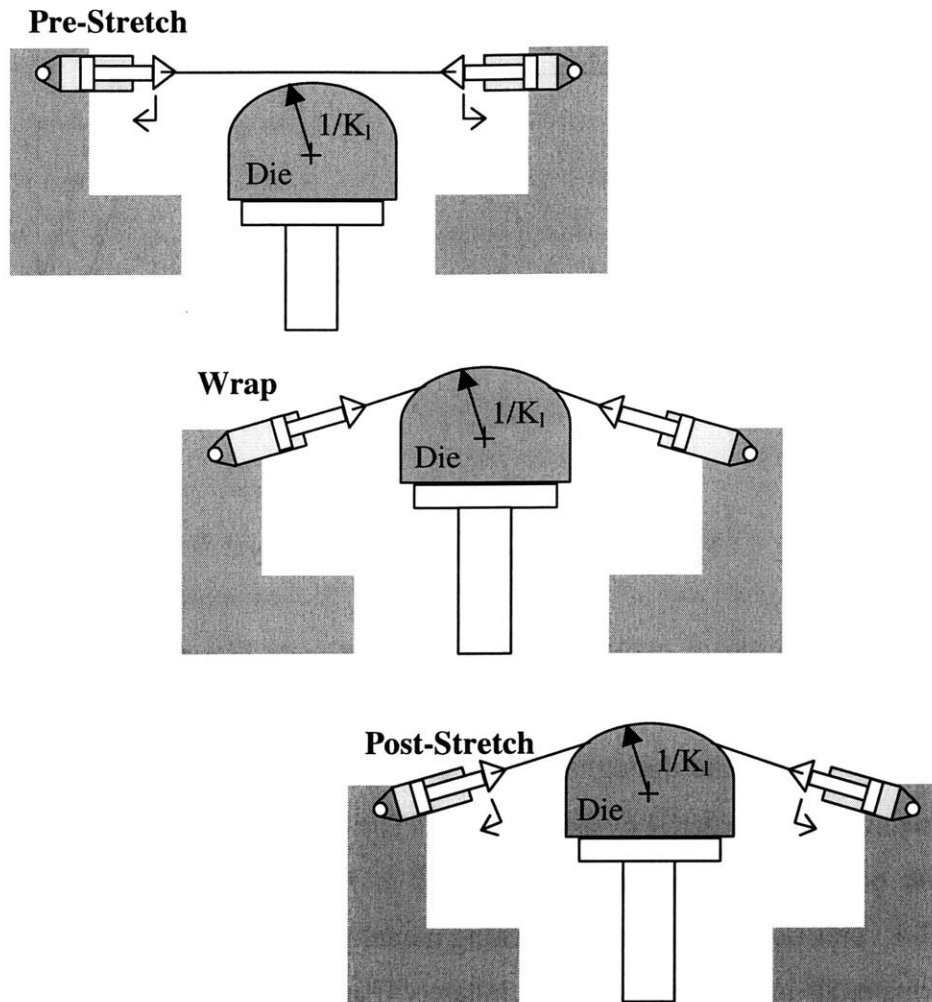


Figure 2-2: Schematic of stretch forming process

A simplified, two-dimensional analysis of constant radius stretch forming is helpful in understanding the mechanics of the forming process. Figure 2-3 details the variables used to describe the geometry of the sheet metal in this analysis.

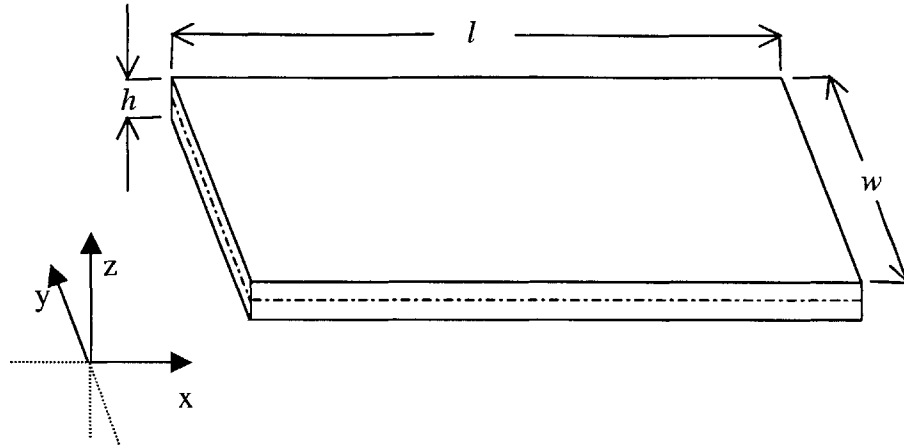


Figure 2-3: Description of sheet geometry

During the pre-stretch phase, the sheet is stretched uniformly in a direction parallel with the length (l) dimension until the desired amount of pre-strain (ϵ_{pre}) is achieved. Ignoring edge effects, the strain state throughout the material thickness ($\epsilon(z)$) is defined through any cross section of the material by Equation 2-1. The corresponding stress state is found through the power law strain-hardening model in Equation 2-2, Equation 2-3, and Equation 2-4. This power law strain-hardening model provides an accurate model for the aluminum materials used in the aerospace industry [Parris, 1996]. Its relationship is also shown graphically in Figure 2-5. Figure 2-4 plots the strain and stress states through a cross section of the sheet metal during the pre-stretch phase.

$$\epsilon(z) = \epsilon_{pre}$$

Equation 2-1

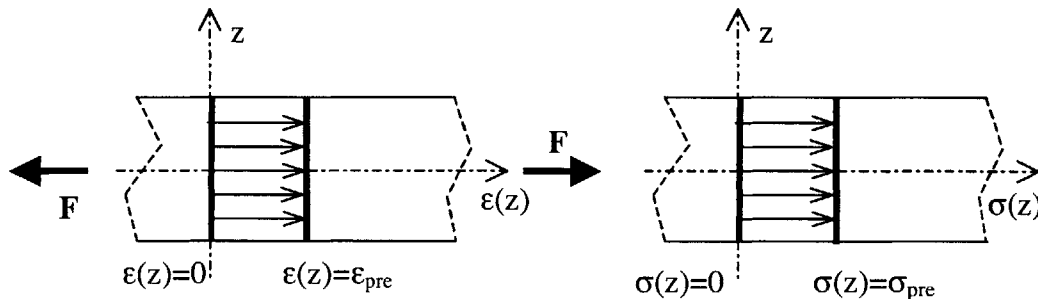


Figure 2-4: Strains & Stresses after pre-stretch phase

$$\sigma = E\varepsilon \quad \varepsilon < \varepsilon_{yield}$$

Equation 2-2

Where:

σ = stress state
 ε = strain state
 ε_{yield} = yield strain

$$\sigma = K\varepsilon^n \quad \varepsilon > \varepsilon_{yield}$$

Equation 2-3

Where:

n = strain hardening coefficient
 K = strength coefficient

$$K = E^n \sigma_{yield}^{1-n}$$

Equation 2-4

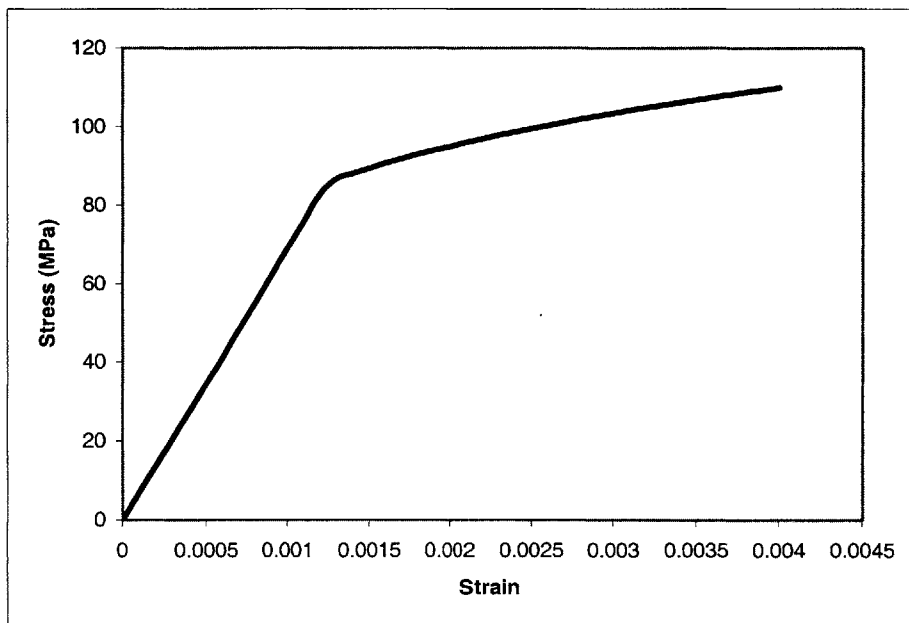


Figure 2-5: Power law, strain hardening curve for Aluminum 2024-O

Next, bending strains are imposed throughout the material thickness by bringing the die into contact with the sheet. The resulting strain distribution is defined by Equation 2-5 and is dependent solely on die and sheet geometry if the sheet is brought into proper contact with the die as shown in Figure 2-6. For this analysis, we are assuming a cylindrical die of constant curvature (K_l), which is equivalent to the inverse of the die radius ($1/R_l$). The subscript 'l' is associated with the die as it represents the shape of the part when it is loaded against the die. The vertical distance from the midline of the sheet, is 'z' as shown in Figure 2-3. The power law strain-hardening model defined by Equation 2-2 and Equation 2-3 are again used to define the corresponding stress state¹.

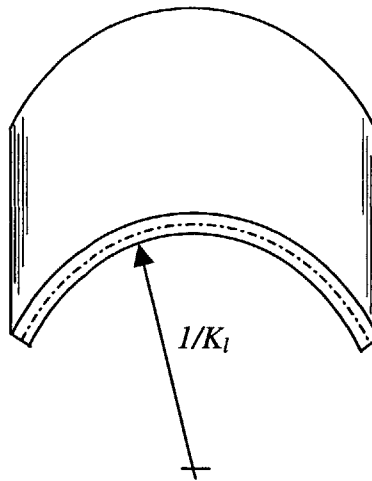


Figure 2-6: Representation of sheet bending

$$\varepsilon(z) = \varepsilon_{pre} + K_l z$$

Equation 2-5

¹ A work hardening model is also needed in cases where the combined level of pre-stress and die curvature allow plastic, compressive stresses to occur.

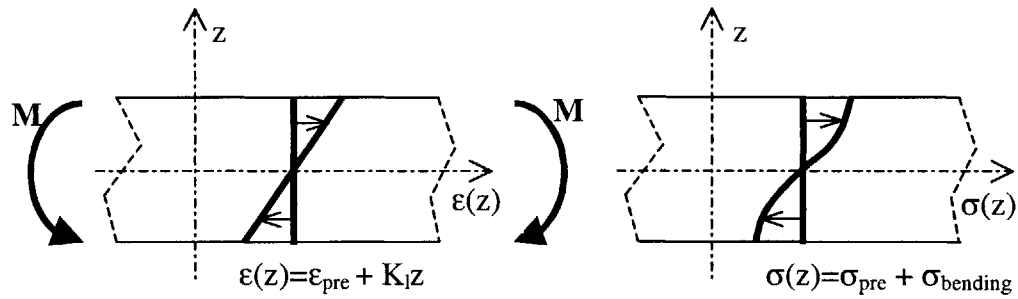


Figure 2-7: Strains & Stresses after bending phase

Next, the stretch cylinders impose additional, uniform tensile strain (ϵ_{post}) throughout the cross section by displacing their jaws further. This leaves the sheet in the final strain state, defined by Equation 2-6 and shown schematically in Figure 2-8.

$$\epsilon(z) = \epsilon_{pre} + K_1 z + \epsilon_{post}$$

Equation 2-6

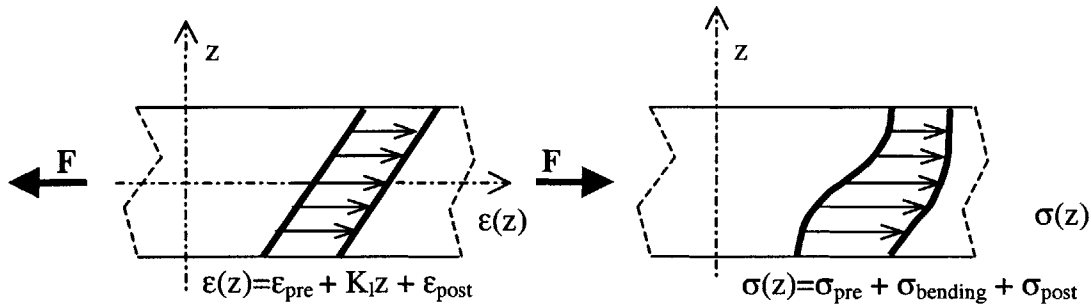


Figure 2-8: Strains & Stresses after post stretch phase

The final loaded axial force in the sheet can be found through Equation 2-7 and the final loaded moment about the mid-plane of the sheet is found through Equation 2-8. Once the part has been released the sheet will “springback” to an equilibrium point where the axial force and the moment about the mid-plane are zero. Figure 2-9 depicts these changes as stress levels reduce until their net value is zero and their net moment is zero. This is also represented in the strain schematic of Figure 2-9 as the strain line moves to the left, which suggests axial relaxation. The strain line also rotates closer to vertical, which indicates a relaxation in curvature.

$$F_l = \int_{-h/2}^{h/2} \sigma(z) w dz$$

Equation 2-7

$$M_l = \int_{-h/2}^{h/2} \sigma(z) z w dz$$

Equation 2-8

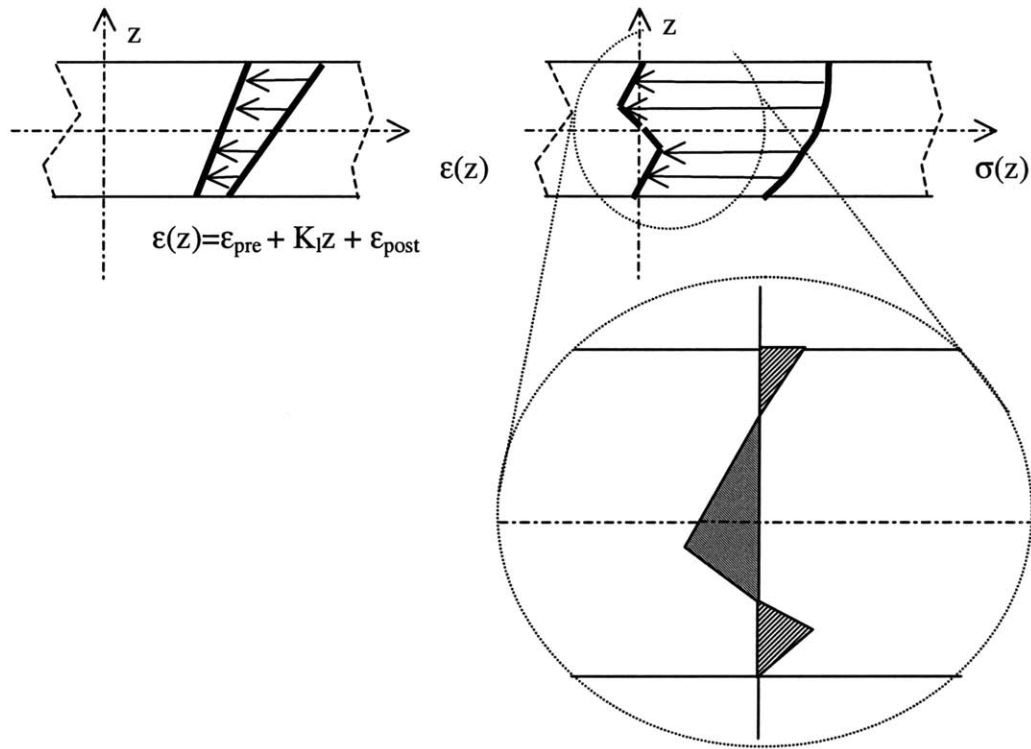


Figure 2-9: Strain and Stress state after sheet release

It is this curvature relaxation that is termed springback. It represents a shape difference between the curvature associated with a sheet loaded onto the die (K_l) and the curvature of the sheet when it is unloaded (K_u) as is shown in Figure 2-10. The springback ratio (ΔK) is the relationship between the loaded and unloaded curvature and is defined by Equation 2-9. The springback ratio will take on values between zero and unity. A zero

value indicates that there is no springback after the part is released, while a value of unity means the sheet springs back to its original shape after the forming process.

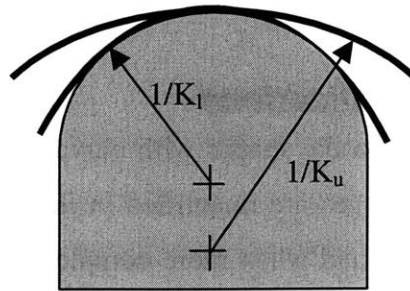


Figure 2-10: Springback of sheet on a cylindrical die

$$\Delta K = 1 - \frac{K_u}{K_l}$$

Equation 2-9

The overview of the constant radius, cylindrical die scenario emphasizes the key concept of stretch forming: the idea that springback can be minimized by the introduction of additional strains. These additional strains serve to create a more consistent level of stress throughout the cross section of the material. This in turn minimizes the moment about the mid-plane of the sheet before the sheet is released and therefore minimizes the shape springback.

Analytic methods to approximate the springback ratio for simple shapes, including developable arcs of constant curvature, are relatively straightforward to prove and use. Equation 2-10, developed by Parris [1996] is the result of such an analysis for the springback of constant curvature arcs following the power law strain-hardening model.

$$\Delta K = \frac{3}{K_l (h/2)^3} \left[\frac{K_l}{3} \left(\frac{\sigma_{yield}}{EK_l} \right)^3 + \frac{K_l}{E(N+2)} \left(\frac{h}{2} \right)^{n+2} - \left(\frac{\sigma_{yield}}{EK_l} \right)^{n+2} \right]$$

Equation 2-10

Surface Geometry Classifications

The prior analysis applies only to die shapes with curvature along only one dimension. Classifications of geometry changes are quantified in this section to define when such analysis techniques can be used and when more complex methods are required. These classifications may also be used by further research proposed in this report.

Consider the surface normal vector on the point of any surface as shown in Figure 2-11. Two orthogonal surface curves can be oriented such that they contain the maximum and minimum surface curvatures at that point. These curvatures are known as the principal curvatures (K_{min} and K_{max}). It is noted that ‘concave down’ curvatures are considered positive curvatures throughout this research.

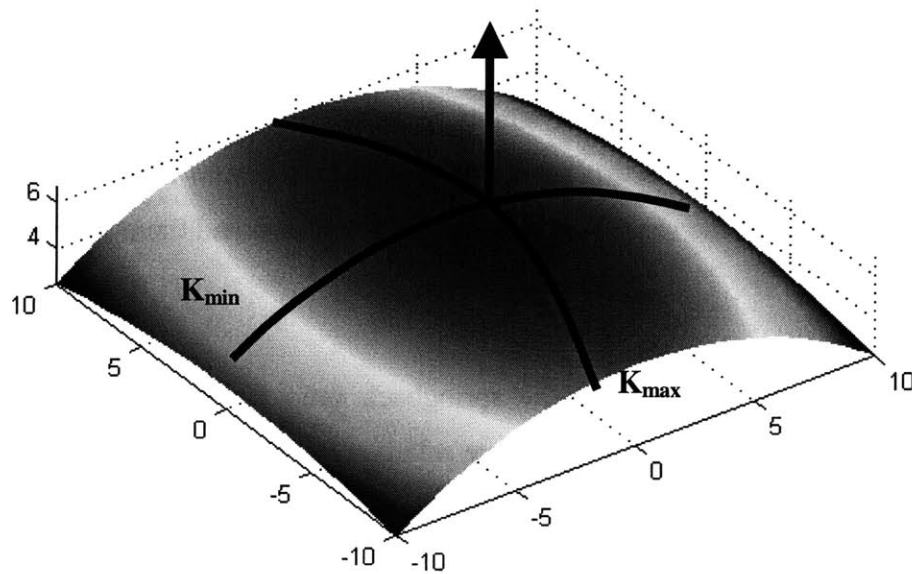


Figure 2-11: Representation of surface curvatures

The average of the two principal curvatures is known as mean curvature and is defined by Equation 2-11 while their product is known as Gaussian curvature and is defined by Equation 2-12.

$$K_{Mean} = \frac{K_{max} + K_{min}}{2}$$

Equation 2-11

$$K_{Gaussian} = K_{max} K_{min}$$

Equation 2-12

Mean and Gaussian curvature can be used to classify the various general forms that are manufactured in the stretch forming process. Figure 2-12 details the various classifications that are possible for the stretch forming process

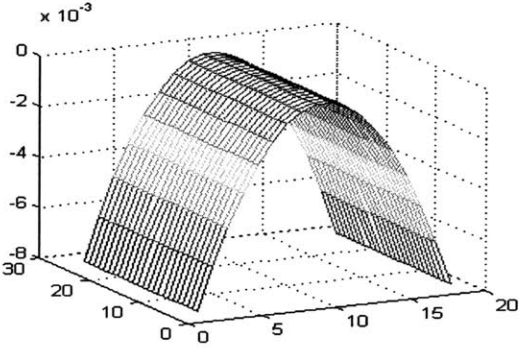
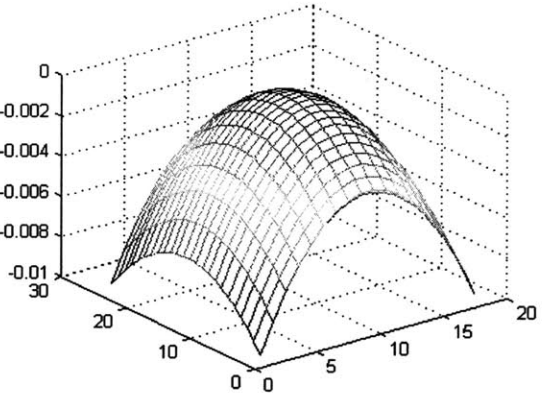
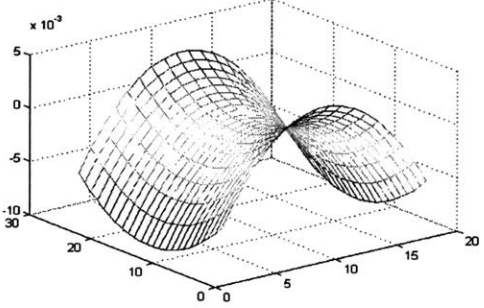
Gaussian Curvature	Mean Curvature	Shape
Zero	Positive	 <p data-bbox="1040 674 1203 709">Developable</p>
Positive	Positive	 <p data-bbox="1057 1157 1187 1192">Elliptical</p>
Negative	Positive, Zero, or Negative	 <p data-bbox="1057 1549 1182 1581">Toroidal</p>

Figure 2-12: Shape classifications based on mean & Gaussian curvature

It is observed that mean curvature generally describes the magnitude of the surface's curvature while Gaussian curvature is more indicative of the general shape classification of the surface (developable, Elliptical, or toroidal). The previous analysis of the stretch forming process only considered beginning with a flat piece of sheet metal ($K_{Mean}=0$,

$K_{\text{Gaussian}}=0$) and developing it around a cylindrical die ($K_{\text{Mean}}=\text{positive}$, $K_{\text{Gaussian}}=0$). Where ‘developing’ a surface indicates the process of modifying a surface’s mean curvature while leaving its Gaussian curvature unchanged. When a surface is developed, all strains are imparted uniformly to the cross of the material except for the bending² strains as shown in Figure 2-6. However, when a sheet’s Gaussian curvature is modified during the forming process, there are additional, in-plane strain modifications that occur in the sheet. These additional deformations can be visualized in Figure 2-14. Here, a grid is drawn over a flat sheet, which represents a material blank in the forming process. In the case where the blank is developed around a die of zero Gaussian curvature, the grid pattern remains un-altered with respect to the surface (ignoring pre-stretch and post-stretch portions of the process). However, when the Gaussian curvature of the sheet is altered, the grid pattern is grossly deformed in relation to the surface. These additional deformations were not included in the previous analysis of the stretch forming process. They are very difficult to account for analytically without the assistance of a complex finite element analysis (FEA).

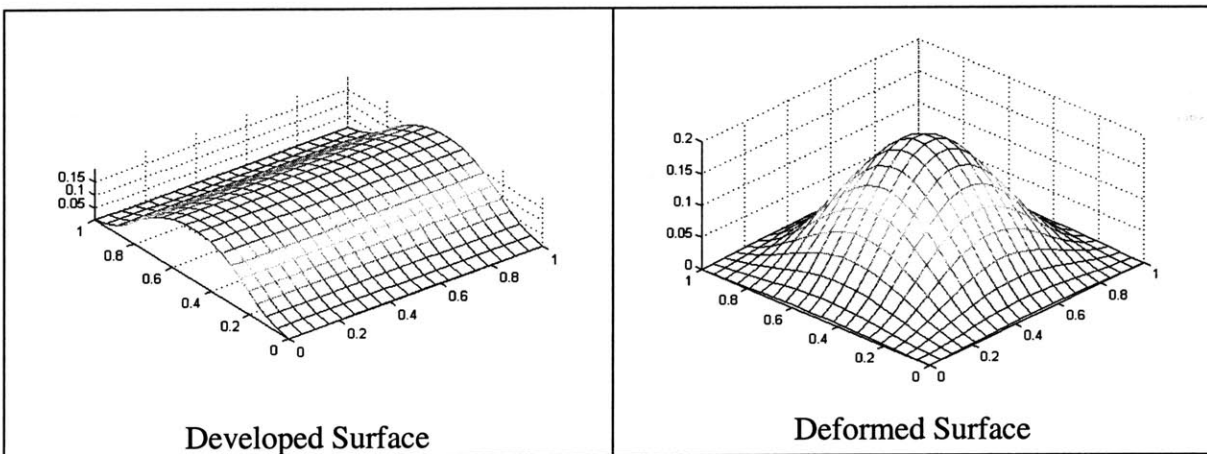


Figure 2-14: An example showing in-plane strain

Mean curvature is often referred to as an extrinsic characteristic of a surface while Gaussian curvature is considered an intrinsic surface characteristic. This makes intuitive sense when one considers a surface to have infinitesimal thickness. In such a case, the forces (or moments) associated with bending or altering the mean curvature of a surface

² Ignoring edge effects

are non-existent. A surface that has only undergone a change in mean curvature is identical when viewed from a point on the surface. That is, the relationship between any two points on the surface has not changed if they are defined in terms of surface coordinates. In other words, the internal or intrinsic properties of the sheet remain unaltered. However, if the surface is defined from an external reference, the representation of the sheet has changed, indicating an extrinsic property change. In the case of changes in Gaussian curvature, the surface representation is altered from both an internal (intrinsic) and external (extrinsic) point of view.

Summary

This chapter has provided an overview of stretch forming basics. This includes the machinery used in stretch forming and the various stages involved in stretch forming sheet metal components. It also included an analysis of the basics mechanics of stretch forming. Furthermore, surface geometry classifications have been presented which may provide useful for future research.

The following chapter provides more details on the specific problems that have been addressed by prior research at MIT. The topics covered in the next chapter include those that are peripheral, but pertinent to the shape control algorithms researched in this thesis.

Chapter 3 : Research Background

Stretch forming in the aerospace industry has been plagued by three primary problems. First, the process is extremely capital intensive, which adds significant costs and delays to operations. Second, the development of stretch forming dies has long been more of an art than a science. This generally leads to poorly quality parts and lengthy product development cycles. Finally, the conventional methods used to control the stretch forming process lead to excessive process variation. This chapter provides details of how prior research has addressed these issues.

Issue # 1: Excessive Capital Costs for Stretch Forming

The stretch forming process is extremely capital intensive. One can see from Figure 1-1, that it takes many different types of skins to cover the average airplane. Traditionally, each of these skins has been associated with at least one rigid tool as shown in Figure 3-1. Low production volumes and extremely large product variety mean that large amounts of storage space are required to contain many costly dies. This also suggests that excessive amounts of non-value added time is spent retrieving dies and changing the stretch press from one die to another whenever a different part is required.

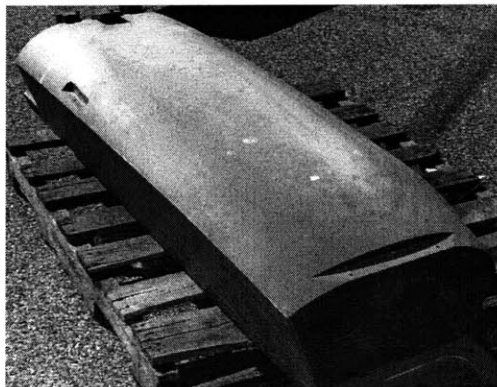


Figure 3-1: Typical rigid stretch forming tool

To address this issue, prior research by MIT [Olsen, 1980], [Walczyk,1981, 1999, 1999], [Goh, 1984], [Robinson, 1987], [Knapke, 1988], [Eigen, 1992], [Boas, 1997] and Northrop Grumman [Papazian, 1999] produced a flexible tool capable of replacing

numerous rigid tools. In part, this was done to reduce the cost associated with die manufacturing, die changeovers, part proliferation, and die storage. The traditional, smooth, rigid die like the one shown in Figure 3-1 is replaced with a re-configurable die created from a grid of spherical tipped, square pins as shown in Figure 3-2 and Figure 3-3. The individual pins that make these dies can be moved independently, which allows this die to change shape.

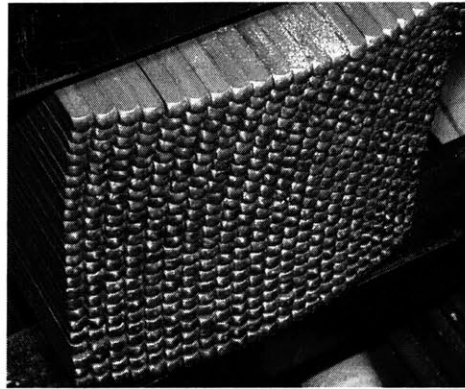
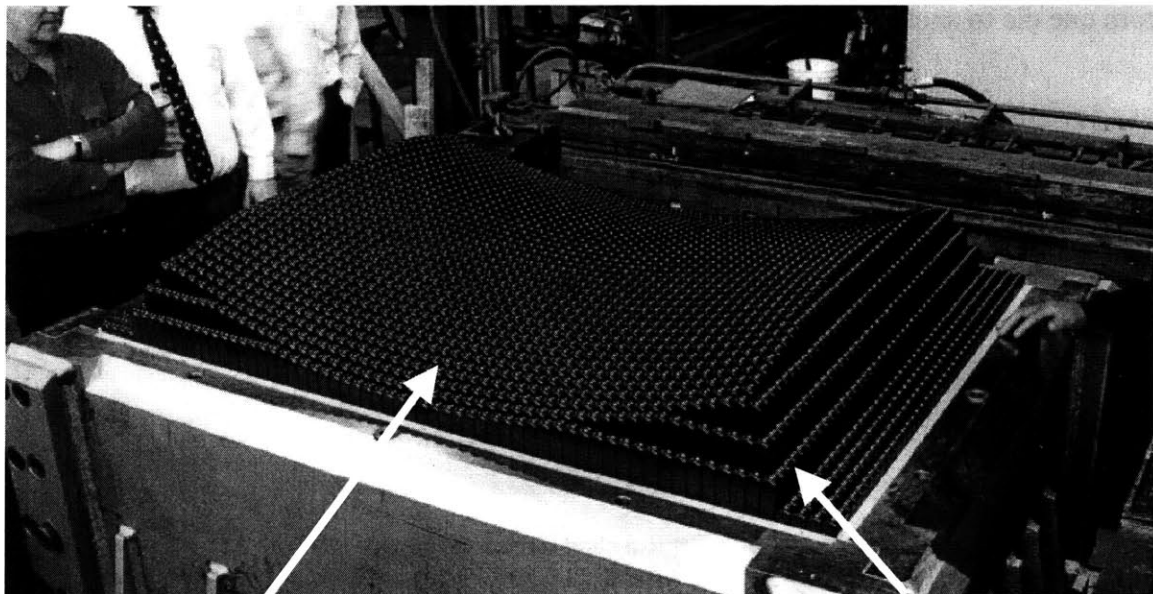


Figure 3-2: Forming surface of MIT, lab scale, re-configurable tool [Valjavec, 1999]



Die Surface
(Comprised of all pins)

Movable Pins
(Controlled by individual motors)

Figure 3-3: Forming surface of Northrop Grumman / MIT, production scale, re-configurable tool

To prevent the discrete, die surface of the re-configurable tool from creating ‘dimples’ in the sheet metal surface, an interpolator is placed between the die and the sheet. The interpolator is an elastomeric material that is used to distribute the contact pressure evenly about the sheet metal to die interface. This prevents the sheet metal parts from taking on a dimpled surface like the die. A drawing representing the placement of the interpolator over spherical capped pins is shown in Figure 3-4. A lubricant, or Teflon sheets are also placed between the interpolator and the sheet metal to reduce the effects of friction.

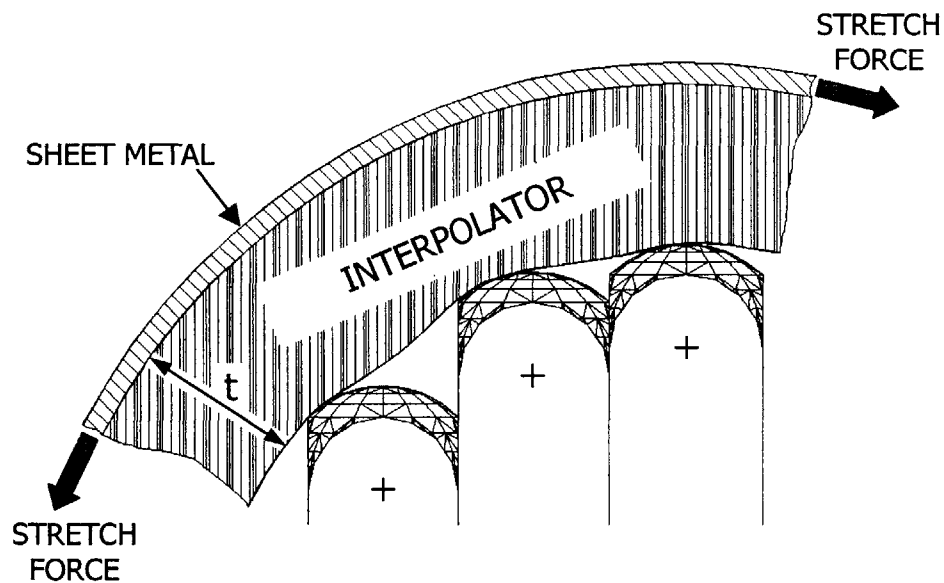


Figure 3-4: Interface between die, interpolator and sheet metal

There are several benefits to using a re-configurable tool. The most visible benefits are reductions in the number of required dies, storage space, and die changeover time. However, the re-configurable tool creates avenues for addressing other problems that have plagued the stretch forming process within the aerospace industry. Die shape can now be adjusted in a much shorter time frame, therefore an iterative approach to defining springback compensated die geometry may be used to replace or assist resource intensive analytic methods.

Issue # 2: Difficulty in Predicting Correct Die Shape

For developable part geometries, the mechanics of springback can be evaluated with relative ease. However, uncertainties about exact material properties and process parameters limit the usefulness of such approaches in predicting springback compensated die shapes. These situations are complicated further when non-developable shapes are desired. In lieu of analytical methods, die designers have traditionally used experience and guesswork to determine what shape to make a given die. This means that the die development process is typified by time-consuming die rework procedures as shown in Figure 3-5 or poor quality parts when designers are imperfect.

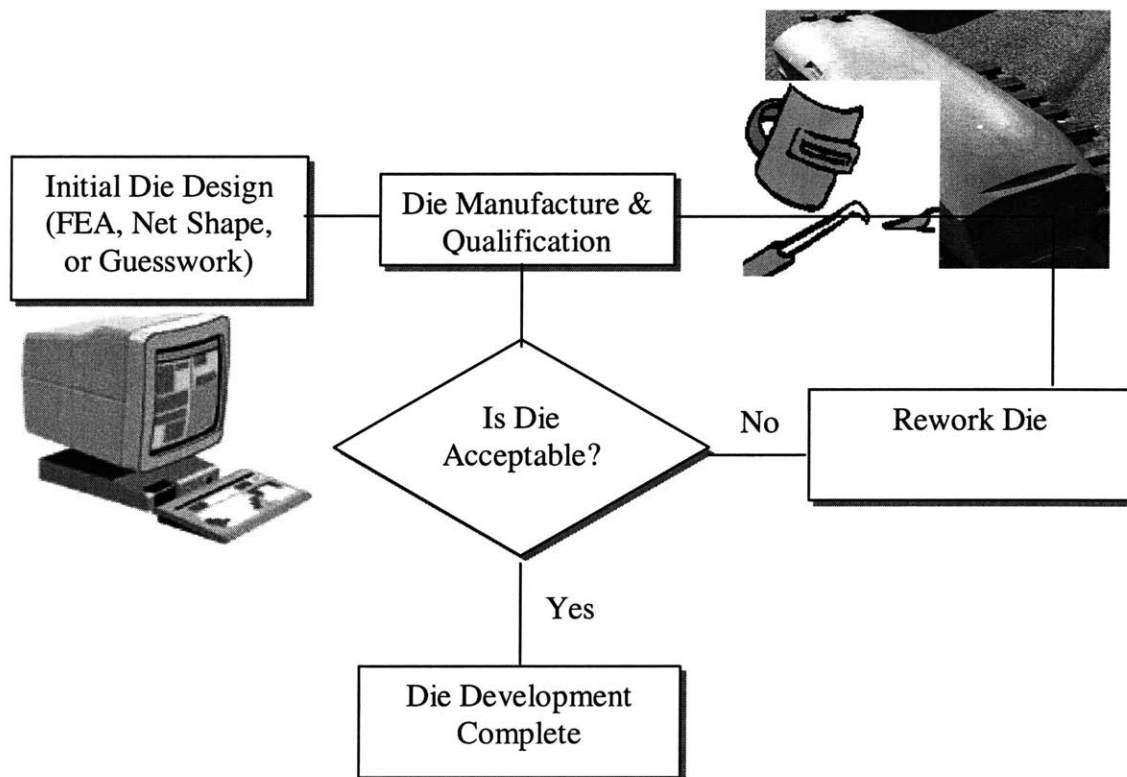


Figure 3-5: Rework Procedure

In recent years, die design has improved through the use of finite element analysis, which aids design engineers in predicting the correct, compensated die shape. The output plot shown in Figure 3-6 is from ABAQUS, one such finite element analysis program. The

Spring Forward Algorithm developed at MIT [Socrate, 1996] is an approach to die design that relies heavily on the use of FEA. In the Spring Forward Algorithm, a part is formed over a die with the reference part shape in simulation. The elastic springback of the part is measured and then subtracted from the reference part shape. This springback compensated shape is then used as the die shape in a real forming trial. While these approaches are a great improvement over guesswork, they do require significant engineering resources for the design of every die. Their success is also directly tied to the how well the design engineer estimates the specific stretch forming process parameters and material properties that will be used with a given die. Such estimates are difficult to make accurately.

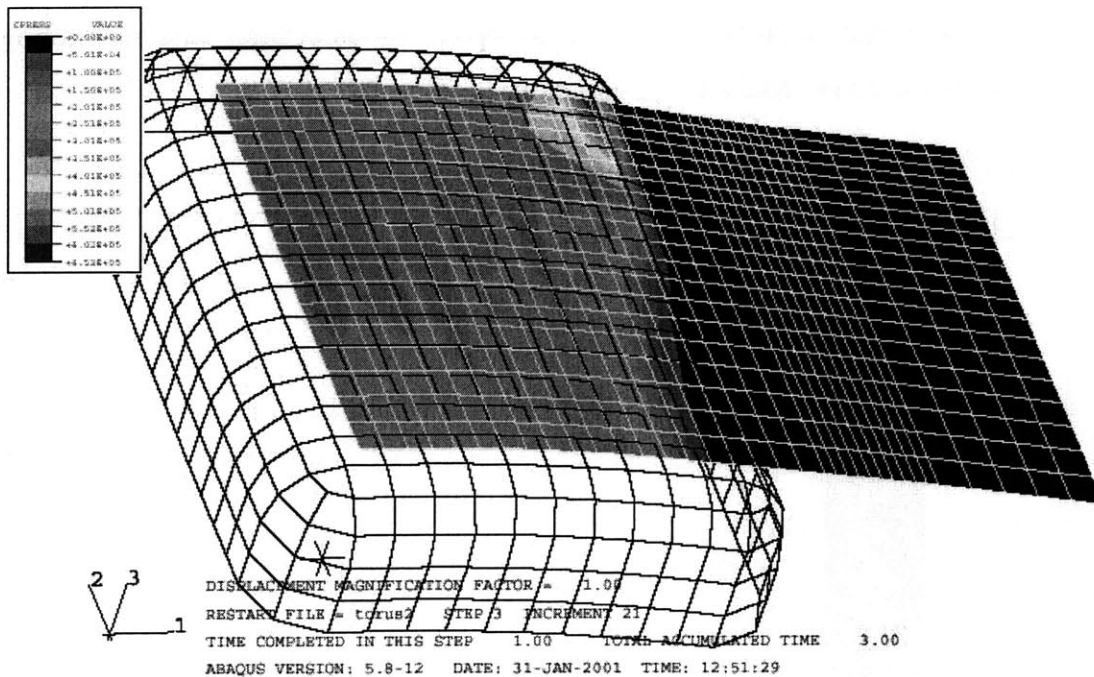


Figure 3-6: sample output of ABAQUS FEA

An in-process algorithmic approach to defining die shapes is made very attractive by the uncertainty of material and process parameters, and the amount engineering resource required to use FEA techniques effectively. The availability of a re-configurable tool also makes such an approach feasible and relatively easy to implement. It was these

factors that led Webb [1981,1987], Osterhout [1991], and Valjavec [1999] to the development of the algorithms detailed in this thesis.

Issue # 3: Excessive Shape Variation in Stretch Forming

The third problem is excessive variation in the geometries of produced parts. A lack of standard operating procedures (SOP's) and the choice of methods used to control the stretch forming process create excessive variation in practice [Parris, 1996]. The trajectory of the stretch forming process is often controlled by monitoring the pressure in the machine's hydraulic cylinders, the displacement of the hydraulic cylinder's, or even through the operator's rough, visual estimates of the process. The inherent variation associated with these control methods exacerbates the variation seen in parts formed under such control schemes. Work performed at MIT by Parris [1996] and Valentin [1999] has shown that implementing standard operating procedures and monitoring material strain levels can reduce process variation significantly.

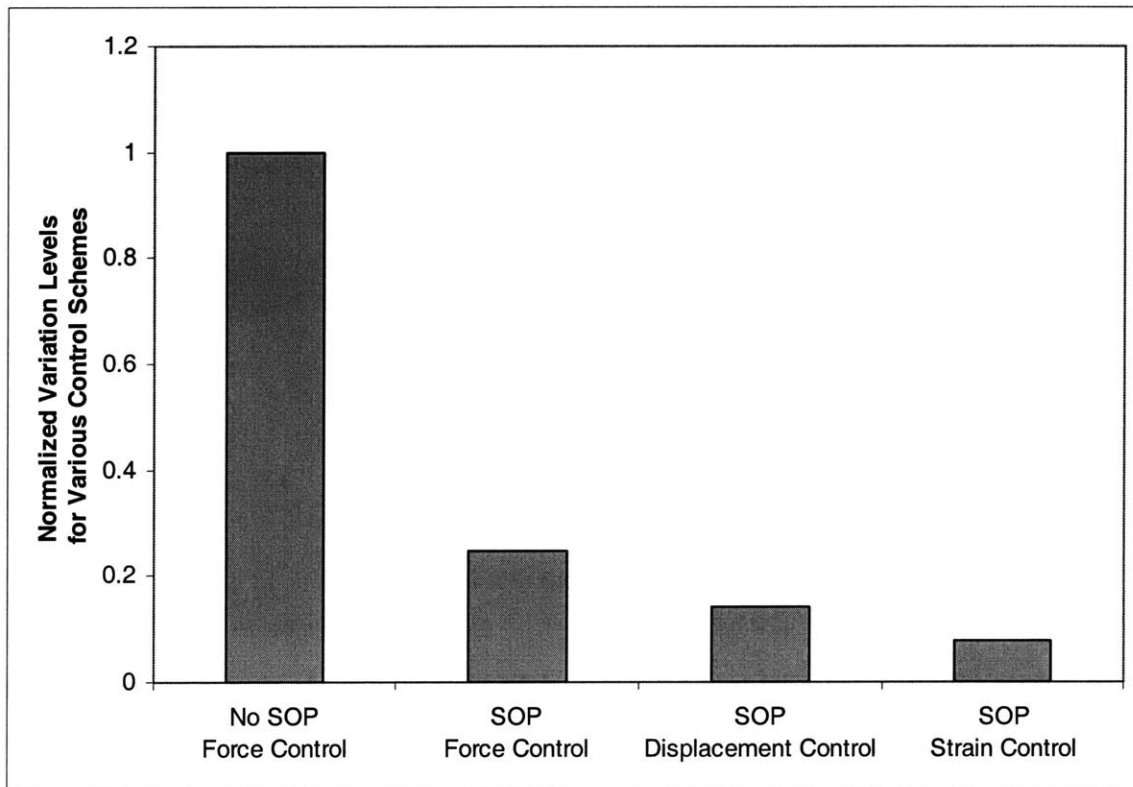


Figure 3-7: Variation levels associated with different process control strategies (SOP: Standard Operating Procedure)

The results of research performed by Andrew Parris [1996] on the variation levels associated with different control schemes are shown in Figure 3-7. The basic argument for using strain control can be explained by viewing the stress strain curve shown in Figure 3-8 [Hardt et al. 2001]. The stretch forming process generally requires all of the sheet metal to be stressed well into the plastic region. The figure depicts how slight variations in stress are associated with much larger variations in strain at these points of the stress-strain curve. When the stretch forming process is operated under force control, it is effectively measuring an integrated stress value over the cross section of the sheet. Minor variations in the measured force level are associated with much greater variations in the corresponding strain state of the material. The opposite is true when strain is measured.

Displacement is sometimes controlled during stretch forming as it is linked to the strain value of the sheet. Controlling displacement can be problematic as it is difficult to discern what portion of measured displacements are associated with the elongation of the sheet metal, and which portions are associated with the flexing of the stretch forming machine structure. Approaches that use displacement control also require significant engineering resources to define trajectories of all of the axes involved with the stretch forming process.

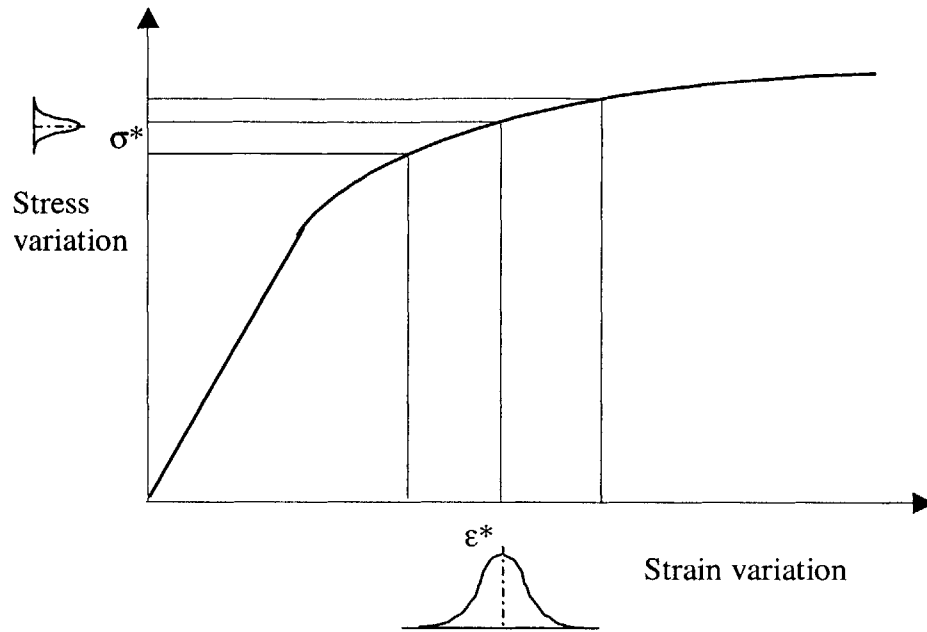


Figure 3-8: Stress – Strain curve sensitivities

Summary

The problems associated with stretch forming are classified into three categories. These categories include the large number of dies required, the difficulty associated with die design and the process variation associated with conventional stretch forming control modes. Prior research performed by MIT and Northrop Grumman has addressed these issues through the development of a reconfigurable die that can replace numerous fixed dies. This reconfigurable tool made possible the use of closed loop shape control algorithms for die design, based on an in-process, cycle-to-cycle shape measurement. This research builds on these shape control algorithms. Prior research has also identified control modes that minimize process variation [Parris, 1991; Valentin, 1999]. This becomes critical to the shape control algorithms as it they can sometime amplify process variation [Siu, 2001].

Chapter 4 : The Generic Algorithm Structure

In this chapter, a general framework for all of the closed loop shape control algorithms is presented. This framework is presented graphically, through recursive equations, and through the context of a discrete control system. These frameworks provide a means to quantify the expected performance of the algorithm. Additionally, some implementation issues common to all forms of the algorithm are presented.

Stretch forming process model

A model of the stretch forming process is first developed to help understand how the algorithm attempts to find the appropriate die shape. The stretch forming process is decomposed into the following categories: input, equipment, material, and output. These categories are shown in Figure 4-1. The equipment is further broken down into equipment properties and states, which collectively represent how the stretch forming process applies energy to the material. The material is also defined by its various properties and states, which quantify how material is affected by the energy applied to it. Some of the various equipment and material properties and states are provided in Figure 4-2.

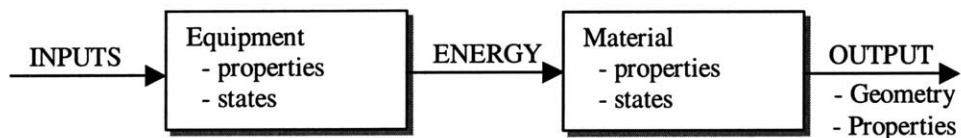


Figure 4-1: Stretch forming process model

	Equipment	Material
Properties	Die Geometry Structural Geometry Structural Stiffness	Yield Stress Elastic modulus Strain Hardening Coefficient
States	Hyd. Cylinder Pressure Hyd. Cylinder Positions	Net Stress or Strain Bending Moment

Figure 4-2: Some stretch forming process parameters

Controlling the geometry output of this process has been the historical problem and is thus the primary concern of the algorithm. Die geometry is the equipment property with the most impact on the output part geometry. It is also the one property that can be readily controlled through the discrete die. For these reasons, it is chosen as the variable to feedback and modulate in attempts to achieve a desired output geometry. Attempts are made to keep all other equipment/material properties and states constant from one forming cycle to the next in the algorithm. Figure 4-3 depicts errors in the process output geometry (part shape) being fed back to change the equipment property (die shape) in effort to correct the output geometry.

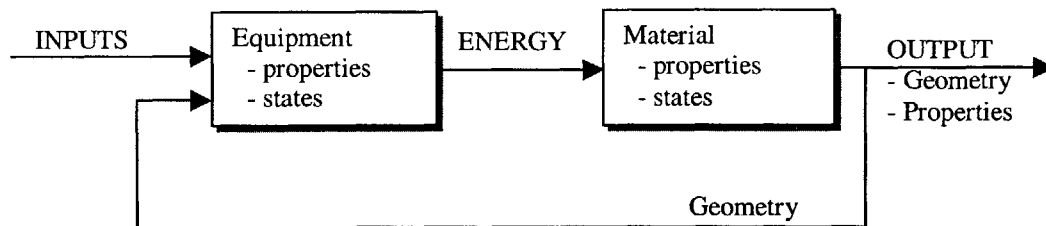


Figure 4-3: Stretch forming process model with feedback

Algorithm Model

Over the past 15 years, the general form of the die prediction algorithm has maintained the same basic structure. However, the algorithm has been applied to different processes by Webb [1981, 1987], Osterhout [1991] and Valjavec [1999]. The goal of the algorithm has always been to find the correct compensated die shape required to form a given part shape. Here, die and part shape are mentioned in generic terms, meaning they can reflect any parameter that is used to define a die or part shape in its entirety, or a collection of parameters used to define the die or part shape.

A flowchart depicting the general concept of the algorithm is shown in Figure 4-4. The process begins by taking the best guess at the appropriate die shape. The best guess can be made with finite element analysis, an evaluation of historical data, intuition or simply by choosing a net shape die as a starting shape. A part is then formed on this die and then

compared to the desired part. If the difference between the desired part and the first formed part (the part error) is within acceptable bounds, the process is terminated and the algorithm is not used. However, if the errors are not acceptable, the die shape is adjusted by a scaled factor of the previous part errors. The scaling factor that is used can be arbitrarily set to a value of unity, or can be defined by knowledge of the appropriate process physics, by historical data, or by system identification techniques performed on similar die and part shapes. A second part is then formed with the updated die shape and differences between the second part and the desired part are measured. The algorithm repeats in this manner until an acceptable part is made.

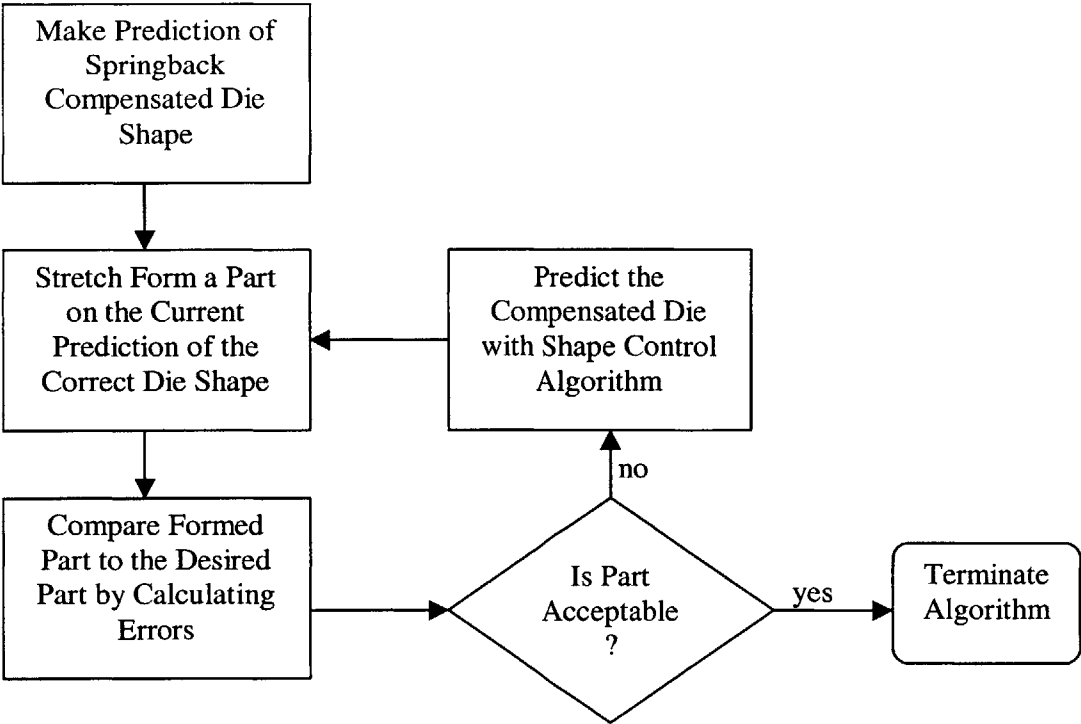


Figure 4-4: Block diagram of generic algorithm structure

A recursive equation is constructed to represent this form of the algorithm. It is shown in Equation 4-1. The difference between the target part and a formed part is the part error (e) as shown by Equation 4-2. This equation can be substituted into Equation 4-1, which produces Equation 4-3.

$$d_i = d_{i-1} + g_C(p_{ref} - p_{i-1})$$

Equation 4-1

Or

$$\begin{bmatrix} 1d_i \\ 2d_i \\ \vdots \\ MNd_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} 1d_{i-1} \\ 2d_{i-1} \\ \vdots \\ MNd_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} 1,1g_C & 1,2g_C & \vdots & 1,MNg_C \\ 2,1g_C & 2,2g_C & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ MN,1g_C & \vdots & \vdots & MN,MNg_C \\ (MN,MN) \end{bmatrix} \times \left(\begin{bmatrix} 1P_{ref} \\ 2P_{ref} \\ \vdots \\ MNP_{ref} \\ (MN,1) \end{bmatrix} - \begin{bmatrix} 1P_{i-1} \\ 2P_{i-1} \\ \vdots \\ MNP_{i-1} \\ (MN,1) \end{bmatrix} \right)$$

Where:

- d_i = vector representing die shape for the i^{th} iteration
- p_i = vector representing part shape for the i^{th} iteration
- i = iteration counter
- p_{ref} = vector representing reference part shape
- g_C = matrix of controller gain

It is helpful to view Equation 4-1 in vector / matrix format as shown below it. Here, each collection of die and part variables are contained in a column vector. For example, a die that consists of a grid of M by N pins is sometimes defined by the position of each of its pins. In this case there are M x N pins. While these pins are in a matrix form in the tool, there are shown as a column vector in Equation 4-1. The coefficients of the part and die values define which specific points of the part or they represent.

However, the controller gains are contained in matrix that has dimensions MN by MN. This is because a gain value is needed to relate every part data point to every die data point. The coefficients for these gain values indicate which particular part and die data points that are being related. The primary diagonal of this matrix contains gain values that relate a data point in a part to the corresponding point in the die. The off-diagonal terms related part data points to different die data points. These terms couple the action of one part data point to another die data point.

$$e_i = p_{ref} - p_i$$

Equation 4-2

Or

$$\begin{bmatrix} 1e_i \\ 2e_i \\ \vdots \\ MNe_i \\ (MN,1) \end{bmatrix} = \left(\begin{bmatrix} 1p_{ref} \\ 2p_{ref} \\ \vdots \\ MNp_{ref} \\ (MN,1) \end{bmatrix} - \begin{bmatrix} 1p_i \\ 2p_i \\ \vdots \\ MNp_i \\ (MN,1) \end{bmatrix} \right)$$

Where:

e_i = variable(s) representing error of the i^{th} part

$$d_i = d_{i-1} + g_C e_{i-1}$$

Equation 4-3

Or

$$\begin{bmatrix} 1d_i \\ 2d_i \\ \vdots \\ MNd_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} 1d_{i-1} \\ 2d_{i-1} \\ \vdots \\ MNd_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} 1,1g_C & 1,2g_C & \vdots & 1,MNg_C \\ 2,1g_C & 2,2g_C & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ MN,1g_C & \vdots & \vdots & MN,MNg_C \\ (MN,MN) \end{bmatrix} \times \begin{bmatrix} 1e_{i-1} \\ 2e_{i-1} \\ \vdots \\ MNe_{i-1} \\ (MN,1) \end{bmatrix}$$

Or alternatively, the change in die can simply be represented by Equation 4-4.

$$\Delta d_i = g_C e_{i-1}$$

Equation 4-4

Or

$$\begin{bmatrix} {}_1\Delta d_i \\ {}_2\Delta d_i \\ \vdots \\ {}_{MN}\Delta d_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} {}_{1,1}g_C & {}_{1,2}g_C & \vdots & {}_{1,MN}g_C \\ {}_{2,1}g_C & {}_{2,2}g_C & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ {}_{MN,1}g_C & \vdots & \vdots & {}_{MN,MN}g_C \\ (MN,MN) \end{bmatrix} X \begin{bmatrix} {}_1e_{i-1} \\ {}_2e_{i-1} \\ \vdots \\ {}_{MN}e_{i-1} \\ (MN,1) \end{bmatrix}$$

Where:

Δd_i = vector representing change to the i^{th} die

This recursive equation represents the feedback of part geometry information shown in Figure 4-3.

Schematic Process Model

Graphical models can be constructed for cases where one parameter is used to define the geometry of an entire part. Cylindrical parts of constant curvature are examples of this as they can be characterized by their curvature alone. Such a model is shown in Figure 4-5 when the assumption is made that other material and process characteristics remain constant, such as the final strain state or the material properties.

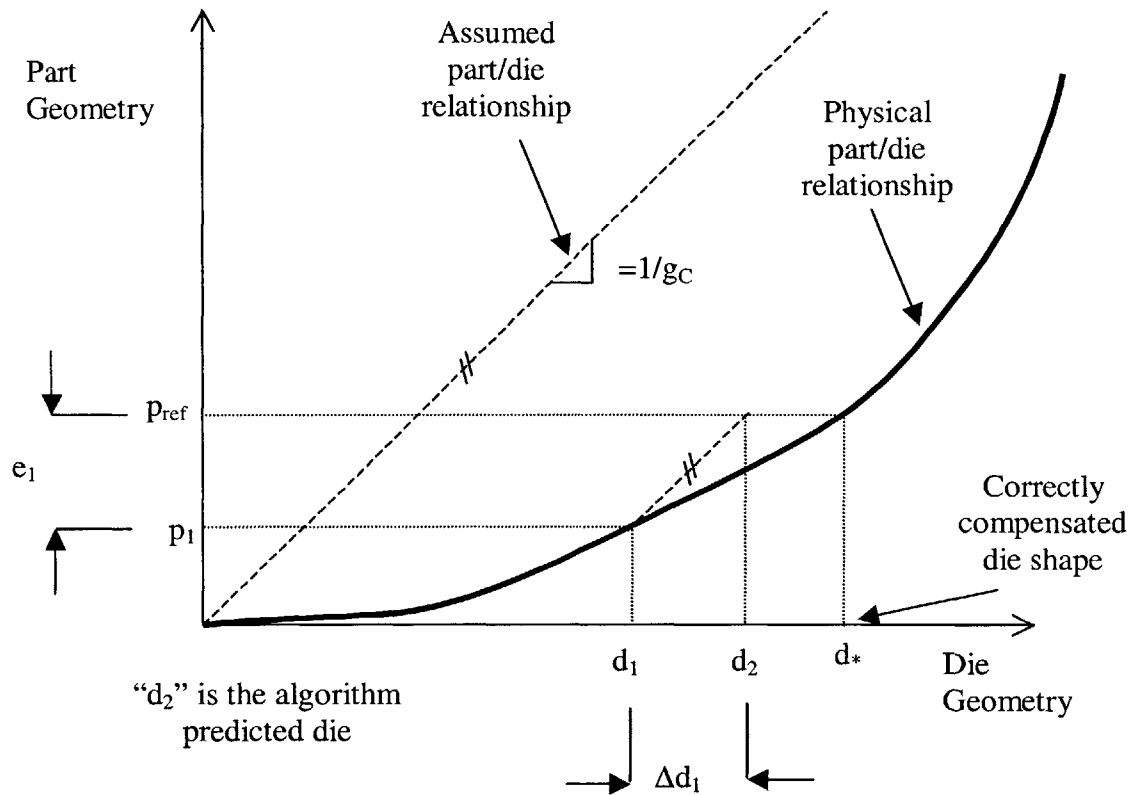


Figure 4-5: Schematic model of algorithm

Here the physical relationship between die geometry (d) and part geometry (p) is represented by the thick curve shown in the graph. While this relationship can be determined analytically for developable sheets of constant curvature, it is treated as unknown in this example. The goal is to find the correct die shape (d^*) that will create the target part geometry (p_{ref}). Because of springback, the line in Figure 4-5 will generally lie below the line with unity slope that crosses through the origin. The first guess at die shape is represented by (d_1) and the corresponding part is shown by (p_1). The difference between the target part and the first formed part is calculated (e_1) and then these errors are multiplied by weighting factors (g_c) to define how much the die should be adjusted. In this example, the weighting factor(s) are arbitrarily set to unity. The weighting factor (g_p) can be interpreted as a guess at the local slope of the physical relationship between part and die geometry. The resulting die prediction is the intersection between the assumed plant gains (g_p) and the reference part (p_{ref}). This is denoted in the figure by (d_2). This figure suggests that an approach that identifies the

slope of the physical part/die relationship may provide a more accurate prediction (one closer to d^*) and converge in fewer trials.

Control system view

The algorithm is now placed into the context of a discrete control system. The discrete control system framework is chosen because it allows us to apply established system analysis tools to the algorithm. This allows algorithm performance to be more easily quantified and also provides some insight as to how the algorithm functions. The framework of this system is shown in Figure 4-6.

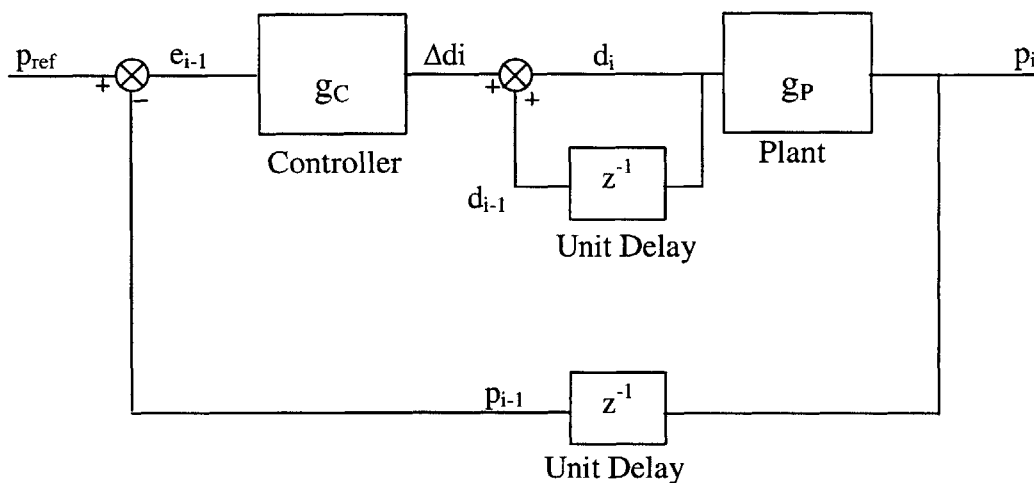


Figure 4-6: Control system view of algorithm

Here the process again begins with (d_i) being the initial ‘best guess’ at the die shape. This die is used to create the first part (p_i). The stretch forming process is represented by gain factors associated with the ‘plant’ (g_P). With the first part formed, the first cycle of the process is complete. The index is thus changed from ($i=1$) to ($i=2$) making the first part (p_{i-1}). This is represented in the schematic by the z -transform unit delay (z^{-1}). Errors (e_{i-1}) are then calculated for (p_{i-1}), and multiplied by the controller gain factors (g_C). The controller uses the errors and its gain factors to define what changes should be made to the die (Δd_i). These changes are then added to the previous die shape (d_{i-1}) to form the next die shape (d_i). The additional unit delay in this portion of the schematic represents

the memory the system has of the prior die shape. A new part is formed and the cycle continues.

Steady state error

It is possible to determine whether a steady state error is to be expected from the structure of the control loop. The first step in understanding this is to calculate the closed loop transfer function. This is performed in Equation 4-5 through Equation 4-7.

$$\frac{P_i}{P_{ref}} = \frac{\frac{g_C g_P}{1 - z^{-1}}}{1 + \frac{g_C g_P z^{-1}}{1 - z^{-1}}}$$

Equation 4-5

$$\frac{P_i}{P_{ref}} = \frac{g_C g_P}{1 - z^{-1} + g_C g_P z^{-1}}$$

Equation 4-6

$$\frac{P_i}{P_{ref}} = \frac{g_C g_P z}{z - 1 + g_C g_P}$$

Equation 4-7

The limit of the closed loop transfer function, multiplied by z and the unit step input is then taken as z approaches unity (this is represented by Equation 4-8). The value produced is unity, which means the desired output will match the input and no steady state error is expected. This is possible because the algorithm is adding the errors of the last part formed to those from all previous parts formed. This “integrating” effect allows convergence to a zero, steady state error. It should be noted that the algorithm takes the general form of a proportional – integrating controller (PI).

$$\lim_{z \rightarrow 1} (z-1) \left(\frac{g_C g_P z}{z-1 + g_C g_P} \right) \left(\frac{z}{z-1} \right) = 1$$

Equation 4-8

Response time

The root locus method can be used to understand the effect of system controller gains on the response of the closed loop system. To perform a root locus analysis, the open loop transfer function must first be calculated to create the root locus plot. Equation 4-9 reflects this transfer function.

$$\frac{P_i}{P_{ref}} = \frac{g_C g_P z}{z(z-1)}$$

Equation 4-9

This open loop transfer function has two poles, one located at one and the other at zero on the real axis. However, the pole located at zero is cancelled by the zero in the transfer function, also located at zero on the real axis. This effectively leaves one pole located at one on the real axis. The root locus for all closed loop gain values is thus a line extending toward negative infinity on the real axis as shown in Figure 4-7. This model assumes that g_C and g_P are system gains without their own dynamics.

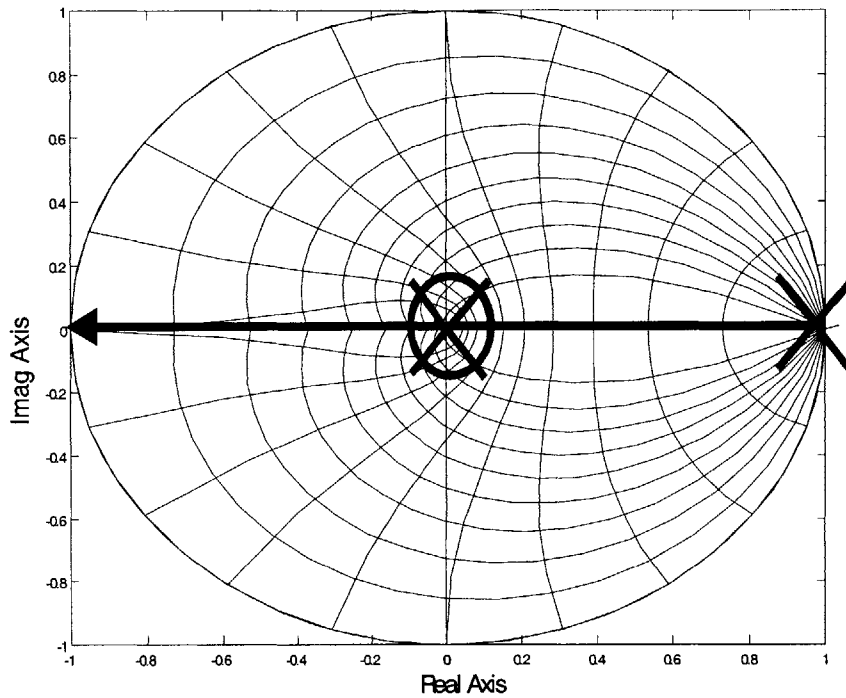


Figure 4-7: Root locus of die configuration algorithm

For loop gains ranging between zero and unity, the root locus suggest the response will exhibit an over damped behavior as depicted in Figure 4-8. These points lie on the real axis of the root locus plot between $Z=1$ and $Z=0$. The response time will be shorter for closed loop gains closer to unity, which represents points closer to the center of the unit circle.

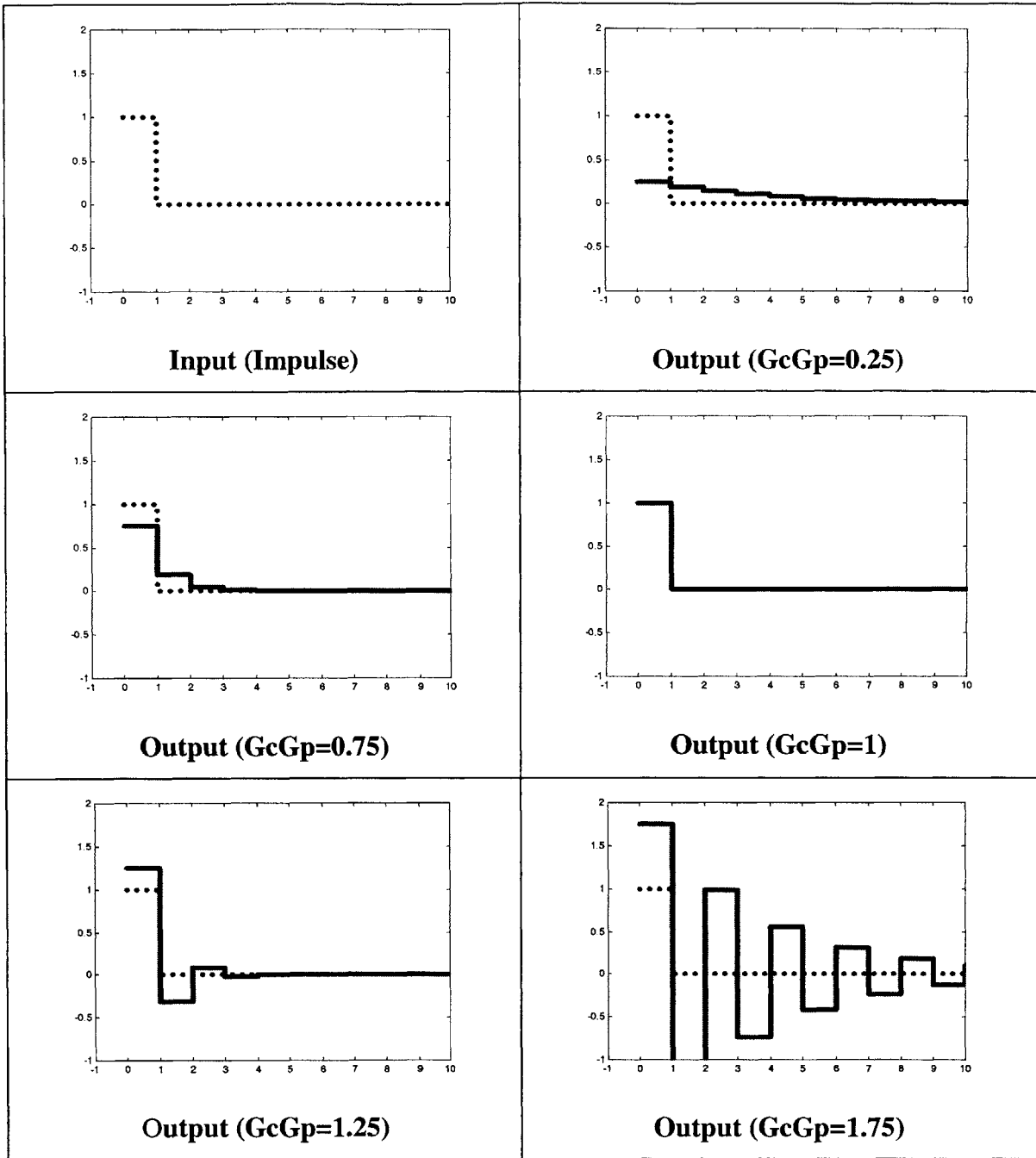


Figure 4-8: discrete, system responses

A closed loop gain value of unity provides the desired system dynamics of a dead-beat controller as shown in Figure 4-8. A controller is known as a ‘dead beat’ controller if it converges in a single iteration. This point lies in the center of the unit circle on the root locus plot.

Closed loop gains between values of one and two will cause the system to respond in the manner of an under damped system as shown in Figure 4-8. Closed loop gain values closer to unity one will converge faster and exhibit less overshoot. These points lie on the real axis of the root locus between $z=-1$ and $z=0$.

The system is conditionally stable at closed loop gain values of two, and the system is unstable beyond this value.

Disturbance Rejection

Another measure of system performance is how well it can reject unwanted disturbances. To better understand this, disturbances are modeled in the system as additive inputs. This is shown in Figure 4-9.

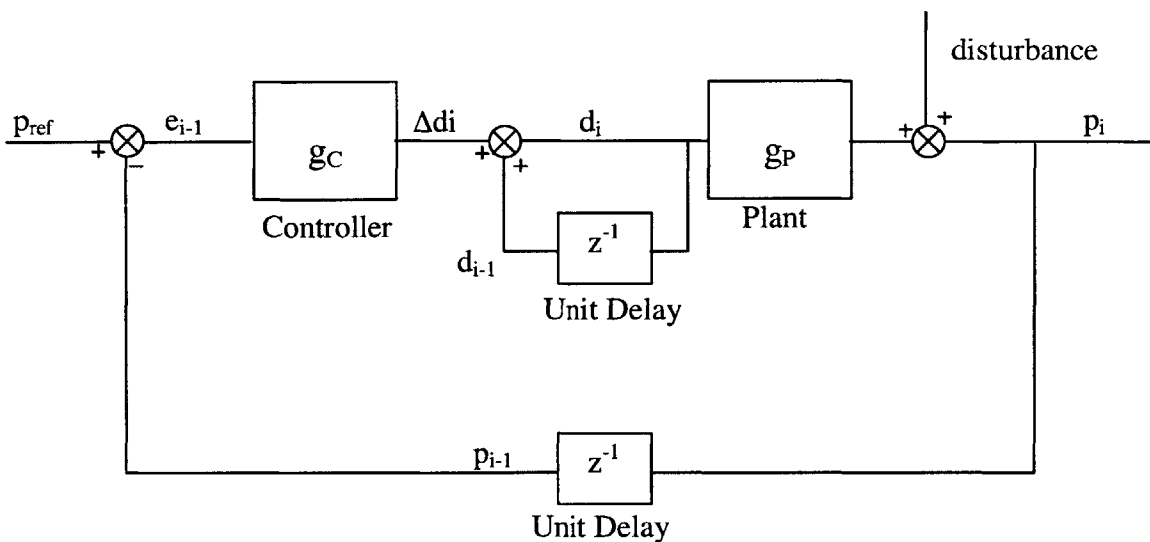


Figure 4-9: Control schematic for evaluating disturbance effects

First, the case of a constant disturbance is considered. Consistent, unexpected events in the forming process can act as constant disturbances. A physical example of a constant disturbance is a consistently mis-aligned forming tooling. To calculate the steady state effect of a constant disturbance, the limit of the closed loop disturbance transfer function is calculated as z goes to unity. The result defines the ratio between the output and the disturbance. This is performed in Equation 4-10 through Equation 4-13.

$$\frac{p_i}{dist} = \frac{1}{1 + \frac{g_C g_P z^{-1}}{1 - z^{-1}}}$$

Equation 4-10

Where:

dist = value of disturbance

$$\frac{p_i}{dist} = \frac{1 - z^{-1}}{1 - z^{-1} + g_C g_P z^{-1}}$$

Equation 4-11

$$\frac{p_i}{dist} = \frac{z - 1}{z - 1 + g_C g_P}$$

Equation 4-12

$$\lim_{z \rightarrow 1} (z - 1) \left(\frac{z - 1}{z - 1 + g_C g_P} \right) \left(\frac{z}{z - 1} \right) = 0$$

Equation 4-13

The resulting value of zero indicates that a constant disturbance will have no effect on the steady state outcome of the algorithm. However, the outcome is different when a varying disturbance is considered. If the varying disturbance is an un-correlated, random process then each value is independent of previous values. The system output will at best represent the error from the previous cycle of the algorithm. This is best understood by examining the recursive form of the algorithm in Equation 4-1. This equation can be modified to the form shown in Equation 4-14 when the error is comprised only of the disturbance.

$$\Delta d_i = g_C \cdot dist_{i-1}$$

Equation 4-14

Where:

Dist_{i-1} = value of disturbance at previous cycle

In the case of un-correlated, random disturbances, each successive value of the disturbance sequence is probably different and knowing prior values of the disturbance will not help in predicting the proper outcome. In many cases, high controller gains can have the adverse effect of increasing the variability of such uncorrelated disturbances [Siu, 2001].

In summary, the feedback structure of the algorithm provides a very effective mechanism for eliminating unwanted trends in the stretch forming process. However, this same mechanism amplifies the random, un-correlated noise that is found within the stretch forming process.

Common Implementation Elements

All of the forms of the algorithms face some common issues associated with implementation. These primarily include the way which the part and die are represented mathematically and what means are used to insure that these representations are accurate.

Geometry Parameters

In the simplest manifestations of the algorithm for constant curvature parts and dies, the die and part shape parameters d_i and p_i are simply the curvature of the i^{th} die and part. For more complex part shapes, the vertical displacements of points on a surface similar to the part are used to represent the component as represented in Figure 4-10. The surface shown in this figure represents a formed sheet that has been aligned with the tool it was formed on. The stem points in the figure represent discrete 'z' coordinate measurements of the surface. These points can be measured on a grid identical to that of the grid of the re-configurable tool or they may be taken on much finer grids. The dies are similarly represented by the spatial, vertical displacements of the individual pins that make up the shape of the die.

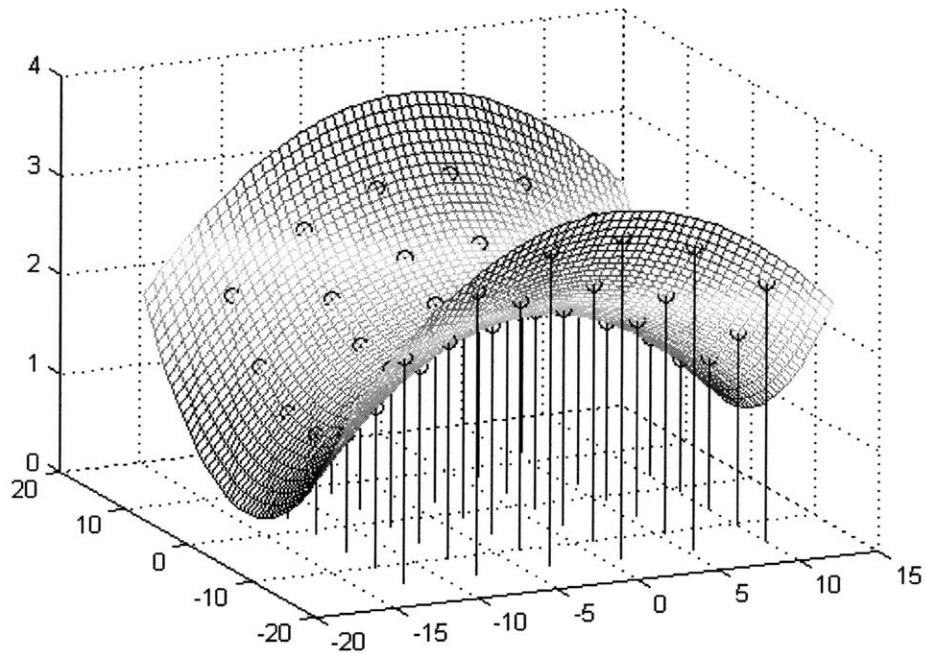


Figure 4-10: Part and Die shape parameters

Surface Fitting

Formed parts are generally measured using a coordinate measurement machine (CMM) similar to that shown in Figure 4-11. In this case, the output of the measurement process is a cloud of Cartesian points that represent the formed part. The CMM may be directed to measure a part on a grid that is spaced identically to the re-configurable tool, or any other desired grid of data points. It is however highly unlikely that the grid of measured points will coincide with the corresponding location of the tool on which it was formed. There thus exists a need to interpolate between the points measured by the CMM such that the part can be registered to the tool. Different approaches have been used to interpolate these data points. Bi-cubic spline interpolation techniques are however the techniques that have been employed most recently and are also the techniques used in this research. Most of various surface-fitting techniques used have the added benefit of smoothing the output data by acting as a low pass filter. This helps reduce some the noise associated with the measurement process. Some recent applications of the algorithm

have attempted to utilize Accordion Fringe Interferometry³ (AFI) for part measurement. AFI can produce much denser point clouds and may allow great improvements in the surface fitting routines.

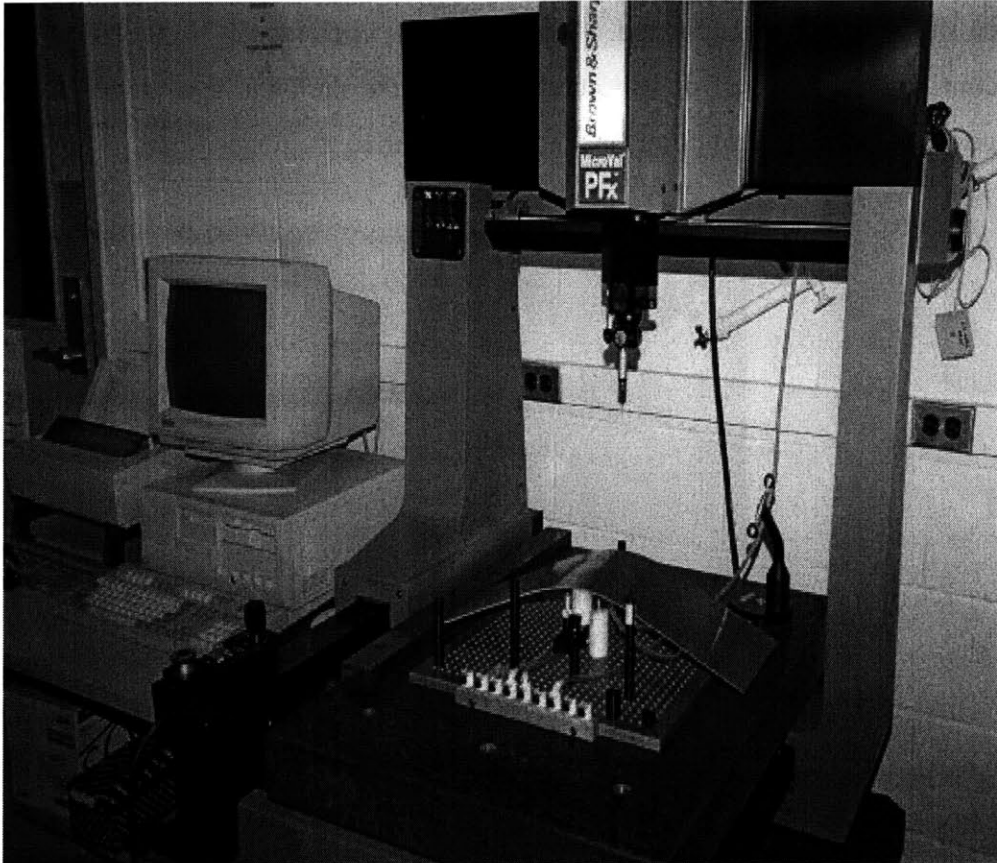


Figure 4-11: Lab Scale Coordinate Measuring Machine

Part Registration

Part registration is the process through which a formed, measured part is aligned with respect to the die it was formed on. In laboratory practice, each formed part is marked with a registration mark during the forming process. This mark orients the part relative to the die on which it was formed. The mark also serves as a common reference point for different parts, including the reference part. For prior lab scale implementations of the algorithm performed by Valjavec [1999], one reference point has been used for each part as shown in Figure 4-12. Using one reference point fixes each part in only two of the six

³AFI technology is currently under development by Dimensional Photonics Incorporated; Bedford, MA

possible degrees of freedom. Each part is still free to rotate about the x, y, and z-axis and also to translate along the z-axis. In order to fix the remaining four axes, the parts are generally fit to the reference part by iteratively adjusting the z-axis translation and the x, y, z-axis rotations until the root mean square (RMS) error between the part and the reference shape has been minimized. A grid of points coincident with the grid of pins in the re-configurable tool can now be created since the part has been registered to the reference shape. This grid of points is then used to define the formed part.

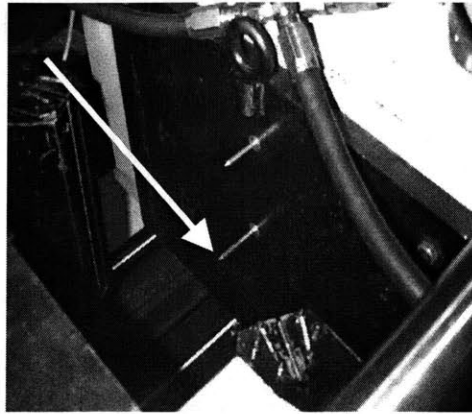


Figure 4-12: Reference mark used with MIT stretch forming press

Error Measurements

Error calculations can be made when formed parts and the target part shape have been registered in an identical manner. These calculations are performed by taking the difference between the z-coordinate component of the i^{th} part and the target reference part as reflected in Equation 4-15.

$$e_i = P_{ref} - P_i$$

Equation 4-15

$$\begin{bmatrix} 1,1 e_i & 1,2 e_i & \vdots & 1,N e_i \\ 2,1 e_i & 2,2 e_i & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ M,1 e_i & \vdots & \vdots & M,N e_i \end{bmatrix} = \begin{bmatrix} 1,1 P_{ref} & 1,2 P_{ref} & \vdots & 1,N P_{ref} \\ 2,1 P_{ref} & 2,2 P_{ref} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ M,1 P_{ref} & \vdots & \vdots & M,N P_{ref} \end{bmatrix} - \begin{bmatrix} 1,1 P_i & 1,2 P_i & \vdots & 1,N P_i \\ 2,1 P_i & 2,2 P_i & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ M,1 P_i & \vdots & \vdots & M,N P_i \end{bmatrix}$$

Or

The error for each part is thus comprised of an entire grid of points that is the same size as the grid used to define the parts and dies. It is most useful to view these grids of points as an error surface, as shown in Figure 4-13. The matrix form shown above should only be used for viewing. It is critical that errors are in column vector form as shown in Equation 4-2 when they are multiplied by system gains.

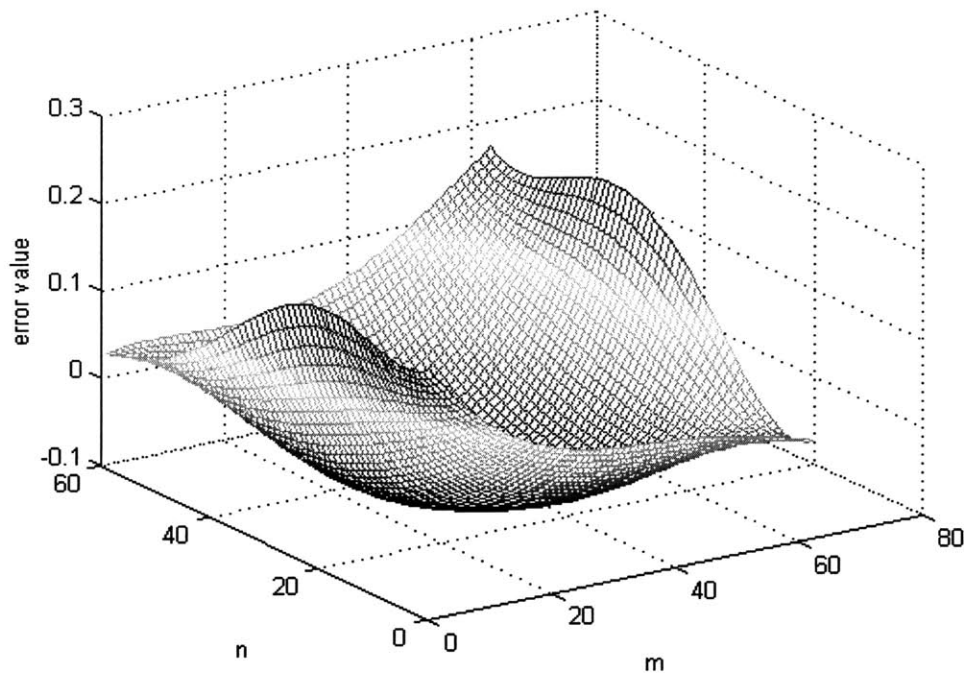


Figure 4-13: Error surface

It is also useful to condense the error of one part into a single scalar value that can be monitored for part quality. Past experiments and recent experiments documented in this thesis use the minimum Root Mean Square (RMS) error value, and maximum error value for this purpose. The value for RMS error is defined by Equation 4-16.

$$RMS \text{ Error} = \sqrt{\frac{\sum_{m=1}^M \sum_{n=1}^N (e_{(m,n)})^2}{MN}}$$

Equation 4-16

Where:

- M = number of measured points in the x-direction
- N = number of measured points in the y-direction
- m = counter for the x-direction
- n = counter for the y-direction
- $e_{(m,n)}$ = error at the point $x=m, y=n$

It is noted that the RMS error value can change if the error surface, or equivalently the formed part or reference part, is translated along the z-axis as shown in Figure 4-14. In this case the average value of the error (\bar{e}) has changed. Such a change in the average value of the error will also change the RMS error value.

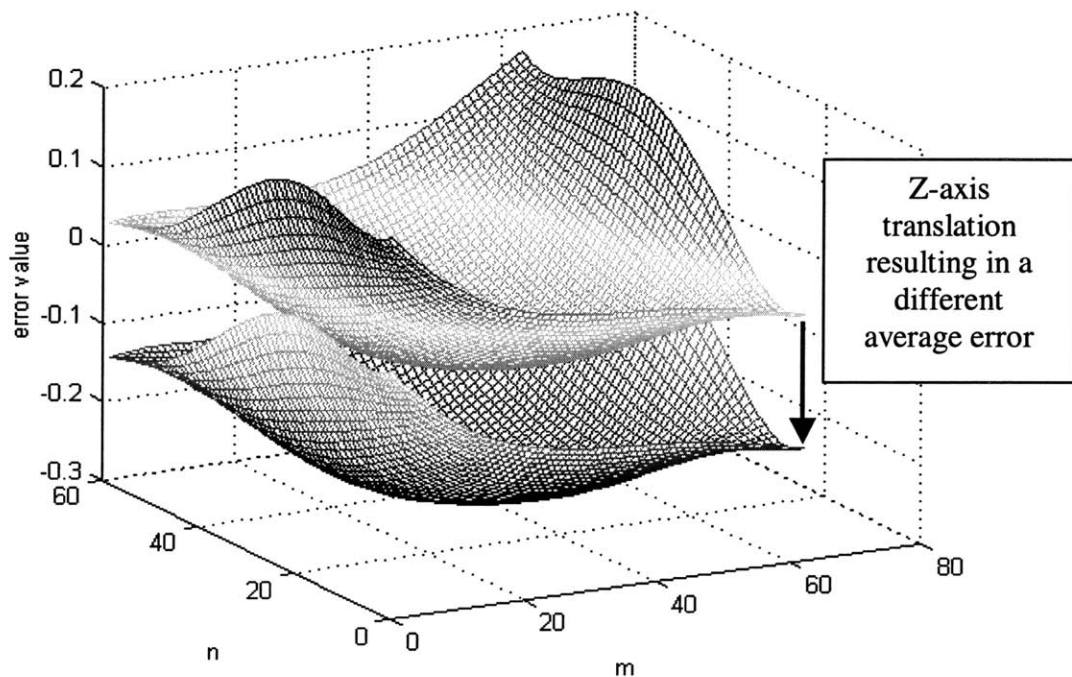


Figure 4-14: Altering the average error (\bar{e}) value by translating along the z-axis

The minimum RMS error does not depend on the average error and will always be the same value for a given part. It thus proves to be a good indicator of how well a part conforms to the reference part. The minimum RMS error is calculated with Equation 4-17, and is effectively the standard deviation of the error. It is noted that the previously mentioned registration procedure should orient the part such that the average error (\bar{e}) is equal to zero.

$$\text{Minimum_RMS_Error} = \sqrt{\frac{\sum_{m=1}^M \sum_{n=1}^N (e_{(m,n)} - \bar{e})^2}{MN}}$$

Equation 4-17

Maximum error is calculated by finding the difference between the maximum error value and the minimum error value. This is reflected in Equation 4-18. This value is also insensitive to the average error value.

$$\text{Maximum_Error} = \text{Max}(e) - \text{Min}(e)$$

Equation 4-18

Summary

This chapter has provided frameworks that all shape control algorithms can be placed into. These frameworks include graphical models, recursive equations and discrete control system schematics. The expected performance of the algorithms is also measured through the tools of discrete control theory. The algorithms are shown to have no expected steady state error, even to constant disturbances. However, random disturbances are shown to affect the algorithm adversely.

Some implementation issues common to all forms of the algorithm were also discussed in this chapter. These issues include the choice of geometry parameters, surface fitting, part registration and error measurement.

In the next three chapters, the different forms of the algorithms are evaluated in the context of the frameworks presented in this chapter.

Chapter 5 : Discussion of Algorithm #1

In this chapter, the first shape control algorithm developed by Webb [1981] is presented⁴. This algorithm is termed algorithm #1 in this research. It is placed into the general algorithm frameworks of chapter 4. Algorithm #1 is shown to be a relatively simple algorithm to implement. However, historical performances of algorithm #1 are reviewed and shown to have less than desirable performance. This sub-optimal performance is attributed to a lack of system identification and system coupling in the algorithm.

Basic Structure

Webb [1981] is responsible for the initial implementation of the die prediction algorithm. It was first used for pure stretch formed parts made on a reconfigurable tool. In this implementation, the generic algorithm defined by Equation 4-1 can be simplified as shown in Equation 5-1. The control system previously detailed in Figure 4-6 schematic can also be simplified to that shown in Figure 5-1, where the controller gain matrix is the identity matrix.

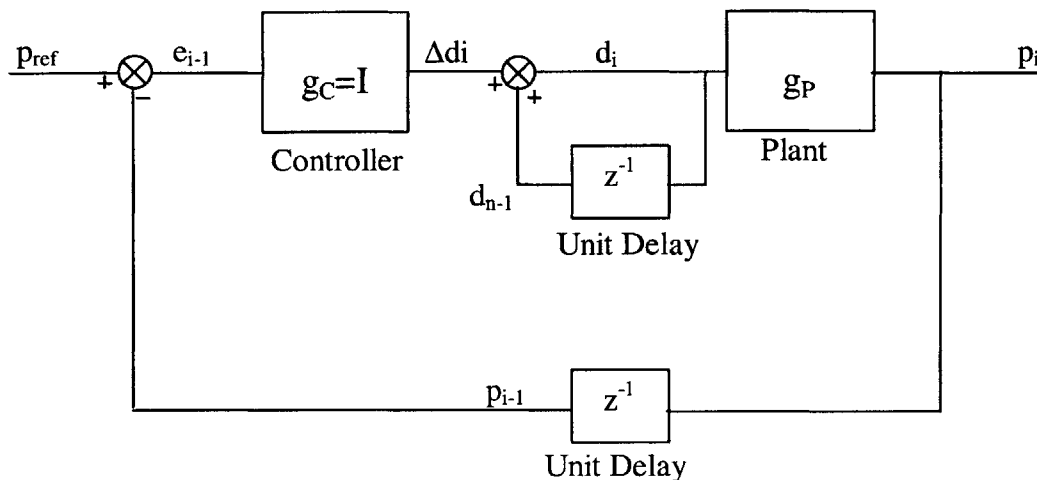


Figure 5-1: Algorithm #1 control schematic

$$d_i = Id_{i-1} + e_{i-1} \quad \text{or} \quad d_i = d_{i-1} + e_{i-1}$$

Equation 5-1

Or

$$\begin{bmatrix} {}_1d_i \\ {}_2d_i \\ \vdots \\ {}_{MN}d_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} {}_1d_{i-1} \\ {}_2d_{i-1} \\ \vdots \\ {}_{MN}d_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \\ (MN,MN) \end{bmatrix} \times \begin{bmatrix} {}_1e_{i-1} \\ {}_2e_{i-1} \\ \vdots \\ {}_{MN}e_{i-1} \\ (MN,1) \end{bmatrix}$$

This is the simplest implementation of the algorithm. Component errors are calculated once formed parts are measured and registered. If the errors are not below an acceptable level, they are added to the prior die shape and a new part is formed. Note that the matrix of controller gains (g_c) equal to the identity matrix implies no coupling and no attempt to understand the true ‘process gain’ as shown in Figure 4-5.

Implementation

Implementation issues associated with algorithm #1 are limited to the standard surface fitting and part registration issues which are discussed in Chapter 4.

Historical Experimental Results

Experiments performed by Webb [1981] on stretch formed parts with positive Gaussian and mean curvature provided positive results as shown in Figure 5-2. Here, the absolute value of the error, the root mean square error (RMS), and the error at the part center decreased with each iteration of the algorithm. However, the algorithm generally takes too many iterations to reach an acceptable error level.

⁴ This algorithm was also referred to as a “zeroth order deformation transfer function” by Valjavec [1999]

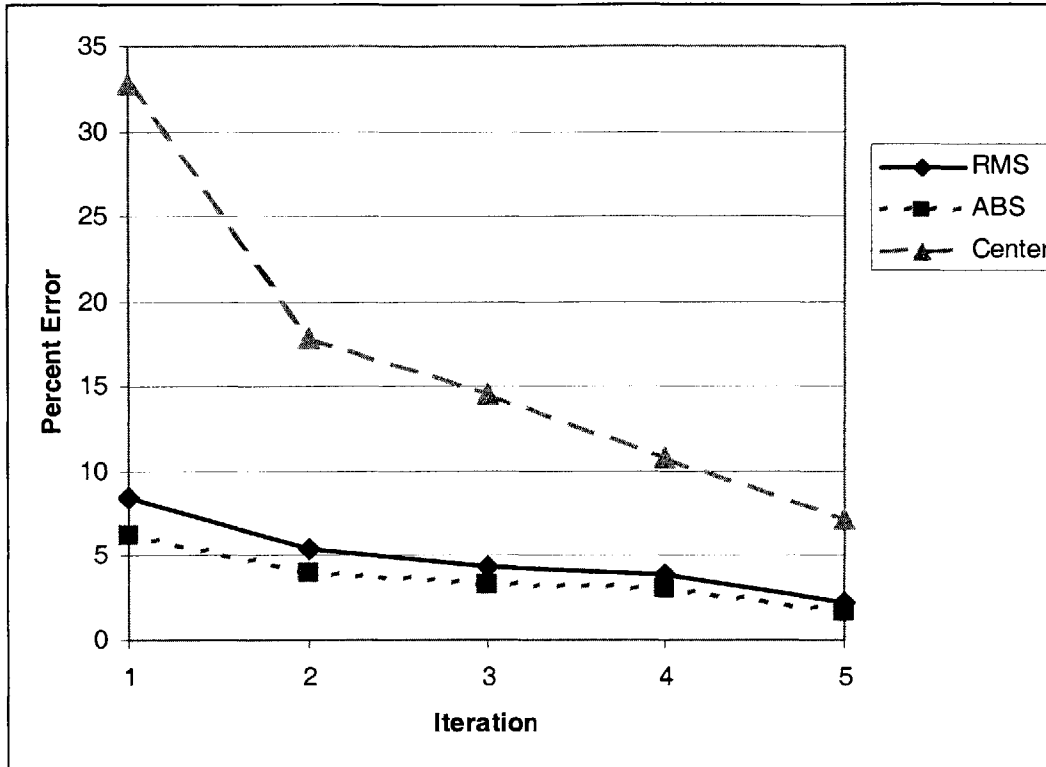


Figure 5-2: Algorithm #1 results from Webb [1987, p. 59] on match die tooling

Algorithm Concerns

The sluggish response of the algorithm was primarily attributed to two shortcomings of this approach. First, this approach treats each pin in the tool and the corresponding portion of the part above it as independent, Single Input – Single Output (SISO) systems. These systems are coupled to one another. Second, no attempts are made to identify the relationship between the input and output of the stretch forming plant.

System Coupling

This algorithm treats each pin within the re-configurable tool and the section of the part that lies directly above it as an independent part of the entire system. This is a reasonable assumption when the view is limited to the tool. However, a basic understanding of the forming process suggests this assumption is invalid. Figure 5-3 is a schematic representing the movement of one pin within a re-configurable tool. It also shows how

the movement of this one pin might effect both the position of the part directly above the pin and the neighboring pins. This implies that treating each pin within the tool, and the part directly above it as a SISO system is a poor assumption because parts can be affected by moving other pins in their vicinity. The next form of the algorithm includes a means to accommodate the coupling that occurs between the individual components of the system.

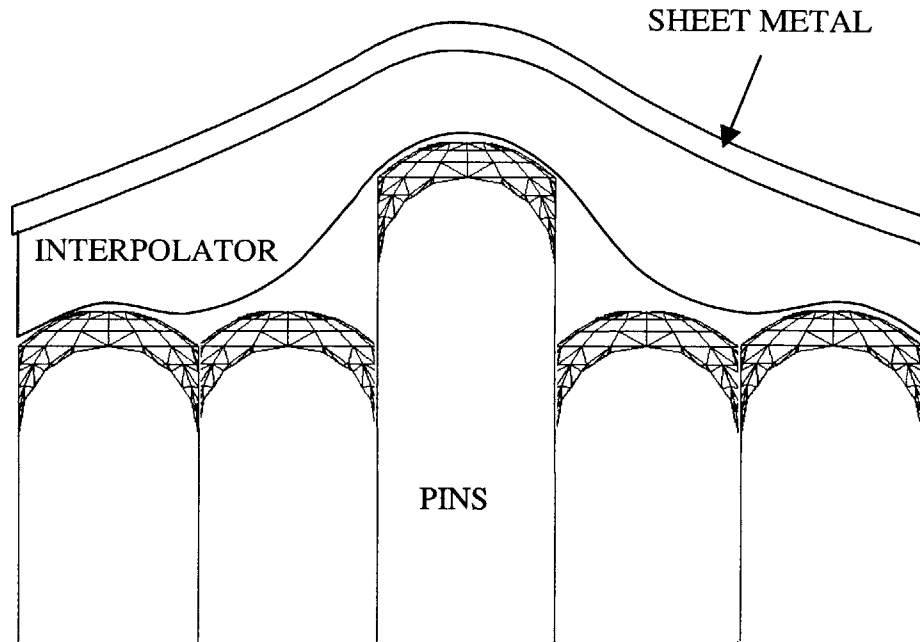


Figure 5-3: Depiction of system coupling

System Identification

The first algorithm makes no attempt to identify the nature of the system that it is controlling. From the discrete control system perspective, it sets the controller gains for each of the independent SISO systems to unity. The system response is entirely up to the nature of the forming system as no attempts are made to compensate for it. The graphical example of a system shown in Figure 5-4 helps make this point. Here, the part and die geometry parameter consist of one element that can define an entire part or die, like curvature for cylindrical parts. The dark line in Figure 5-4 represents the relationship between these two parameters.

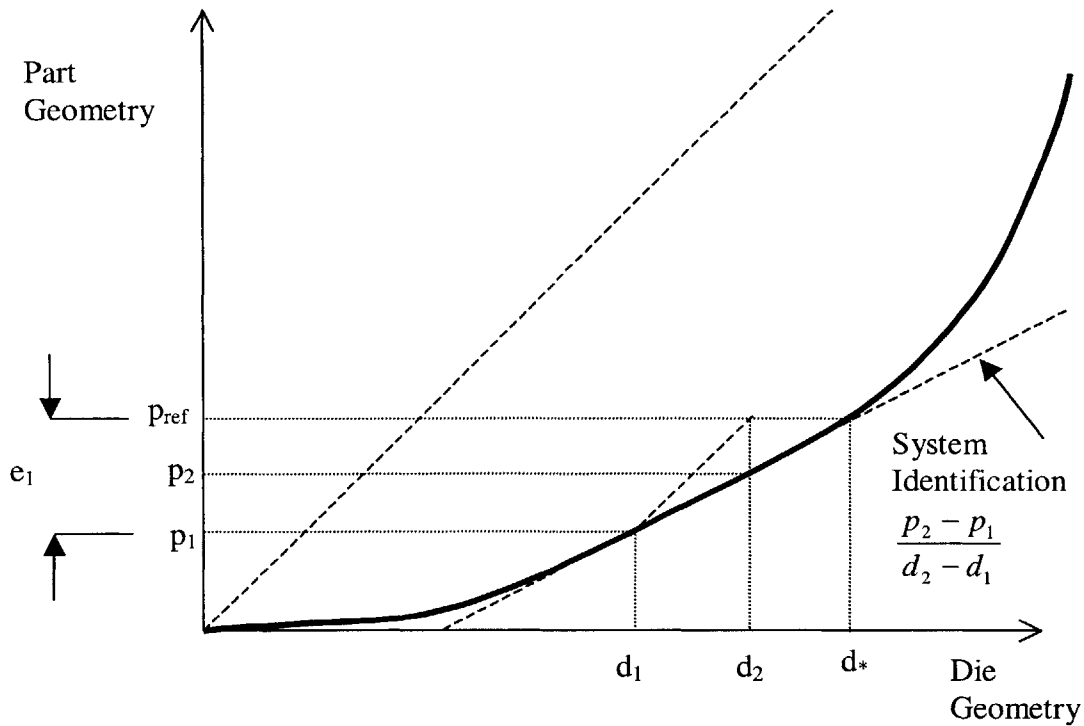


Figure 5-4: graphical interpretation of system identification

Without a-priori knowledge of the relationship between the system input (die shape) and output (part shape), it is assumed that a change in one parameter is equivalent to the change in the other. This is represented in Figure 5-4, by the dashed line that has a unity slope value. This assumption leads to the sub-optimal performance experienced with Algorithm #1. However, if an additional calibration part is formed (p_2) on a different die (d_2), the relationship between the change in die parameter and change in part parameter can be estimated. This concept is shown in by the linear estimation of the lines slope between the two calibration parts and dies.

Summary

Algorithm #1 is shown to be a relatively simple algorithm to implement. However, it requires too many iterations to arrive at an acceptable part. Algorithm #1 is believed to perform slowly because it makes no attempt to identify the system it is controlling and also makes no attempt to accommodate the coupling in the system it is controlling.

Chapter 6 : Discussion of Algorithm #2

In this chapter, algorithm #2 is presented. This algorithm addresses the two previously mentioned areas of concern with algorithm #1. Prior to this research, the operations involved with algorithm #2 have only been viewed from the frequency domain perspective. Alternative, spatial domain views are now presented which add insight into the algorithm. These views allow the coupling introduced by algorithm #2 to be quantified, and thus a means to quality check the algorithm. Algorithm #2 is shown to improve on the performance of algorithm #1, but at the cost of increasing complexity and sensitivity to noise.

Basic Structure

In order to improve on the algorithm, Webb [1987], Osterhout [1991] and Valjavec [1999] address two primary areas of concern with algorithm #1. First, steps are taken to quantify the physical relationship between the part and the die through system identification techniques. This system identification is then used to calculate appropriate controller gains to achieve the most desirable system response. This is reflected in the Figure 6-1 system schematic by the controller block with controller gains (g_c) which are determined by monitoring the plant (G_p). Second, the stretch forming process is modeled as a coupled system. This is accomplished through the introduction of the Fourier transform. The algorithm is applied to the part and die representations in the frequency domain. This provides coupling to the system as a change to a single frequency of a part or die can affect multiple points about the respective die or part. Figure 6-1 reflects the addition of coupling by inclusion of the Fourier transform and inverse Fourier transform blocks to the control schematic.

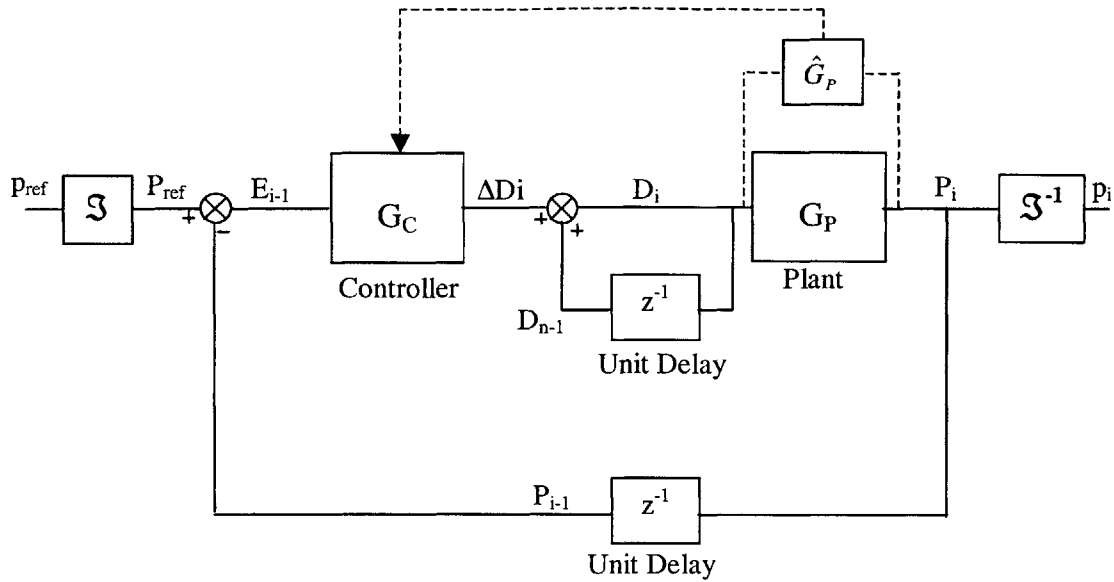


Figure 6-1: Algorithm #2 control system schematic
 (Note: all variables are Fourier transforms of particular part or die shapes)

This implementation of the algorithm is also described by Equation 6-1 through Equation 6-3, which reflect operations made in the frequency domain, or by Equation 6-4 and Equation 6-5, where key operations are performed in the spatial domain. In both of these cases, a full view of the equation is included to help view the differences. The part and die parameters shown in capital letters reflect frequency domain representations.

$$D_i = D_{i-1} + G_c (P_{ref} - P_{i-1})$$

Equation 6-1

Or

$$\begin{bmatrix} {}_1D_i \\ {}_2D_i \\ \vdots \\ {}_{MN}D_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} {}_1D_{i-1} \\ {}_2D_{i-1} \\ \vdots \\ {}_{MN}D_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} {}_1G_C & 0 & 0 & 0 \\ 0 & {}_2G_C & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & {}_{MN}G_C \\ (MN,MN) \end{bmatrix} \times \left(\begin{bmatrix} {}_1P_{ref} \\ {}_2P_{ref} \\ \vdots \\ {}_{MN}P_{ref} \\ (MN,1) \end{bmatrix} - \begin{bmatrix} {}_1P_{i-1} \\ {}_2P_{i-1} \\ \vdots \\ {}_{MN}P_{i-1} \\ (MN,1) \end{bmatrix} \right)$$

Where:

$$\begin{aligned} D_i &= \mathcal{I}(d_i) \\ P_i &= \mathcal{I}(p_i) \\ P_{ref} &= \mathcal{I}(p_{ref}) \end{aligned}$$

$$\hat{G}_P = \frac{P_2 - P_1}{D_2 - D_1}$$

Equation 6-2

$$G_C = \hat{G}_P^{-1}$$

Equation 6-3

In the view suggested by Equation 6-1, the multiplication operation of the algorithm is the same as in the general formula, but the spatial geometry parameters are replaced with their frequency domain equivalents via the Fourier transform. In this form, the matrix of controller gains (G_C) shows no off-diagonal values and thus suggests no coupling in this domain. These controller gains are calculated by taking the inverse of the estimated 'plant' gains as shown by Equation 6-2 and Equation 6-3. This is the view of the algorithm presented by Webb [1987], Osterhout [1991] and Valjavec [1999].

As suggested by Equation 6-4, algorithm #2 can be viewed in two different but equivalent manners. In Equation 6-4, the geometry parameters used to describe the part are identical to the spatial parameters used in the previous algorithm. However, the multiplication operation has been replaced with a cyclic convolution. The cyclic convolution can be calculated through matrix multiplications if the controller gain matrix (g_c) is replaced with an equivalent full, circulant matrix. A full circulant matrix is shown in the last view of the 'full' equation. This $MN \times MN$ matrix is fully populated by repeating a single $MN \times 1$ column vector of information throughout the matrix. This full circulant matrix thus provides coupling through the non-zero, off-diagonal values [Strang, 1986].

$$d_i = d_{i-1} + g_c \otimes (p_{ref} - p_{i-1})$$

Equation 6-4

Or

$$\begin{bmatrix} {}_1d_i \\ {}_2d_i \\ \vdots \\ {}_{MN}d_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} {}_1d_{i-1} \\ {}_2d_{i-1} \\ \vdots \\ {}_{MN}d_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} {}_1g_c & 0 & 0 & 0 \\ 0 & {}_2g_c & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & {}_{MN}g_c \\ (MN,MN) \end{bmatrix} \otimes \left(\begin{bmatrix} {}_1p_{ref} \\ {}_2p_{ref} \\ \vdots \\ {}_{MN}p_{ref} \\ (MN,1) \end{bmatrix} - \begin{bmatrix} {}_1p_{i-1} \\ {}_2p_{i-1} \\ \vdots \\ {}_{MN}p_{i-1} \\ (MN,1) \end{bmatrix} \right)$$

Or

$$\begin{bmatrix} {}_1d_i \\ {}_2d_i \\ \vdots \\ {}_{MN}d_i \\ (MN,1) \end{bmatrix} = \begin{bmatrix} {}_1d_{i-1} \\ {}_2d_{i-1} \\ \vdots \\ {}_{MN}d_{i-1} \\ (MN,1) \end{bmatrix} + \begin{bmatrix} {}_1g_c & {}_{MN}g_c & \vdots & {}_2g_c \\ {}_2g_c & {}_1g_c & {}_{MN}g_c & {}_{MN}g_c \\ \vdots & {}_2g_c & {}_1g_c & \vdots \\ {}_{MN}g_c & \vdots & {}_2g_c & {}_1g_c \\ (MN,MN) \end{bmatrix} \otimes \left(\begin{bmatrix} {}_1p_{ref} \\ {}_2p_{ref} \\ \vdots \\ {}_{MN}p_{ref} \\ (MN,1) \end{bmatrix} - \begin{bmatrix} {}_1p_{i-1} \\ {}_2p_{i-1} \\ \vdots \\ {}_{MN}p_{i-1} \\ (MN,1) \end{bmatrix} \right)$$

$$g_c = \mathfrak{Z}^{-1}(\hat{G}_P^{-1}) = \mathfrak{Z}^{-1}\left(\frac{\mathfrak{Z}(d_2 - d_1)}{\mathfrak{Z}(p_2 - p_1)}\right)$$

Equation 6-5

Linear Systems Background

A review of some linear system principles lays the foundation for investigations into the system identification and coupling of algorithm #2. Consider the linear, continuous time system of Figure 6-2. Here, $u(t)$ represents the system input and $y(t)$ is the system output.

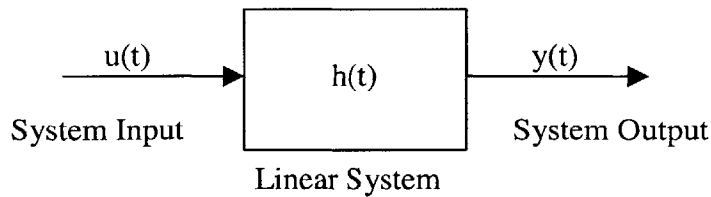


Figure 6-2: Linear, continuous-time system

Now consider the arbitrary input to the system as shown in Figure 6-3. Here the input is represented as a smooth, continuous-time function ($u(t)$) as well as a staircase function of a (N) piecewise constant sections of fixed duration T (denoted $u(nT)$).

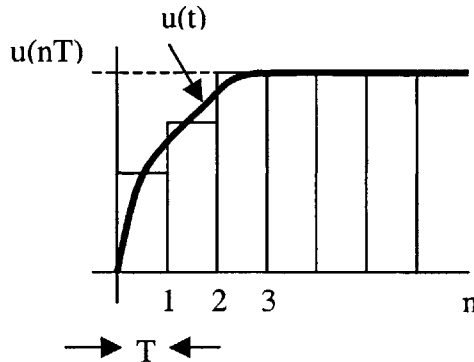


Figure 6-3: System input

It is clear that the representations of the input are identical in the limit, as the period T approaches zero. Through the staircase representation, the same input is viewed in

discrete time as a sum of non-overlapping pulses with duration T and each with a specific time delay (nT) into the system. Thus, the continuous time system of Figure 6-2 can be represented by the discrete time system shown in Figure 6-4.

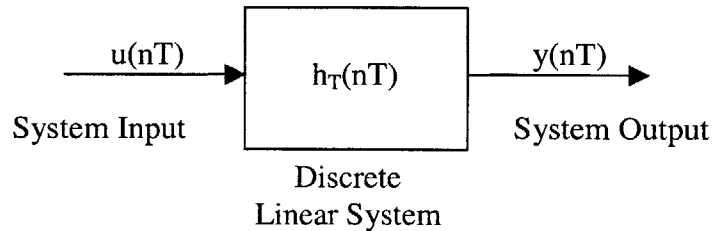


Figure 6-4: Discrete time system

The Superposition principle of linear systems states that the total system response to the staircase input can be written as the sum of the responses to the individual pulses that comprise the input. This is expressed in Equation 6-6 where h_T is the response to the unit impulse.

$$y(nT) = \sum_{k=0}^{N-1} u(kT)h_T[(n-k)T]$$

Equation 6-6

Where:

$y(nT)$	= discrete time system output
$u(nT)$	= discrete time system input
$h_T(nT)$	= discrete time impulse response
N	= # of points in system input, output and impulse response
T	= time duration of each point in discrete system
n	= counter defining which output point is being computed
k	= counter used during summation

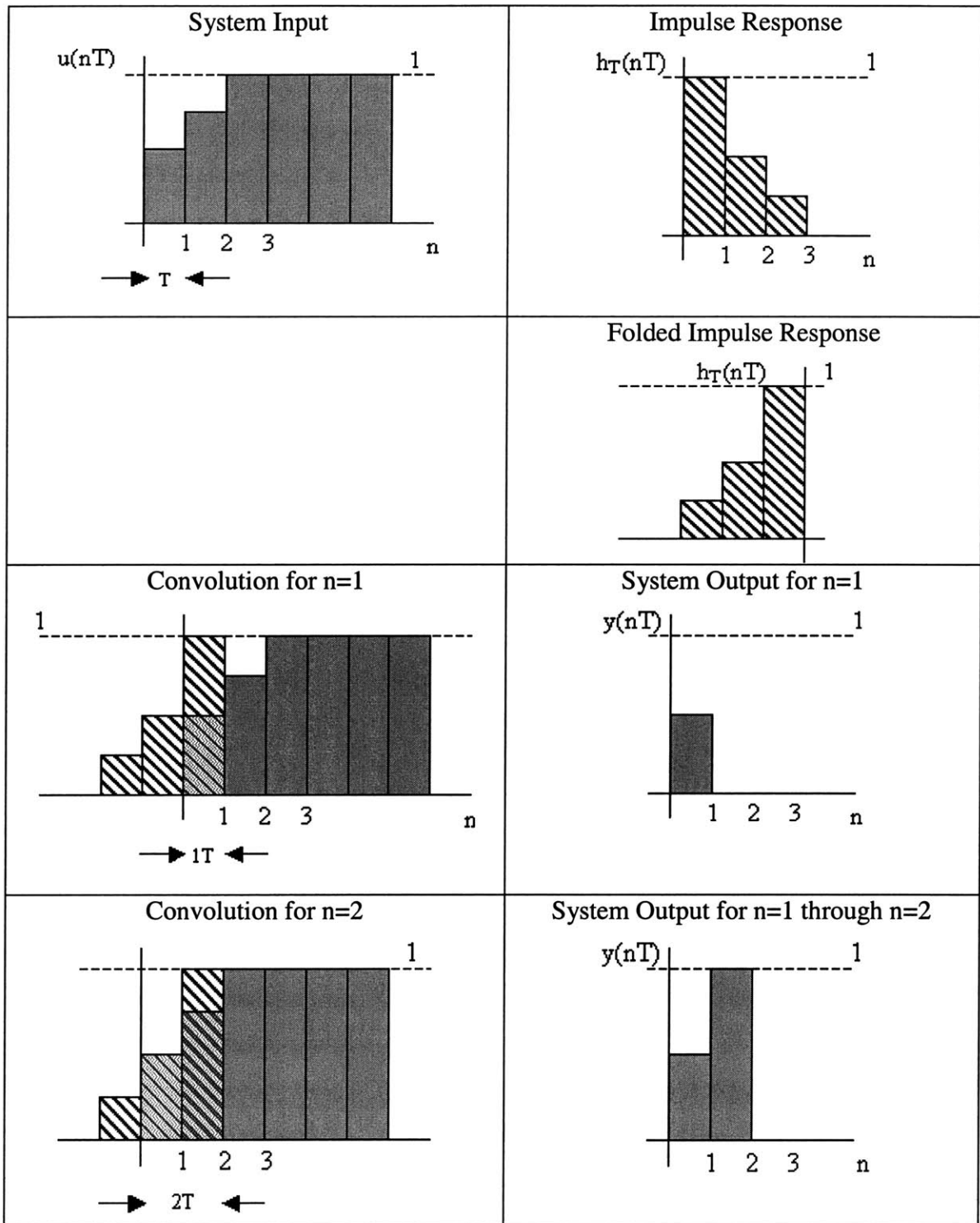
This summation suggests the response for a time less than zero will be zero and that future components of the input cannot contribute to the sum. In other words, time responses must be causal; the output cannot anticipate the input.

The operation defined in Equation 6-6 is alternately known as a discrete convolution and can be represented more conveniently by the \otimes symbol shown in Equation 6-7. These summations state that a system can be entirely characterized by its response to an impulse function.

$$y(nT) = \sum_{k=0}^{N-1} u(kT)h_T[(n-k)T] = u \otimes h$$

Equation 6-7

It is helpful to view the process of convolution graphically, as is shown in Figure 6-5. The process begins with a known impulse response and a system input for which the corresponding system output is desired. To perform the convolution and compute the output, the impulse response of the system is first folded about the time=0 axis and then laid over the system input. The reversed impulse response is then shifted to the right until its beginning point (now the point of the response that is furthest to the right) is located at time t_1 ($t=1$). The folded impulse response is then multiplied by the system input and the total product is summed. This summation represents the system output value at the corresponding time. The process of shifting, multiplying and summing is repeated. The resulting values represent the total system response.



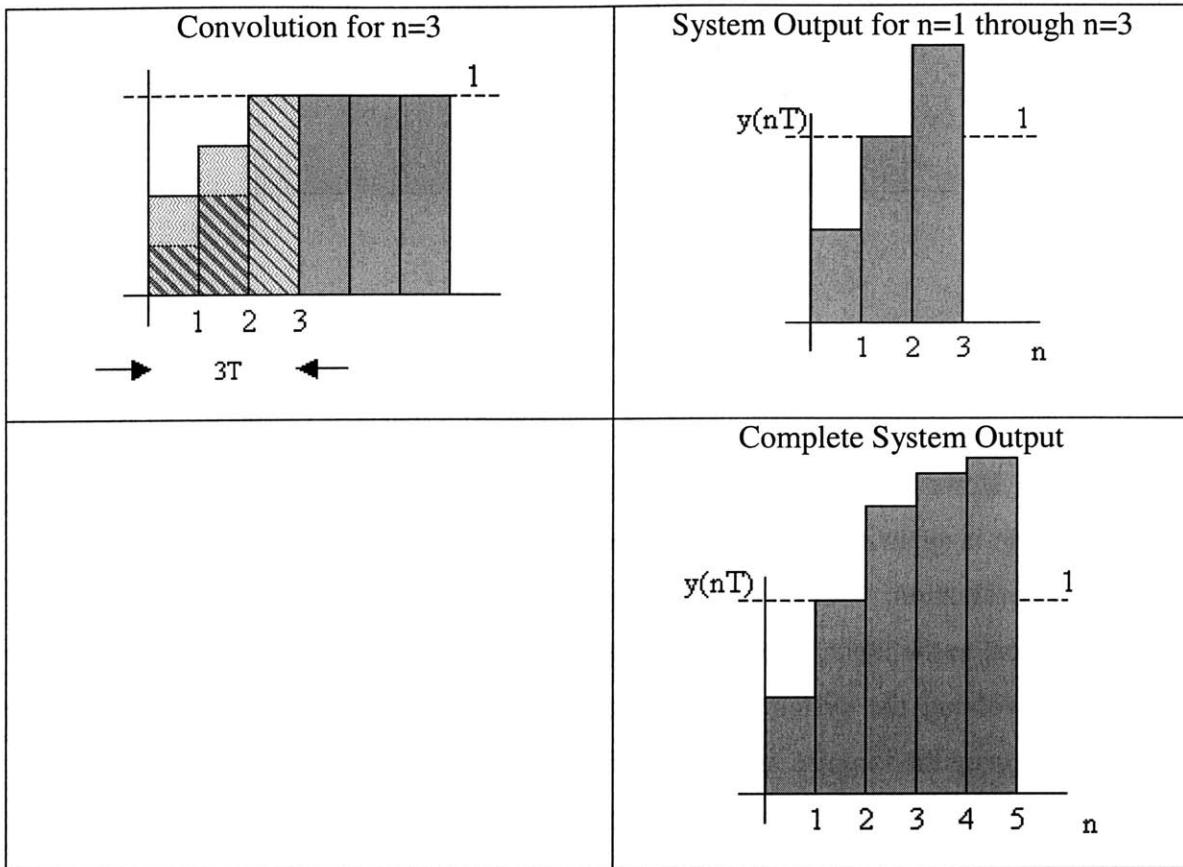


Figure 6-5: graphical depiction of convolution

Periodic System

The parts and dies used in the algorithm are of identical finite length and width. The impulse response used by the algorithm is also the size of die and part. However, the convolution produces an output that is twice the number of finite units comprising die or part minus one. This presents a problem, as the algorithm requires input and output of identical size. To handle this all dies, parts, and impulse responses are treated as cyclic events. This means that each shape or response is treated as though it repeats at a period equal to the size of the part / die. This is shown graphically in Figure 6-6.

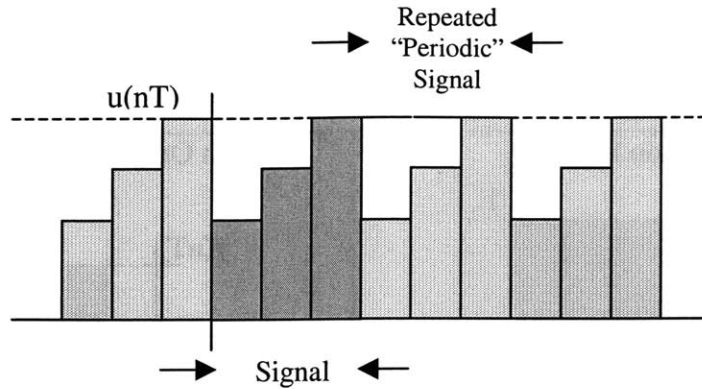
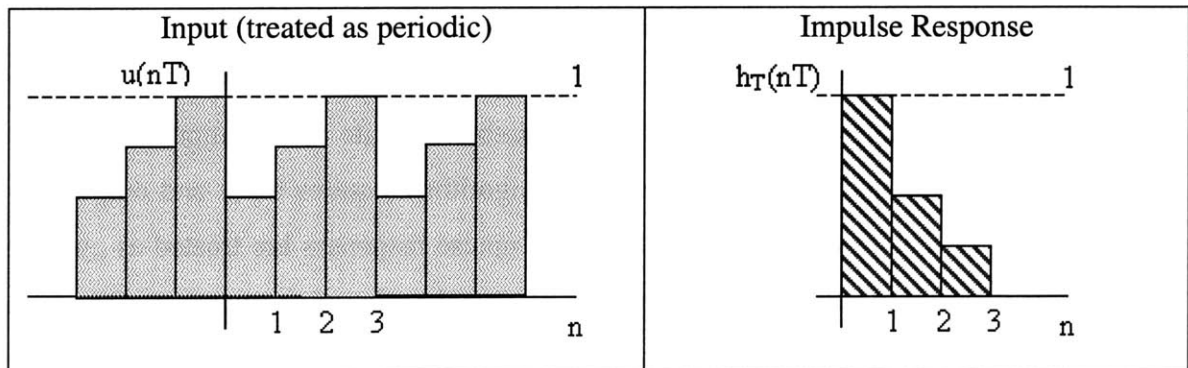


Figure 6-6: Assumption of periodicity

The process of convolution for a periodic system is similar to that of the standard convolution. However, the cyclic nature allows us to make some shortcuts. The process of convolution is again shown graphically in Figure 6-7, but this time a cyclical input is used in the evaluation. The output of the system also displays a cyclic response of a period identical to the input. The system output can thus be characterized entirely without actually convolving the system response over a stream of repeated, identical system inputs. Wrapping the impulse response of the system around the system input during the convolution performs this shortcut. This is shown graphically in Figure 6-8, and the result is the same cyclic response found if a standard convolution is performed on a stream of repeated system inputs.



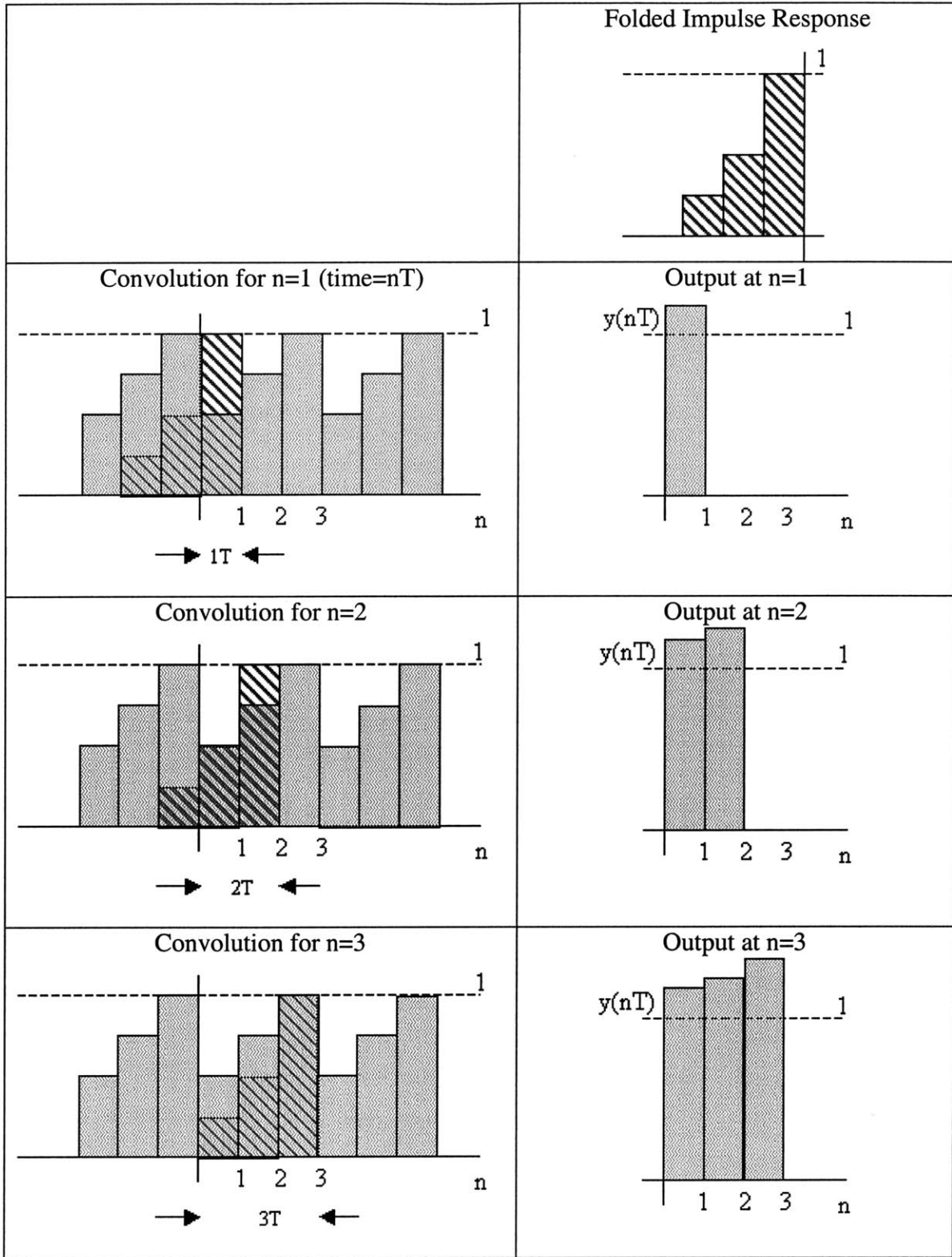


Figure 6-7: Convolution of a cyclic input

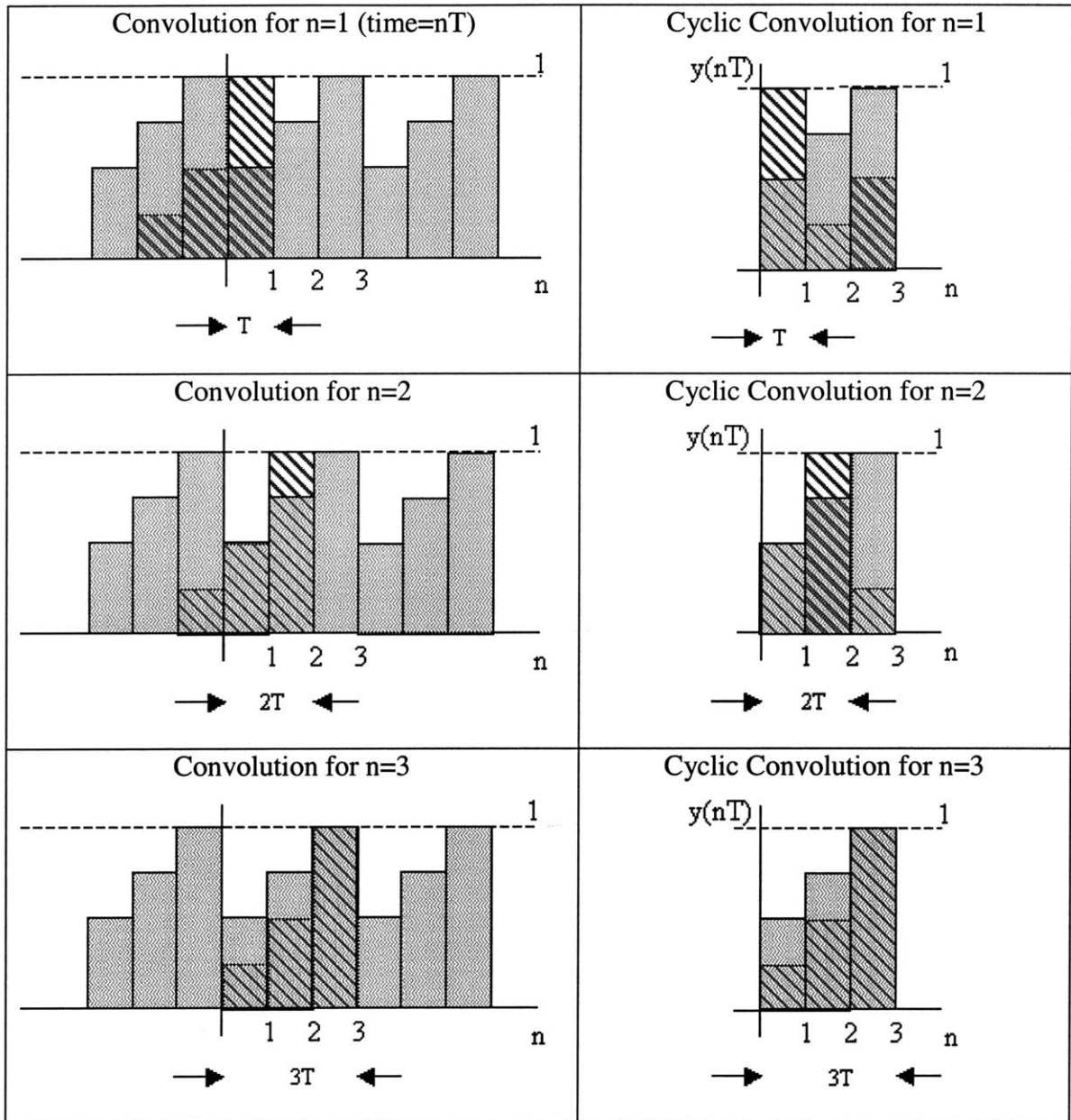


Figure 6-8: Cyclic convolution

Time based system vs. Spatial based system

Through the idea of the convolution, linear system theory provides a means to introduce coupling into the algorithm. Linear systems are most typically applied when the independent variable is time. However, the system here is spatial. This changes little

except that spatial systems can have a-causal responses, where time based systems cannot. In other words, a time system cannot anticipate the outcome of an event that will occur at a later time. However, the output of a spatial system at one point can be affected by what occurs at points to either side. The cyclic convolution creates an avenue to allow a-causal responses into the system model. Examples of both types of systems are provided in Figure 6-9.

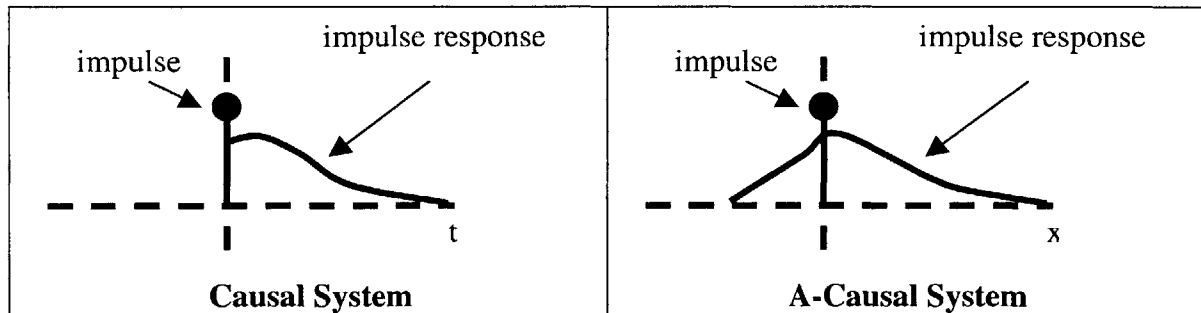


Figure 6-9: Causal and A-Causal Systems

It should be noted that an a-causal response is implicitly introduced to a system when it is treated as periodic. The final value of an impulse response is effectively next to the first value in the periodic assumption. Periodic impulse responses are often centered about zero for easier viewing. An example of this is shown in Figure 6-10.

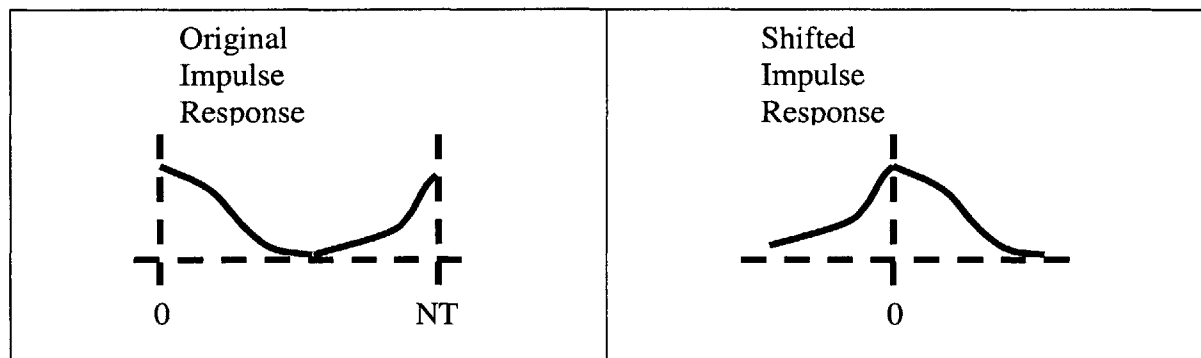


Figure 6-10: Centering a 1D Impulse response for easier viewing

2-Dimensional Convolution

We now have an explanation of the convolution process in the context of a one-dimensional, cyclic, spatial system. The parts and dies in the stretch forming process exist as functions of two, independent variables as show in Figure 6-11. We thus need to extend the convolution to handle cyclic, two-dimensional, spatial systems.

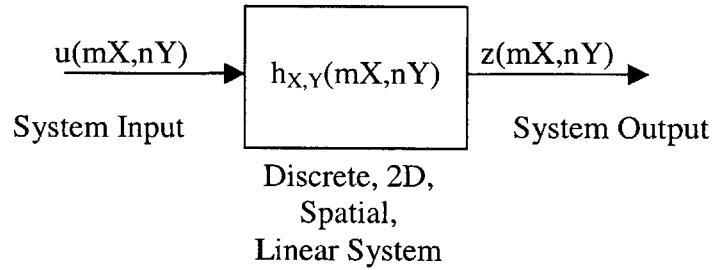


Figure 6-11: Two-dimensional, cyclic, spatial system

The processing is similar to one-dimensional systems. Again, the response of the system to a unit impulse is needed. When dealing with two-dimensional systems, this impulse response is often called a Point Spread Function (PSF) [Karu, 1999]. Once the PSF is known, it is then folded across both of its own axes. The folded PSF is then placed such that its origin is coincident with the origin of the input. Any portion of the PSF not directly over a corresponding portion of the input are then wrapped around the edges of the input. The values of the folded PSF and the system are then multiplied together. The sum of these multiplications constitutes the system output for this position ($m=1,n=1$). The folded PSF is then shifted in one direction such that the origin of the folded PSF now lies over position ($m=1,n=2$) of the system input. The product of the two signals is computed and then summed to define the value of the output signal at position ($m=1,n=2$). This process is repeated until the system output is defined for all 'M' positions in the x-direction and all 'N' positions in the y-direction. This process is represented by Equation 6-8.

$$z(mX, nY) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} u(iX, jY) h_{\tau}[(m-i)X, n-j)Y] = u \otimes h$$

Equation 6-8

Where:

<i>X</i>	= distance between points in the X direction
<i>Y</i>	= distance between points in the Y direction
<i>M</i>	= number of points in the x direction
<i>N</i>	= number of points in the y direction
<i>m</i>	= counter in the x direction
<i>n</i>	= counter in the y direction
<i>i</i>	= counter in the x direction
<i>j</i>	= counter in the y direction

Viewing the Point Spread Function

Viewing the PSF for 2-dimensional, periodic systems can be confusing. The point at the origin of the 2-dimensional PSF represents the contribution of a point over which it is placed during the convolution process. The points adjacent to the origin represent the contributions that those adjacent points will have during the convolution. However, for periodic systems the points on opposite corners are effectively adjacent to the point at the origin. Points on any edge are adjacent to the points on opposing edges as well. It is difficult to visually connect all of the boundaries of the PSF plot. It is difficult to view how a PSF represent the system response as it is challenging to visually connect all of the boundaries of the PSF plot. To address this the origin, which typically has the largest value of the PSF, is shifted to the center for viewing purposes alone. An example of this is shown in Figure 6-12. A schematic depicting the shifting process is shown in Figure 6-13.

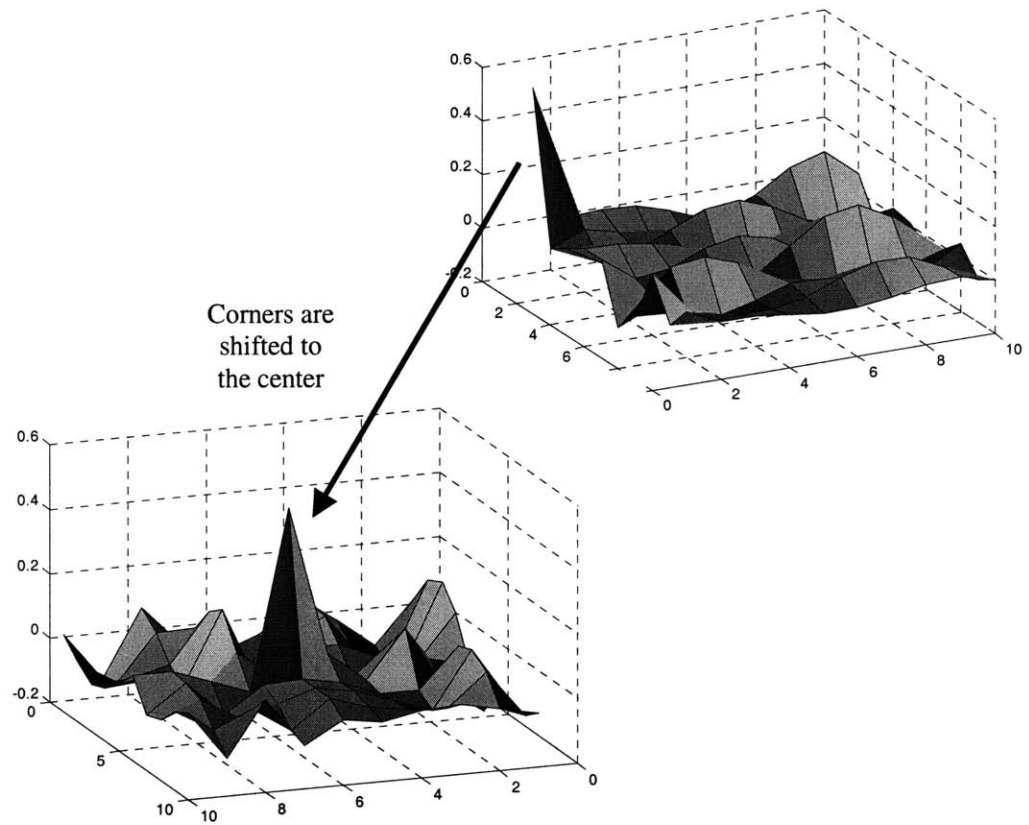


Figure 6-12: Shifting of the 2-D, periodic PSF to assist viewing

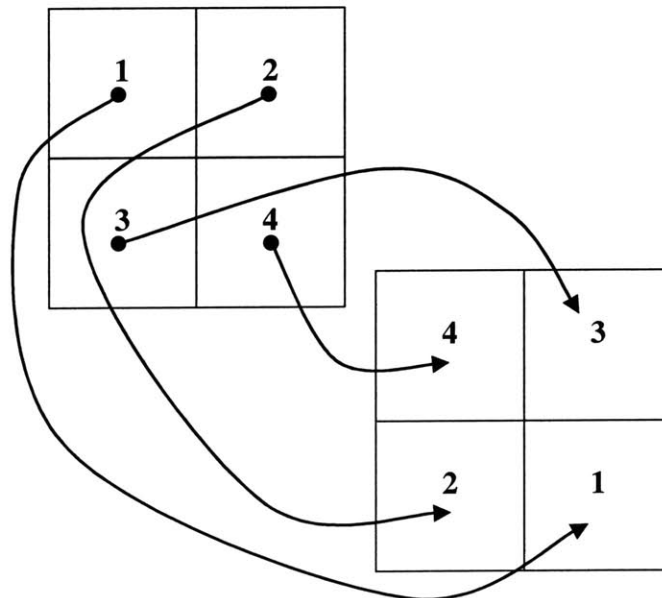


Figure 6-13: Schematic of 2-D, PSF shifting

Quantifying Coupling

The system coupling represented in a given PSF is easily seen once the PSF has been shifted. The center point of the PSF represents the output contribution made by at the same point in time (space) as the input. The magnitudes of adjacent points represent their respective contributions to the system output. The PSF's of various systems are shown in Figure 6-14.

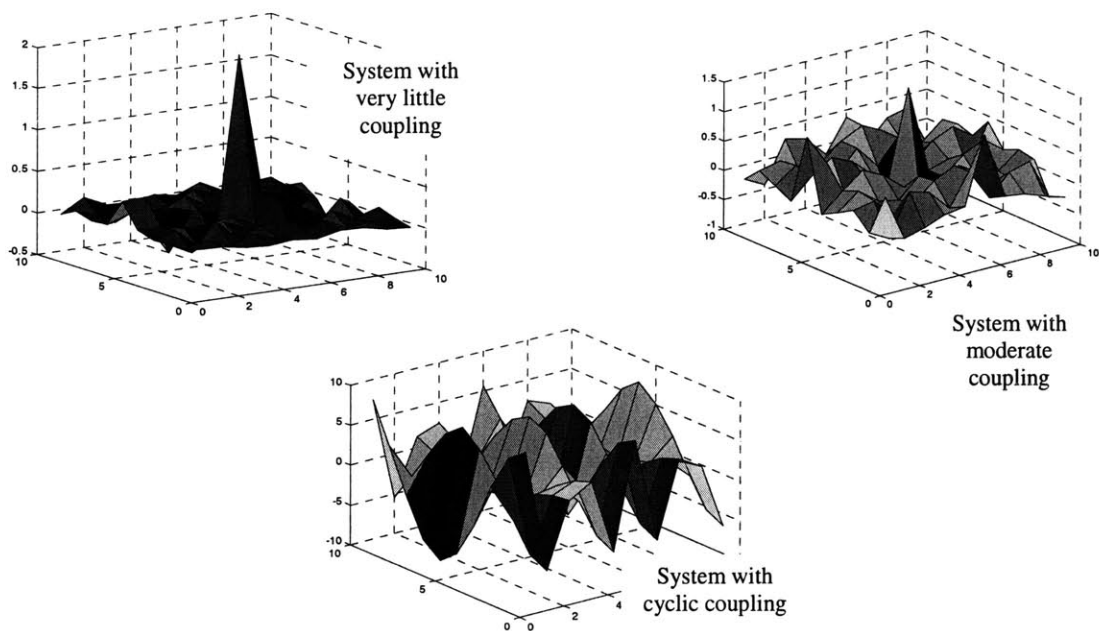


Figure 6-14: PSF functions displaying coupling

Benefits of the Fourier Transform

Historically, linear system computation techniques have been performed through convolutions, although a simpler technique does exist. In this simpler method, system inputs and impulse response or PSF functions are converted to the frequency domain through the Fourier transform. Properties of the Fourier transform, primarily the superposition principle (Equation 6-9) and the convolution identity (Equation 6-10) allow for computationally simpler work once signals are in their frequency domain representation. Through the convolution identity, we can see that the computationally intensive process of convolution is replaced by simple multiplication.

$$af(x, y) + bh(x, y) \overset{\S}{\leftrightarrow} aF(u, v) + bH(u, v)$$

Equation 6-9

Where:

a & b	= constants
f & h	= 2 dimensional signals
x & y	= spatial, x & y coordinates
F & H	= frequency domain representations of f & h
u & v	= frequencies of F & H

$$f(x, y) \otimes h(x, y) \overset{\S}{\leftrightarrow} F(u, v)H(u, v)$$

Equation 6-10

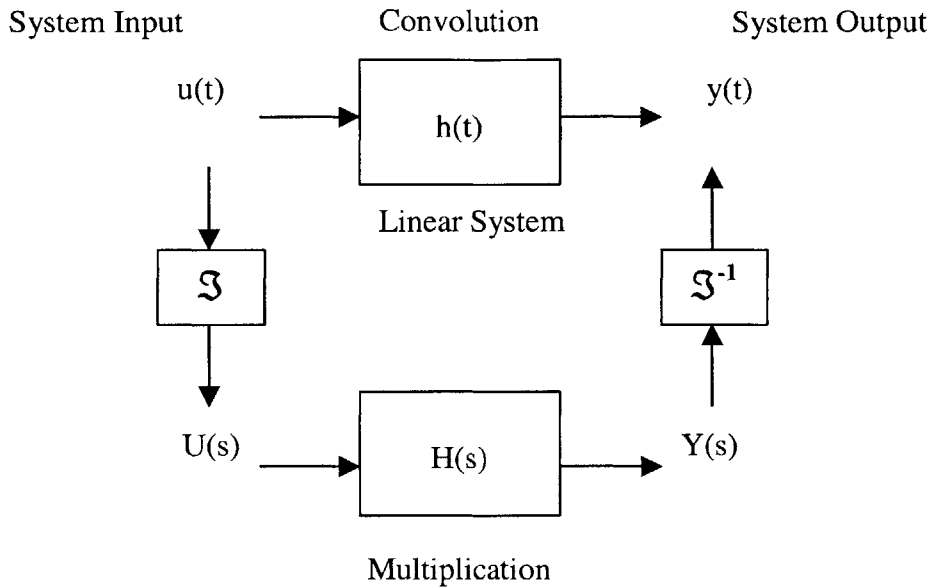


Figure 6-15: Linear system computation methods

The equivalent methods for computing the system output are shown in Figure 6-15. Today, approaches that compute the convolutions directly are seldom used. Prior to the 1960's, computing the Fourier transform of a signal and the inverse Fourier transform of a signal cost more computationally than performing a convolution. However, the advent of the fast Fourier transform by Cooley and Tukey in the 1960's drastically reduced the computation associated with the Fourier transform. This made the frequency domain approaches more expedient than direct convolution methods.

System Identification / De-Convolution

System identification is the process of estimating a system $h(t)$ when given its input $u(t)$ and output $y(t)$. The Fourier transform is extremely useful in performing this task. Once the input and output of a system are transformed into the frequency domain, they can be used to define the system that relates them through Equation 6-11. This frequency domain representation of the system can then be converted back to the spatial domain where it can serve as an estimate of the system impulse response or PSF as suggested by Equation 6-12.

$$H(s) = \frac{Y(s)}{U(s)}$$

Equation 6-11

Where:

s = frequency
 H = frequency domain representation of system identification
 U = frequency domain representation of system input
 Y = frequency domain representation of system output

$$h(t) = \mathfrak{S}^{-1} H(s)$$

Equation 6-12

There is often interest in defining a system that will reproduce the system input $x(t)$ when it is convolved with the system output $y(t)$. Such a system is said to “de-convolve” the output from the system. Placing such a de-convolution operation into our system allows the output to precisely follow the system input as shown in Figure 6-16.

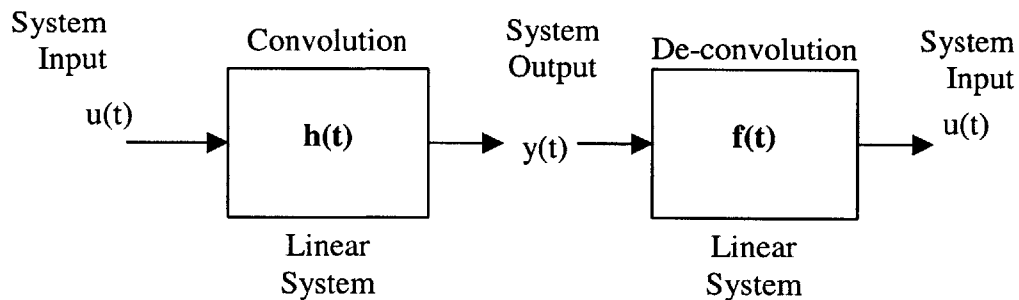


Figure 6-16: Convolution & De-convolution

The same effect is observed if the de-convolution is performed before the system acts on the input, as shown in Figure 6-17.

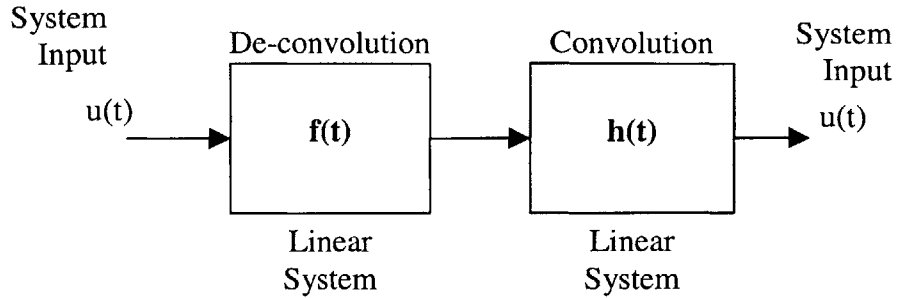


Figure 6-17: De-convolution & convolution

It is noted that convolving any signal with a unit impulse leaves the initial signal intact. Therefore, the convolution of $f(t)$ and $h(t)$ can be equivalently represented by a unit impulse at time=0 as indicated by Equation 6-13. Once again, frequency domain techniques are extremely helpful in finding this new signal $f(t)$. Equation 6-13 is easier to solve when it is viewed in the frequency domain as in Equation 6-14.

$$f(t) \otimes h(t) = \delta(0)$$

Equation 6-13

Where:

$\delta(0)$ = unit impulse at time zero

$$F(s)H(s) = 1$$

Equation 6-14

Equation 6-14 is solved for $F(s)$, represented by Equation 6-15. This $F(s)$ can then be converted back from the frequency domain through an inverse Fourier transform as shown in Equation 6-16.

$$F(s) = 1/H(s)$$

Equation 6-15

$$f(t) = \mathfrak{J}^{-1}F(s)$$

Equation 6-16

Blind De-Convolution

The process of inverting a frequency domain system identification (Equation 6-15) and converting it back to the time or space domain (Equation 6-16) is alternatively known as de-convolution. De-convolution is extremely useful in signal and image processing. It can be used to remove the linear streaking that occurs when an image is captured of an object moving at a known speed. De-convolution can also be used to decrypt messages that have been deliberately convolved with a known encryption signal. However, in both of these cases there is exact knowledge of the system $h(t)$ that initially convolves the signal. This is not the case with most situations, including the algorithms discussed in this research. The technique of identifying an impulse response or PSF to de-convolve a system without exact knowledge of the initial convolving system is known as blind de-convolution [Karu, 1999]. Blind de-convolution handles additive noise poorly. The sharp transitions associated with additive noise are amplified greatly through blind de-convolution [Karu, 1999]. The process of convolution can be viewed as an averaging, or smoothing process. The process of blind de-convolution then performs an opposite ‘sharpening’ process. Any sharp transitions already present in the data will be amplified by the de-convolution ‘sharpening’ process.

Windowing

In signal or image processing, it is often the case that system identification is made on finite data sets from systems that are being excited throughout the duration of the data set. When this is the case, the system response does not always return to its initial value by

the end of the data set. Figure 6-18 shows examples where a system response truncated in this manner.

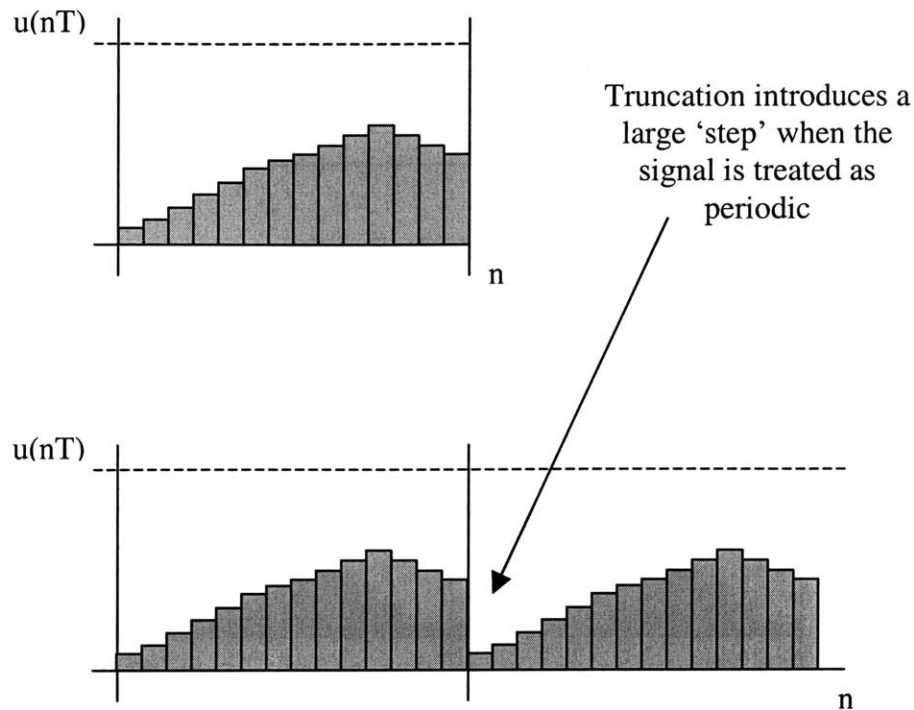


Figure 6-18: Truncated system responses

In order to allow processing, finite data sets are treated as periodic signals. The periodic assumption suggests that the output repeats itself just after the signal terminates. This can introduce sharp transitions between the end and the beginning of each of the signals in Figure 6-18. These sharp transitions are amplified like noise through the deconvolution process. To reduce the effect of this sharp transition, the system outputs are often multiplied by a 'windowing' function. Such windowing functions remove these sharp transitions but also change the actual signals being used. An example of a windowing function is shown in Figure 6-19. Figure 6-20 displays the results of applying this windowing function to the signals in Figure 6-18.

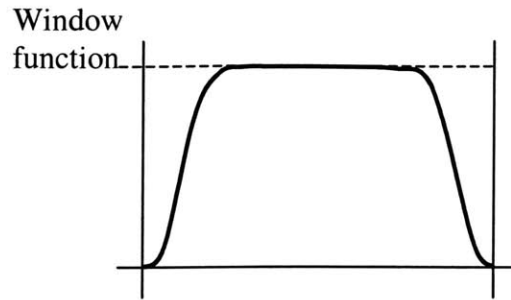


Figure 6-19: Windowing Function

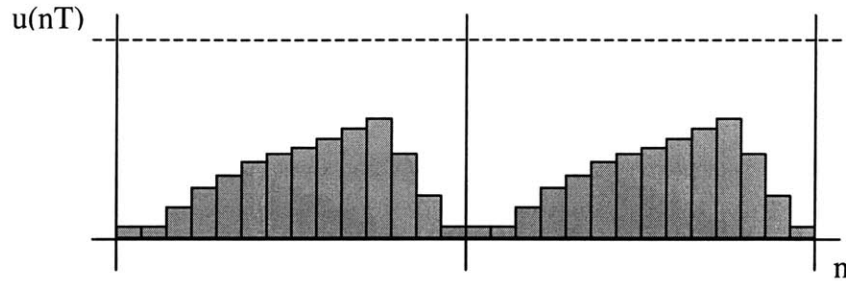


Figure 6-20: Windowed Functions

Implementation

In practice, the first step to implementing the DTF algorithm is to define the best guess at the target die geometry. Once this is complete, the calibration dies are chosen. Several different approaches have been used to define the calibration dies. Valjavec [1999] utilized a backward difference approach that ‘brackets’ the first two dies around the best guess at the target die shape. The calibration dies are then defined by multiplying the target die shape by constant factor (like 5%) which is reflected in Equation 6-17.

$$d_1 = 1.05(\hat{d}_*)$$

$$d_2 = 0.95(\hat{d}_*)$$

Equation 6-17

Where:

$$\hat{d}_* \quad = \text{initial estimate of springback compensated die}$$

$$d_1 \quad = \text{die used to form first part}$$

$$d_2 \quad = \text{die used to form second part}$$

In other implementations, and in the tests performed within this research the first die shape is set to the best guess target die shape. The second die shape is then identified by applying algorithm #1, as shown by Equation 6-18.

$$d_1 = d_*$$

$$d_2 = d_1 + p_{ref} - p_1$$

Equation 6-18

Parts are then formed and the resulting components are measured and registered in the normal manner. The part and die representations are then placed into Equation 6-19 to form an estimate of the PSF that defines how parts respond to a unit impulse in the die. It should be noted that several methods to smooth the part representations are sometimes used before Equation 6-19 is applied. Such smoothing techniques include windowing and Welch's method⁵.

⁵ Welch's method is a technique used to smooth the periodogram representation of a signal

$$\hat{g}_P = \mathfrak{F}^{-1}\left(\frac{\mathfrak{F}(p_2 - p_1)}{\mathfrak{F}(d_2 - d_1)}\right)$$

Equation 6-19

Where:

$$\hat{g}_P \quad = \text{estimate of the system PSF}$$

The estimated system PSF is then inverted in the frequency domain and converted back to the spatial domain to identify a PSF for the controller (g_C). This is the process of de-convolution, which results in a new PSF that will allow part errors to be compensated for. Equation 6-20 represents the process of blind de-convolution. It is critical to note that the controller PSF is not simply the inverse of the system PSF. This is emphasized by Equation 6-21.

$$g_C = \mathfrak{F}^{-1}\left(\frac{\mathfrak{F}(d_2 - d_1)}{\mathfrak{F}(p_2 - p_1)}\right)$$

Equation 6-20

$$g_C = \mathfrak{F}^{-1}(\hat{G}_P^{-1}) \neq g_P^{-1}$$

Equation 6-21

Equation 6-20 provides the frequency domain controller gains (G_C), or equivalently, the de-convolving PSF (g_C) that are used by Equation 6-1 or Equation 6-4, which govern Algorithm #2. Valjavec [1999] has determined that using the same (g_C) value for all subsequent iterations of algorithm #2 reduces its sensitivity to variation.

Historical Experimental Results

Valjavec [1999] performed experiments of algorithm #2 on MIT's lab scale stretch forming press. The common parameters used in these experiments are detailed in Figure 6-21.

Parameter	Value
Pin size	0.5 inch
Stretch Force	6125 lbs.
Blank Material	Al 2024-O, 19.5 x 5.5 x 0.063 inches
Final part footprint	5.0 x 5.0 inches
Force trajectory	Pre-stretch, wrap, no post stretch
Control mode	Force control
Interpolator	Elvax 360 (0.535 in. thick) covered with two layers of Teflon

Figure 6-21: Common Parameters for Valjavec's [1999] tests of algorithm #2

Experiments were carried out on four different target part geometries. These include a cylinder, a sphere, a saddle, and an ellipsoid. Plots of these target shapes are shown in Figure 6-22.

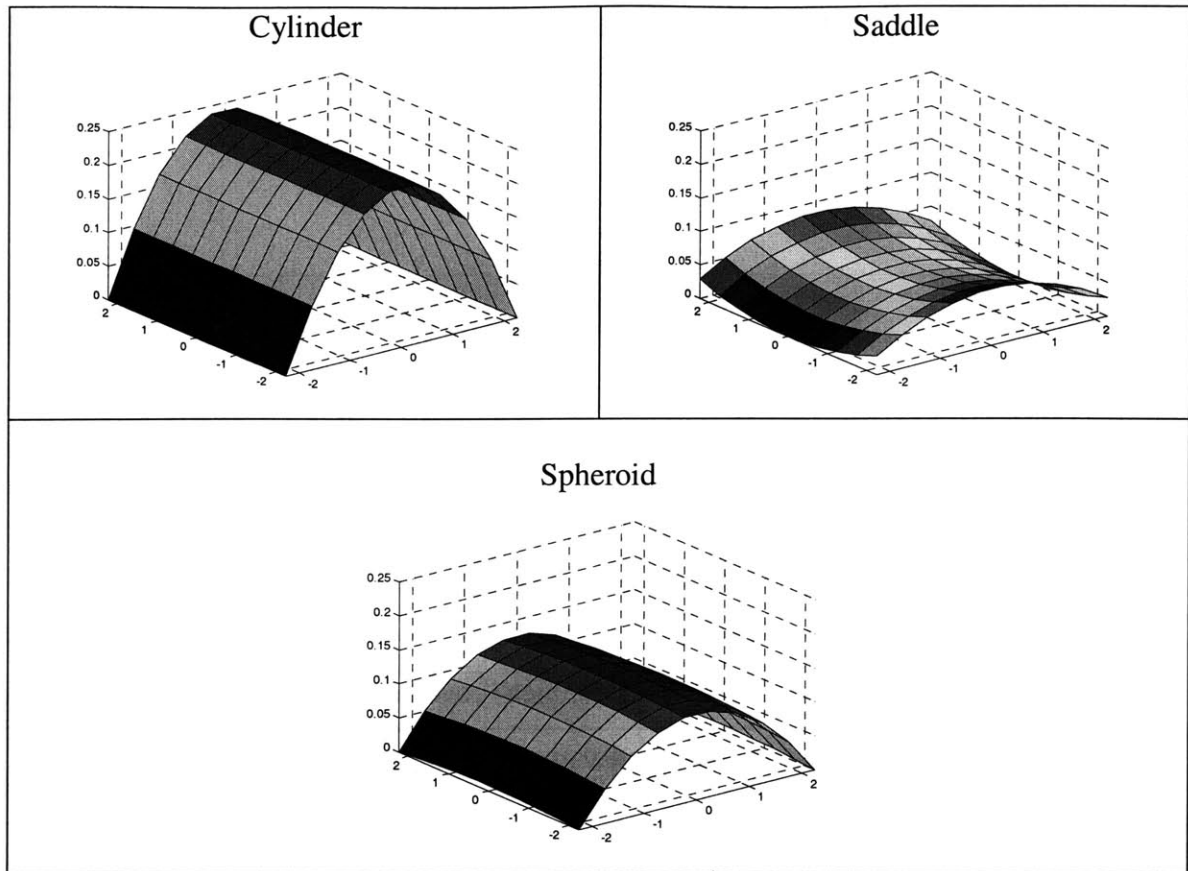


Figure 6-22: Reference shapes for Valjavec's [1999] experiments

In these experiments, the initial die shape is set equal to the target part shape and then adjusted for the ½ inch thick interpolator layer. Algorithm #1 is used to identify the second die shape. Algorithm #2 is then used to define the geometry of the third and fourth die. The results of the experiments are shown in Figure 6-23.

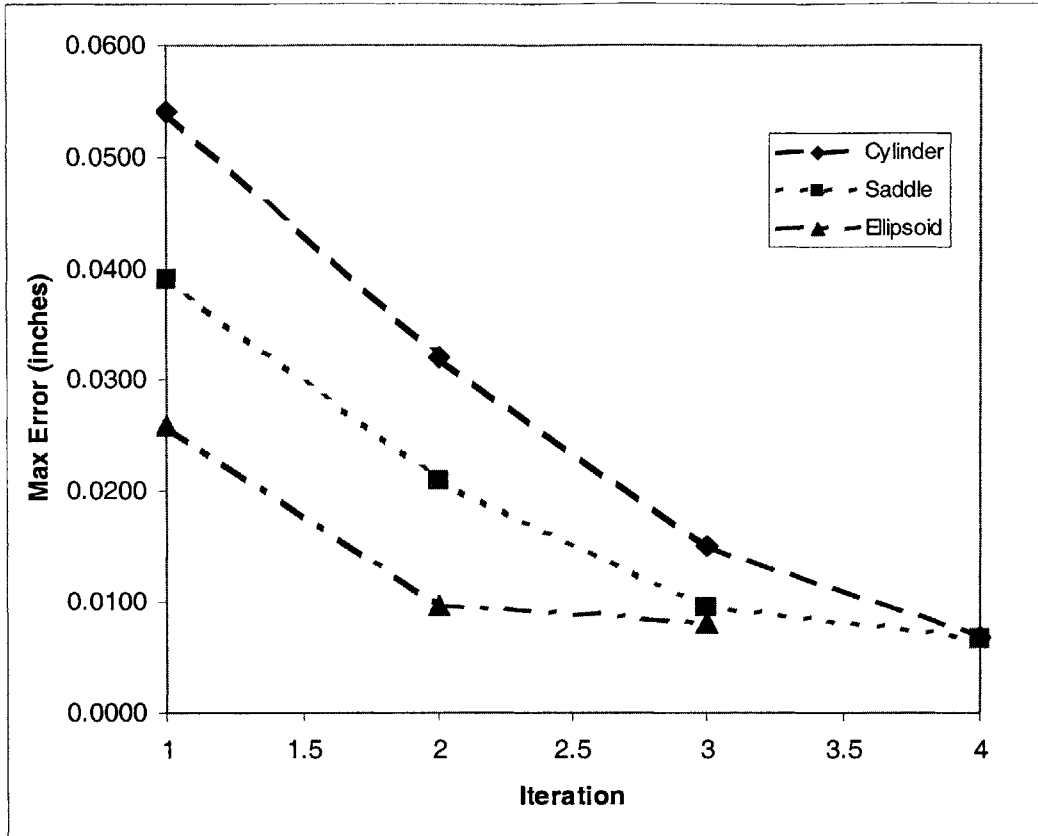


Figure 6-23: Valjavec's [1999] experimental results

It is noted that experiments were carried out until the maximum error fell below the 0.0100 inch level. This was defined as an acceptable error level for this experiment. However, the sphere experiment was terminated because it failed to converge. This shape was later determined incapable of being formed on the MIT lab scale machine.

Algorithm Concerns

The results detailed in Figure 6-23 are an improvement over results from similar tests performed with algorithm #1. However, experiences in the production scale environment have not been able to replicate this performance. Northrop Grumman has performed a limited number of forming trials on a production scale reconfigurable tool. The die shapes predicted by data collected in these trials have been flawed. Excessive variation levels are believed to have caused these problems.

Superposition

Linear system theory requires the system being modeled to obey the superposition principle. Superposition states that the system input can be modeled as a set of offset, scaled impulses. These impulses each produce a response defined by the impulse response or PSF of the system. The output of the system will be the sum of the responses to each delayed and scaled input. In the context of stretch forming on a discrete die, this implies that each pin, when raised alone, will cause a response that is a linear function of the system PSF.

However, the interpolating layer used in stretch forming is known to introduce non-linearity into the system through a region of dead band. This dead band means that a system input has to reach a certain level before it is recognized in the output. Experiments performed by Valjavec [1999] on MIT's stretch-forming press show that the affect of raising a single pin is not apparent in system output until it is raised above 0.01 inches. However, Valjavec's [1999] experiment also shows changes in the displacement of the part above the moved pin is roughly linear once the dead band threshold has been passed. Experiments have not been performed to identify the shape of the responses seen from raising a single pin by different amounts. Superposition implies that the responses should have similar shapes, regardless of how much an individual pin is raised, or at what portion of the part it is raised in.

System Identification

There are some general difficulties associated with system identification. The accuracy of the system identification depends on several variables, including the linearity of the relationship between the die and part, how close the calibration dies are to the target die, and how distant the calibration dies are from one another.

Figure 6-24 displays a case where the calibration dies are spread far apart in a region where the relationship between part and die display non-linearity. In this case, the system identification will predict a die shape that is less accurate than one made in a more linear range.

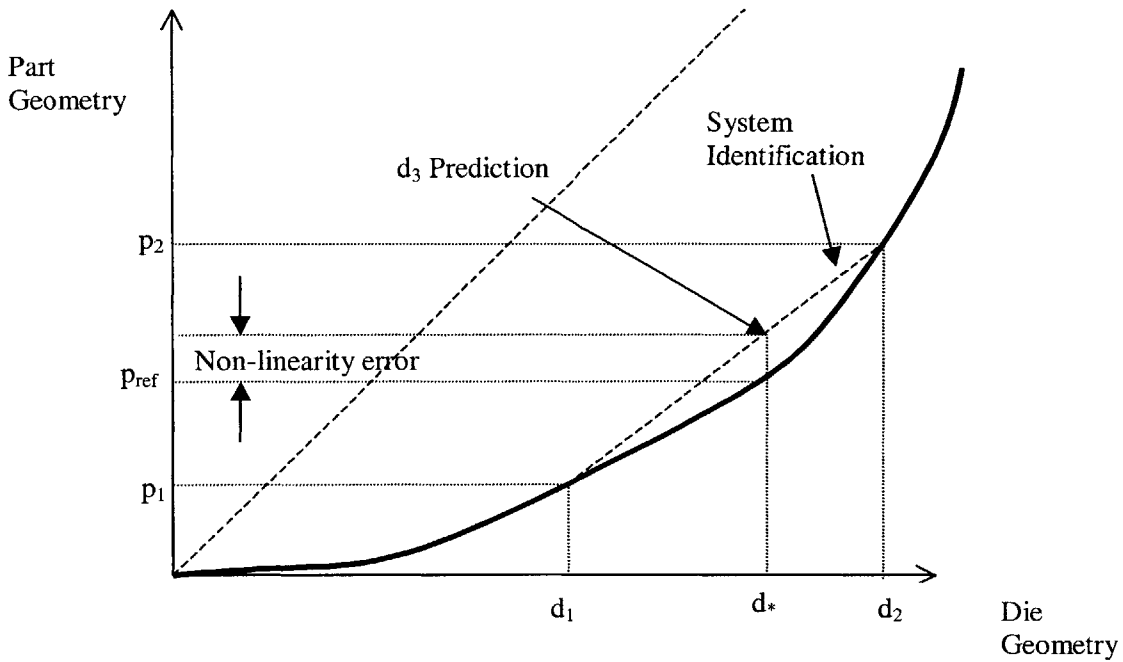


Figure 6-24: system identification performed with calibration die shapes far apart

Figure 6-25 depicts a system identification performed on the same system but with calibration dies that are closer to the target die shape. Error bars have been added to the figure to represent variation in the part forming and measuring process. The dotted lines represent the different system identifications that could result from the same system. It is apparent that the closer the calibration dies, the greater the effect noise will have on the accuracy of the system identification and subsequent die predictions.

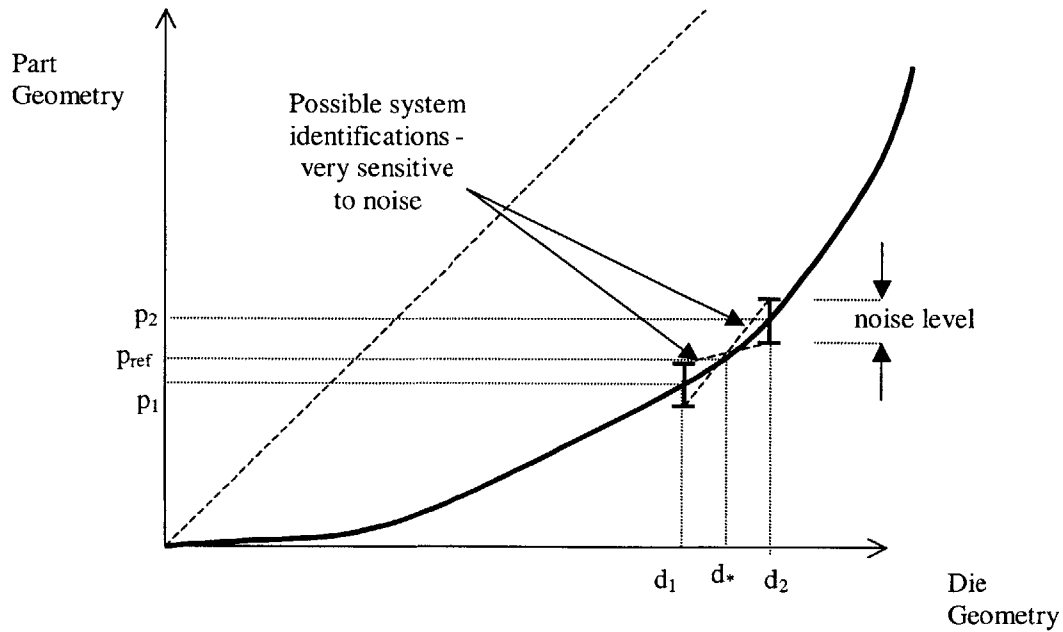


Figure 6-25: System identification performed with close calibration dies and noise

In the absence of variation, calibration dies closer to each other and to the target die will produce a better system identification. However, the presence of variation can contaminate die guesses that are close to one another. Thus there exists trade-offs in choosing the best location for calibration dies. One of these tradeoffs is the adjustment of the spread between the two initial calibration die shapes. Another tradeoff is the relative position of these calibration dies to the apparent location of the target die. Research performed by Marko Valjavek [1999] has shown that the optimal choice is calibration dies centered about the apparent location of the correct die shape. These calibration dies are offset from the apparent location of the correct die shape by 5% or 10% in either direction.

Summary

This chapter has presented algorithm #2 in the context of the frameworks presented in chapter 4. New insights are provided into algorithm #2 by viewing its operations as convolutions in the spatial domain instead of multiplications in the frequency domain. These insights allow the system coupling modeled by algorithm #2 to be quantified and thus quality checked. Algorithm #2 is shown through historical results to have improved

on the performance of algorithm #1. However, it also increases the complexity of the algorithm and its sensitivity to noise.

Chapter 7 : Presentation and Discussion of Algorithm #3

In this chapter, a new algorithm is presented that uses system identification in a less complex manner than algorithm #2. This algorithm is placed into the frameworks presented in chapter 4 where it is analyzed. It is shown to have sensitivity to how parts and dies are referenced. Neither algorithm #1 nor #2 have this concern. However, a heuristic is developed that eliminates this issue. In summary, algorithm #3 addresses the concerns of algorithm #1 in a simpler manner than algorithm #2 without sacrificing performance.

Basic Structure

Algorithm #2 introduces the concept of system identification and system coupling to improve the convergence of algorithm #1. It also adds a great deal of complexity to the algorithm and increases the sensitivity to variation. The difficulties that high variation levels present to the process of blind de-convolution are believed to be responsible for problems experienced during implementation of algorithm #2 into production. For these reasons, algorithm #3 was developed to have the benefits of system identification without the difficulties associated with convolution and de-convolution. Algorithm #3 is charted along with algorithms #1 and #2 in Figure 7-1 where the differences can be noted. The basic control schematic of algorithm #3 is also shown in Figure 7-2.

No Coupling	<u>Algorithm #1</u>	<u>Algorithm #3</u>
	- No System ID - No Coupling	- System ID - No Coupling
Coupling		<u>Algorithm #2</u> "Deformation Transfer Function" - System ID - Coupling via Convolution
	No Identification	Identification

Figure 7-1: Shape control algorithms

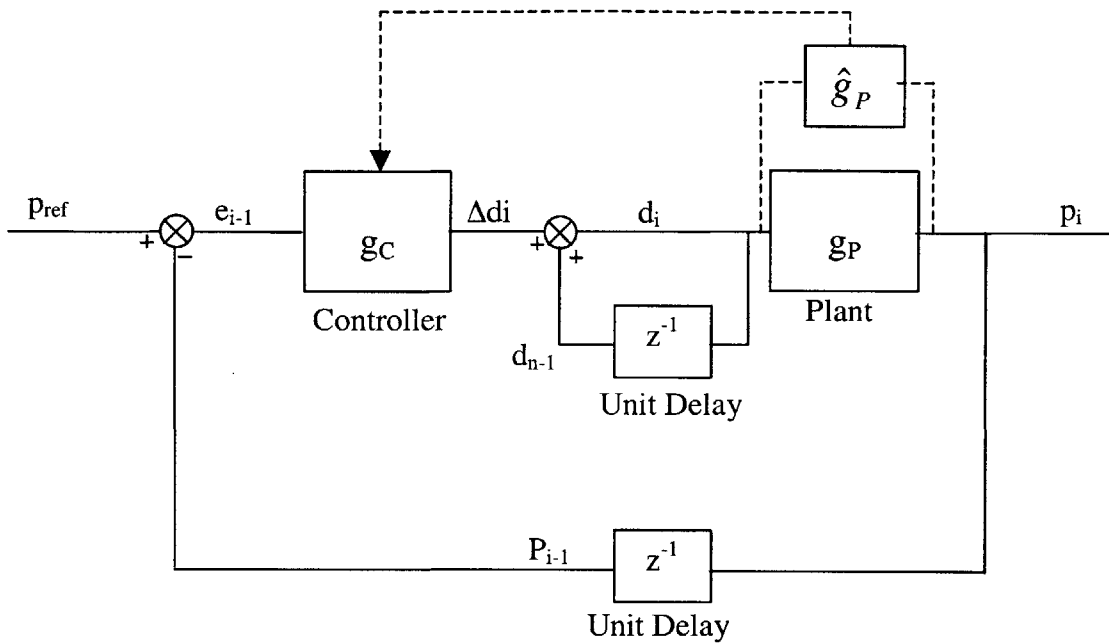


Figure 7-2: Algorithm #3 control schematic
(Note: all variables are spatial coordinates)

Algorithm #3 is governed by Equation 7-1, which has a form identical to the general Equation 4-1. Once again, the full view of the equation is provided to help with understanding. It is critical to note that the off-diagonal values of the gain matrix are all zero. However, each diagonal values each represent a gain value that is identified with Equation 7-2.

$$d_i = d_{i-1} + g_C (p_{ref} - p_{i-1})$$

Equation 7-1

Or

$$\begin{bmatrix} {}_1 d_i \\ {}_2 d_i \\ \vdots \\ {}_{MN} d_i \end{bmatrix}_{(MN,1)} = \begin{bmatrix} {}_1 d_{i-1} \\ {}_2 d_{i-1} \\ \vdots \\ {}_{MN} d_{i-1} \end{bmatrix}_{(MN,1)} + \begin{bmatrix} {}_1 g_C & 0 & 0 & 0 \\ 0 & {}_2 g_C & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & {}_{MN} g_C \end{bmatrix}_{(MN,MN)} \times \left(\begin{bmatrix} {}_1 P_{ref} \\ {}_2 P_{ref} \\ \vdots \\ {}_{MN} P_{ref} \end{bmatrix}_{(MN,1)} - \begin{bmatrix} {}_1 P_{i-1} \\ {}_2 P_{i-1} \\ \vdots \\ {}_{MN} P_{i-1} \end{bmatrix}_{(MN,1)} \right)$$

$$g_c = \frac{d_2 - d_1}{p_2 - p_1}$$

Equation 7-2

Implementation

System Identification

System identification generally refers to the linear system techniques used by algorithm #2. These techniques add coupling through convolution, but they also add significant complexity and sensitivity to variation. The premise behind algorithm #3 is to use a much simpler form of system identification to avoid the complexities and sensitivities of algorithm #2.

Algorithm #3, like algorithm #1 treats each individual pin and the corresponding part that lies directly above it as a SISO system. However, algorithm #3 tunes the gains for each of these SISO systems by evaluating the behavior of the entire system. It views the surfaces of the dies and parts as if they represent a collection of SISO systems. The characteristics of these collective SISO systems are calculated through Equation 7-3 and represented by a gain values (g_p) associated with the forming process

$$g_p = \frac{p_2 - p_1}{d_2 - d_1}$$

Equation 7-3

The controller gains are then calculated by inverting the identified plant gains as shown in Equation 7-4.

$$g_C = g_P^{-1} = \frac{d_2 - d_1}{p_2 - p_1}$$

Equation 7-4

Controller gains (g_C) are set in this manner to compensate for the effects of the stretch forming 'plant' and to allow the system output to follow the system input.

Part "Referencing"

Parts and dies are defined for use in all of the algorithms as a collection of z-axis coordinates on a Cartesian grid. A z-coordinate value helps define a surface by its relation to other z-coordinate values. In a similar fashion, the magnitudes of the z-coordinate values that define a part can be collectively offset by a constant without altering the surfaces they represent. This is shown in Figure 7-3. The average z-coordinates value of parts and dies are of little concern for algorithms #1 or #2. However, algorithm #3 is very sensitive to the average z-coordinate position of dies and parts. For this reason, attention is placed on how a part or die is referenced to the $z=0$ coordinate plane.

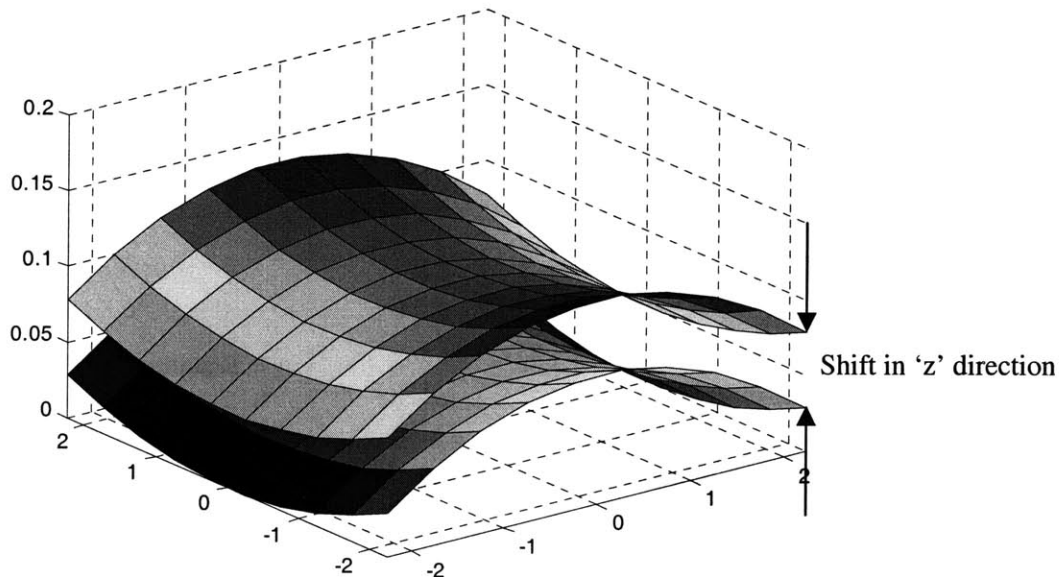


Figure 7-3: Offset z-axis coordinates

Adding a constant to the z-coordinate value of a die poses no problem to the definition of a surface. This is because in practice, only net changes are actually applied to the die. If this were not the case, the trajectories of the stretch forming process would still accommodate the die shift as shown in Figure 7-4. Here the pre-stretch phase of the stretch forming process ends when the die contacts the sheet. This happens at different die table positions when the constant (α) is added to the die. This difference is noted automatically if the press is operated in strain, or force control modes. The shape of the die remains identical if the constant is added. This proves that average z-axis location of the die is not important to the stretch forming process, unless the press is operated in open-loop displacement control mode.

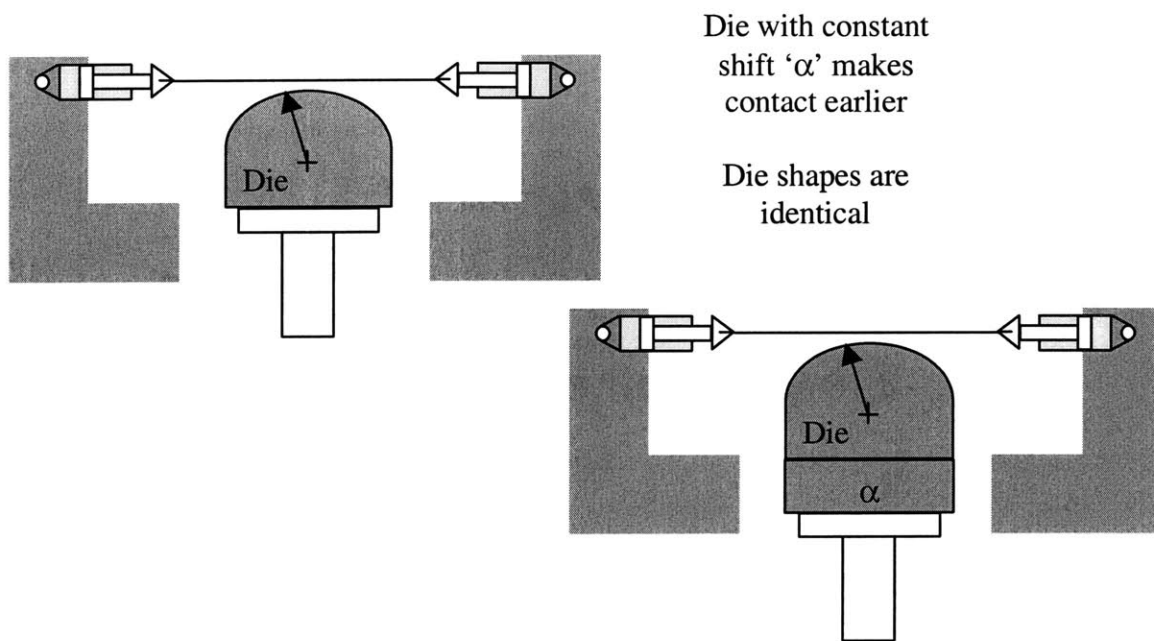


Figure 7-4: Stretch forming die with and without added constant ' α '

Effect of Z-coordinate Shift on Algorithm #1

In the case of algorithm #1, a constant added to every z-axis coordinate will result in the same constant being added to the z position of every pin in the next die. This can be seen in Equation 7-5 where a constant variable (α) is added to the previous formed part in the

governing equation of algorithm #1. This constant is reflected identically in the die change (Δd_i).

$$\Delta d_i + \alpha = e_{i-1} + \alpha$$

Equation 7-5

Effect of Z-coordinate Shift on Algorithm #2

Three different cases will be investigated to understand how shifting a part or die representation along the z-axis effects algorithm #2. These include shifting a part not involved with system identification, shifting a part only involved with system identification, and shifting a die involved with system identification. These three scenarios are represented by the modified forms of Equation 6-1 that are shown in Equation 7-6 through Equation 7-8.

$$\Delta d_i = (d_2 - d_1) \otimes (p_2 - p_1)^{-1} \otimes (p_{ref} - p_{i-1} + \alpha)$$

Equation 7-6

$$\Delta d_i = (d_2 - d_1) \otimes (p_2 - p_1 + \alpha)^{-1} \otimes (p_{ref} - p_{i-1})$$

Equation 7-7

$$\Delta d_i = (d_2 - d_1 + \alpha) \otimes (p_2 - p_1)^{-1} \otimes (p_{ref} - p_{i-1})$$

Equation 7-8

In each of these cases, the Fourier transform is applied to the algorithm to simplify the convolution process. The resulting equations are shown in Equation 7-9 through Equation 7-11. In every case, the constant (α) is represented in the frequency domain by (A). Since (A) is the Fourier representation of a constant waveform, only its first term has a non-zero value. This first term is also known as the ‘DC’ term of the frequency

domain representation. It represents the sum of all points in the corresponding spatial domain representation of the respective waveform.

$$\Delta D_i = \frac{(D_2 - D_1)}{(P_2 - P_1)}(P_{ref} - P_{i-1} + A)$$

Equation 7-9

$$\Delta D_i = \frac{(D_2 - D_1)}{(P_2 - P_1 + A)}(P_{ref} - P_{i-1})$$

Equation 7-10

$$\Delta D_i = \frac{(D_2 - D_1 + A)}{(P_2 - P_1)}(P_{ref} - P_{i-1})$$

Equation 7-11

Because the only non-zero term in (A) is its first term, its inclusion can only alter the first terms in the frequency domain representation of any part, die, or change in die. Changes to these frequency domain ‘DC’ values only result in z-axis shifts in the spatial domain. Such shifts do not affect the shape of dies or parts. The reference location of parts and dies thus does not matter to algorithm #2.

Effect of Z-coordinate Shift on Algorithm #3

The shape of the die predicted by algorithm #3 is sensitive to the average z-axis position of all dies and parts used in the algorithm calculations. The importance of part and die z-coordinates is evaluated by observing the effects of adding a constant to a die and a part that are used in system identification, and a then to the reference part. These situations are represented in Equation 7-13 through Equation 7-15 when the governing equation of algorithm #3 is placed into the form of Equation 7-12.

$$\Delta d_i = \frac{\Delta d_2}{\Delta p_2} e_{i-1}$$

Equation 7-12

$$\Delta d_i = \frac{\Delta d_2 + \alpha}{\Delta p_2} e_{i-1}$$

Equation 7-13

$$\Delta d_i = \frac{\Delta d_2}{\Delta p_2 + \alpha} e_{i-1}$$

Equation 7-14

$$\Delta d_i = \frac{\Delta d_2}{\Delta p_2} (e_{i-1} + \alpha)$$

Equation 7-15

In each of these situations, the change to die variable (Δd_i) is altered differently at every data point. The addition of a constant, or inconsistent referencing of any part or die thus changes the shape of the die predicted by the algorithm.

Increased Sensitivity Near Zero

The shape changes introduced by shifting part and die positions are amplified when there are small differences between the parts used for system identification. The denominator of the equation used to define the controller gains (g_C) is the change in part position (Δp). When change in part position (Δp) is small, variation can overwhelm the controller gain equation. Figure 7-5 provides an illustration of this situation. Here, a Gaussian distribution drawn along the ordinate represents normal variation in formed parts. The resulting variation in the predicted dies lies along the abscissa. Variation has been increases significantly in this situation.

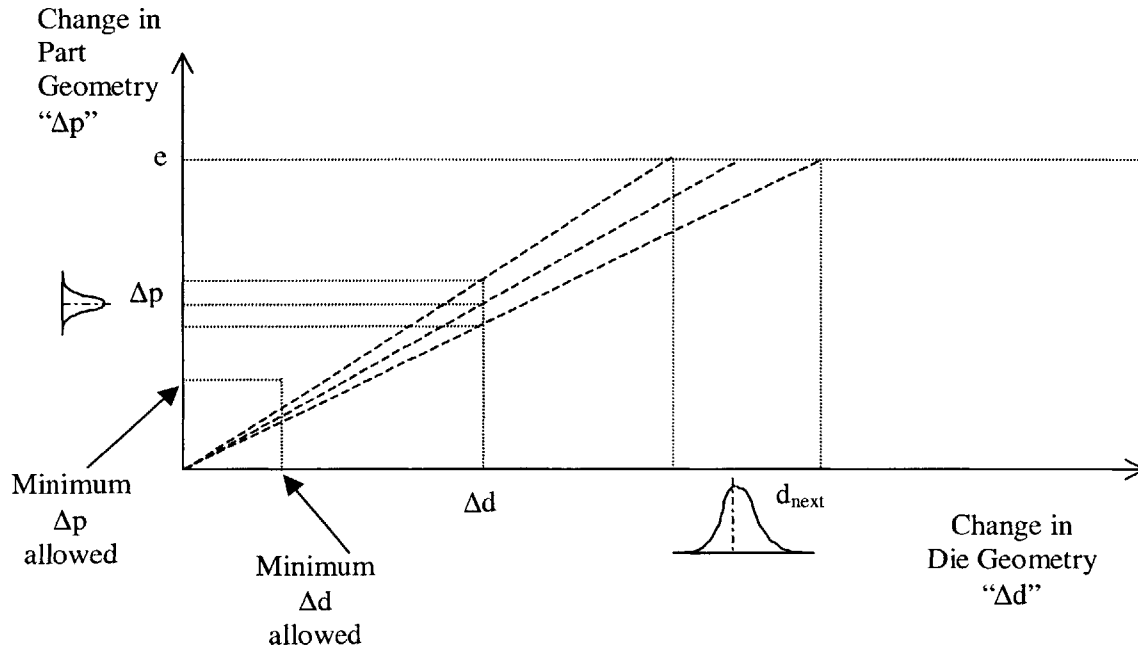


Figure 7-5: Sensitivity near 'zero'

Figure 7-6 provides a production scale example⁶ of the problem associated with a change in part value being near zero. Here a surface representing the change in the calibration parts is allowed to approach and cross the zero value position. The resulting system identifications, or gain values at the corresponding points are extremely high. These high gain values will most likely predict inaccurate die positions.

⁶ Data provided by Northrop Grumman

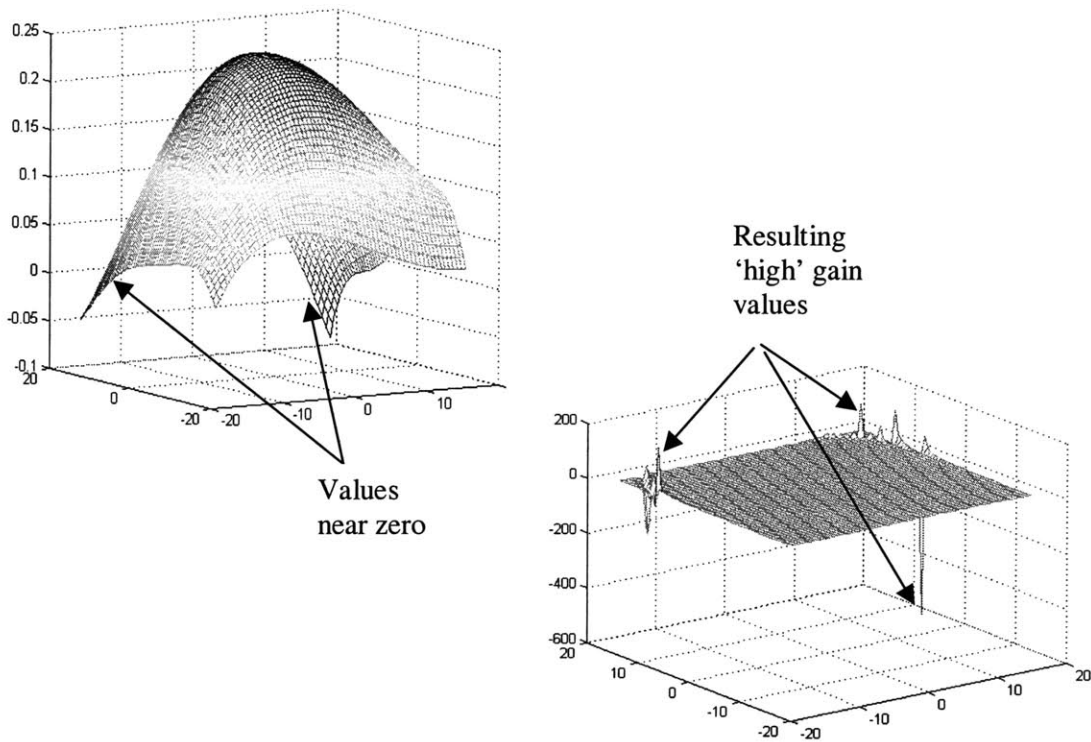


Figure 7-6: Production scale example of sensitivity near zero

Optimal Part Referencing for Algorithm #3

The datum chosen for any given part or die is arbitrary when one is only interested in the surface description. There is no correct position along the z-axis for a given part or die. However, algorithm #3 is sensitive to the z-axis position of parts and dies, which complicates matters.

To address this issue, a heuristic is devised that consistently references parts and dies along the z-axis. It does this in a manner that reduces the sensitivity the algorithm experiences near 'zero' values while maintaining optimal performance. This referencing heuristic is explained by the steps in Figure 7-7.

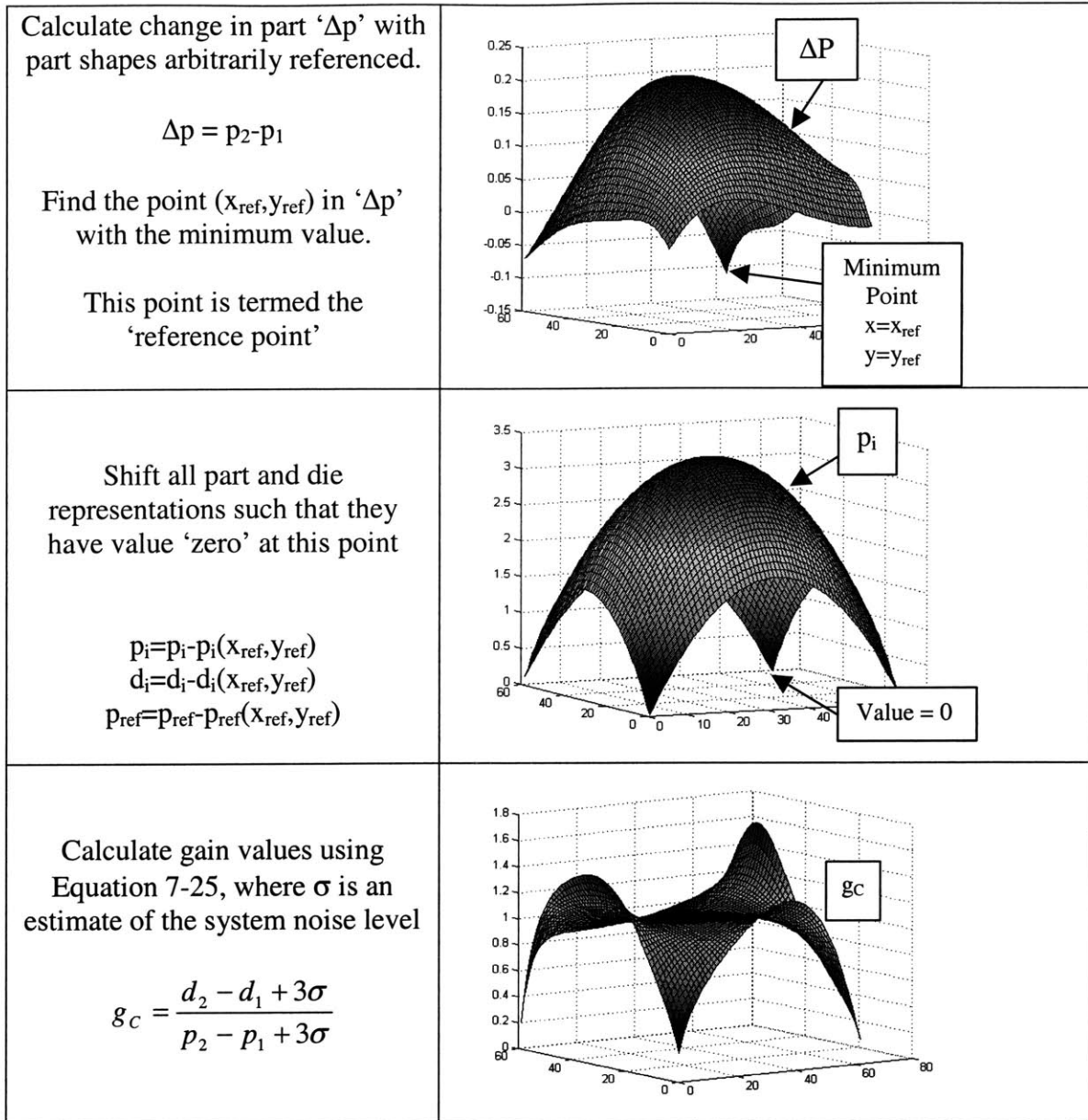


Figure 7-7: Heuristic for Algorithm #3

This heuristic safely translates all of the parts and dies away from zero and reduces the high sensitivity that occurs near the reference point. This concept is shown by Figure 7-8 where an identical minimum value is set for both the change in die (Δd) and change in part (Δp). In practice, this value is set to a 3σ , where σ represents the standard deviation of the process variation. This is reflected in Equation 7-16.

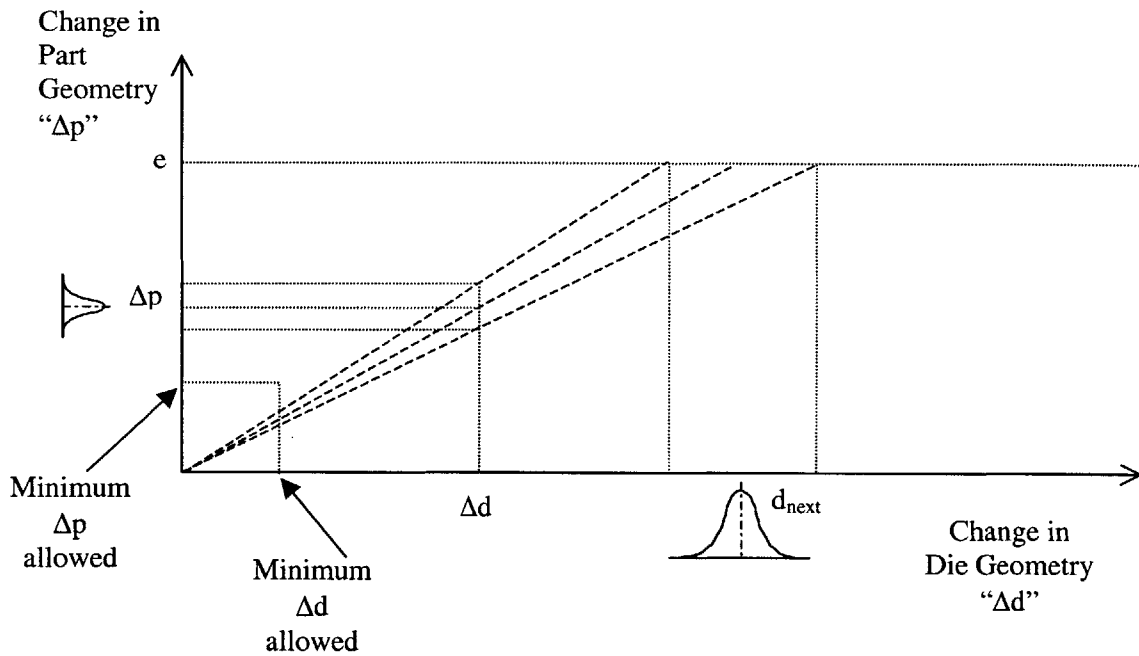


Figure 7-8: Reducing sensitivity near 'zero'

$$g_c = \frac{d_2 - d_1 + 3\sigma}{p_2 - p_1 + 3\sigma}$$

Equation 7-16

For high levels of process variation (σ), Equation 7-16 approaches a value of unity as reflected in Equation 7-17. This reduces the benefit of system identification by biasing all gain values toward unity. This eventually makes algorithm #3 identical to algorithm #1. Lower system variation levels therefore should improve algorithm performance.

$$\lim_{\sigma \rightarrow \infty} g_c = 1$$

Equation 7-17

Estimating Process Variation

An estimate of the process variation is found by first forming multiple parts ($p_1, p_2, p_3, \dots, p_K$) on a single die shape. The minimum RMS errors associated with these parts should be calculated with Equation 4-17 after they are registered in the normal manner. These errors ($e_1, e_2, e_3, \dots, e_K$) and the average of these minimum RMS errors (\bar{e}) are then used in conjunction with Equation 7-18 to define the process standard deviation.

$$\sigma = \sqrt{\frac{\sum_{k=1}^K (e_k - \bar{e})^2}{K}}$$

Equation 7-18

Algorithm Concerns

Variation Treated Differently Across the Die

For any common change in surface, some change in part and change in die pin positions are closer than others to the z-axis value of zero in the system identification calculations. Those values closer are more sensitive to noise in the z-direction. The heuristic developed in this research addresses this concern by biasing all of such values away from zero. This has caused two concerns. First, all gain values have been biased away from their ideal value. Second, those values closer to zero remain more sensitive to noise. This uneven treatment of noise has raised concerns as it may cause some areas of the part to converge faster than others.

Coupling Ignored

Algorithm #3 treats each pin as a SISO system. However, coupling does exist in the system as was shown in Figure 5-3. The model used by algorithm #3 ignores this coupling, but tunes the SISO system gains based on the behavior of the coupled system. Nonetheless, algorithm #3 ignores the fact that a movement in one die pin affects the outcome of multiple part positions.

System Identification Ambiguity

It has been pointed out by researchers at Northrop Grumman that system identification (g_c) values are dependent on which calibration part and die are defined as the first part and die and which are defined as the second part and die. Changing the order of these parameters will change which portion of the part and die that coincide with controller gain values which are more sensitive to noise. To address this concern, a strategy that places these more sensitive portions of the part nearer to the physical tools reference locations is suggested. System noise is likely to have less effect on portions of the part nearer to the physical reference points.

Summary

In summary, a new algorithm was presented in this chapter. This algorithm addresses the concerns of algorithm #1 in a much simpler manner than algorithm #2 without compromising performance. The new algorithm does have a sensitivity to how parts are references, which is addressed through a simple heuristic. The next chapter provides an overview of the tools used to evaluate all of the algorithms.

Chapter 8 : Simulation Methods

Throughout this research, various simulation techniques are employed to assist in understanding the behavior of the different forms of the algorithm. The simulations are presented and discussed within this chapter.

Full Part Forming Simulation

Dr. Simona Socrate and Dr. Mary Boyce [2000] have performed extensive investigations into FEA techniques that accurately represent the mechanics of stretch forming over a discrete die. Dr. Socrate and Dr. Boyce created the full forming simulation presented herein to mimic the specifics of MIT's lab scale stretch press shown in Figure 8-1. Only die and sheet geometry parameters have been modified by the author to match specifics of Valjavec's [1999] experiments.

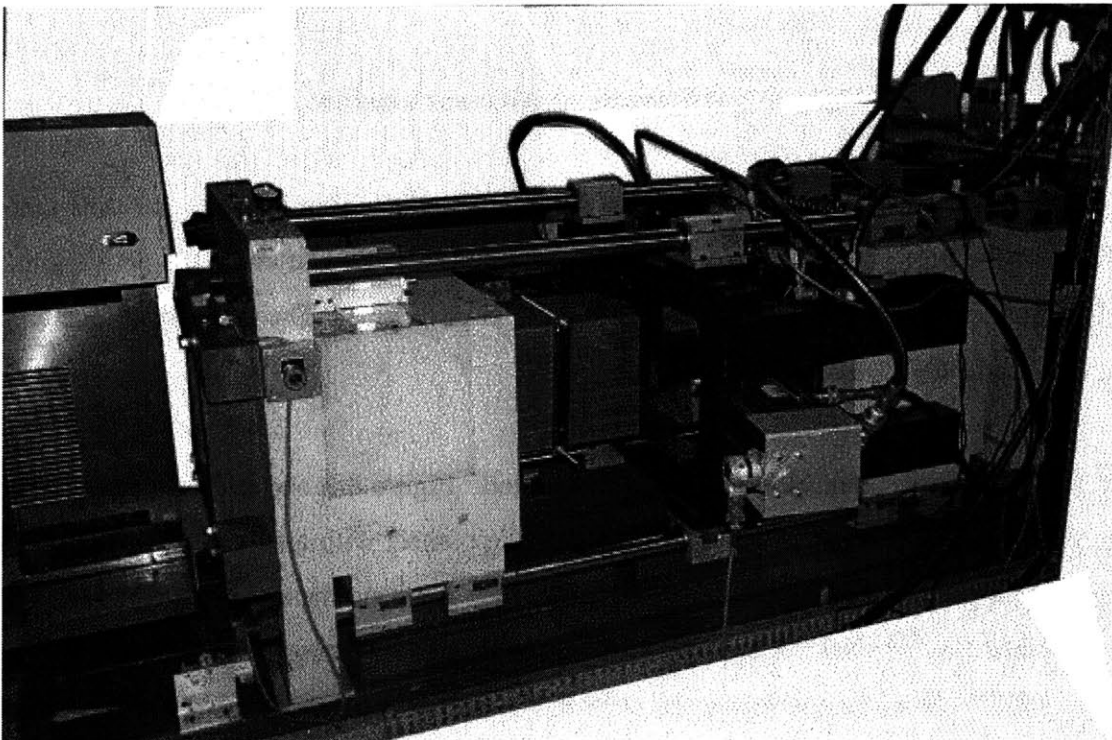


Figure 8-1: MIT, lab scale stretch press

The full part simulation is used so that algorithm performance can be evaluated in a deterministic environment. The part forming simulation consists of several programs, including an ABAQUS finite element model and several companion Matlab programs. The flow of the simulation process is documented in Figure 8-2.

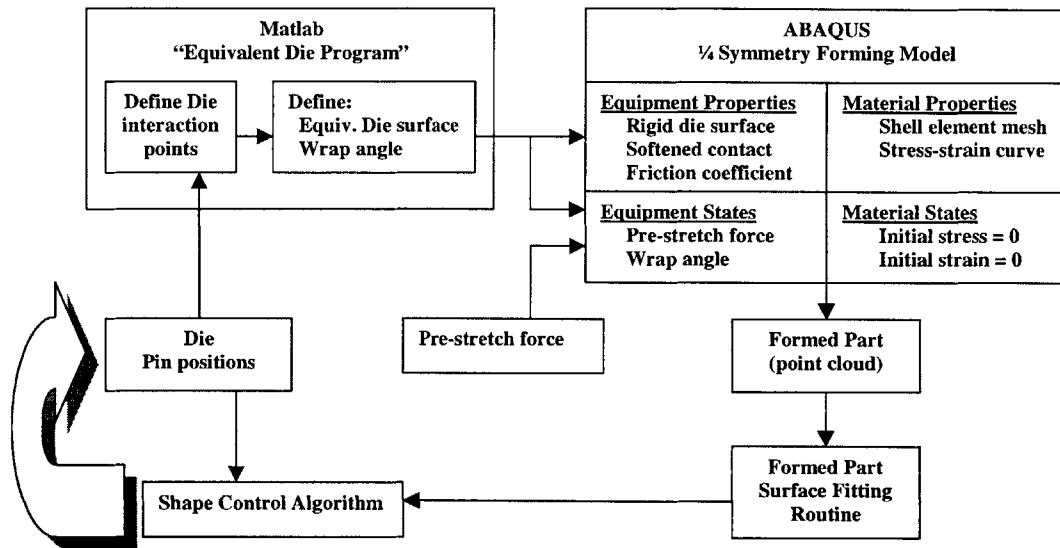


Figure 8-2: Full forming simulation process flow

The process begins by defining the pin positions of the die. This includes the pins that are beneath the portion of the part that is eventually used, also known as the active pins and the pins under support sections of the die, or passive pins. These pin positions are input into the Matlab[®] ‘equivalent die’ program.

Next, the interaction point of every pin is calculated. Figure 8-3 shows how the tip of a given pin may not actually be the first point to make contact with a sheet that is laid over the die. This portion of the program identifies which point on the spherical tip of every pin will make first contact. A smooth surface is then created from the interaction points of every pin. This surface is formatted such that it will be recognized as a surface mesh by ABAQUS. An example of a mesh is shown in Figure 8-4. The Matlab program additionally defines the average pull off angle at the edge of the equivalent die shape.

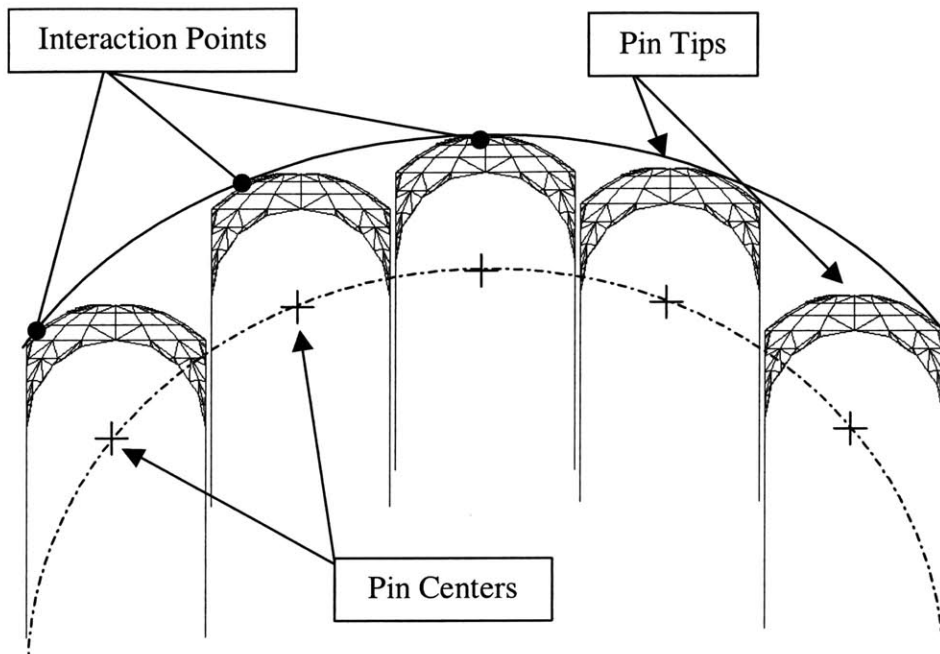


Figure 8-3: Pin position and die interaction positions

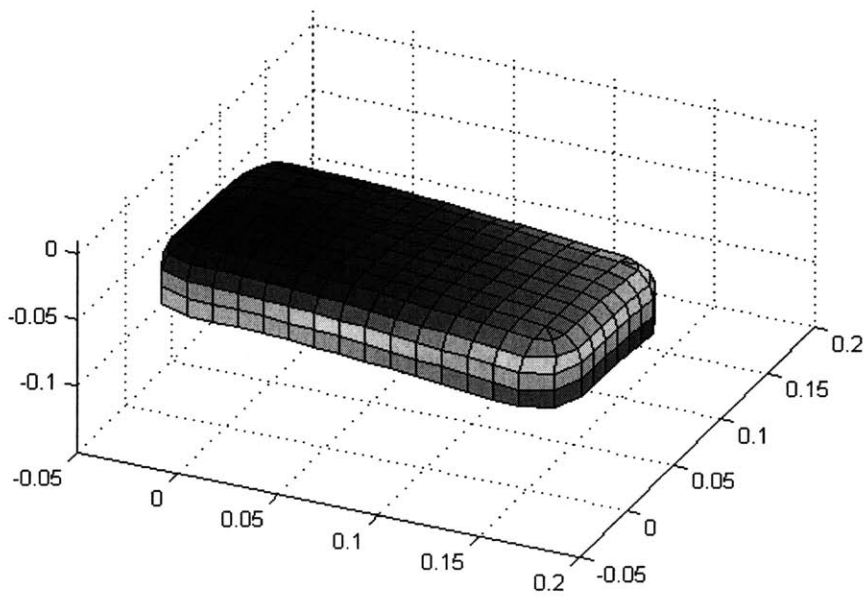


Figure 8-4: Equivalent, rigid die surface

The rigid die surface and the pull off angle are then passed into an ABAQUS model. These are the only equipment states and parameters that are changed in the course of simulating how an algorithm converges for a particular part. The ABAQUS model however has numerous other parameters that must be defined.

Material properties that must be defined include the stress-strain behavior of the material, and the material thickness. These properties are applied to the shell element mesh that is representing the sheet metal blank in the forming process. This blank is shown along with a rigid die in Figure 8-5.

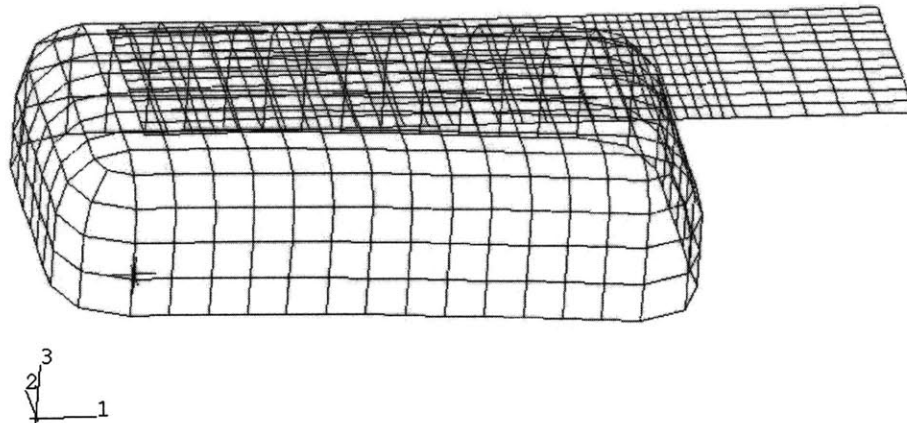


Figure 8-5: Representation of material blank and die in ABAQUS

As previously mentioned, the ABAQUS program only simulates $\frac{1}{4}$ of the actual part. The simulation is thus only valid for parts that are symmetric in nature. The assumption of $\frac{1}{4}$ symmetry allows the ABAQUS program to compute solutions in much shorter time. To represent $\frac{1}{4}$ symmetric parts, boundary conditions are applied to the mesh that represents the sheet metal blank. These boundary conditions are shown Figure 8-6. It should be noted that the material is assumed to have no initial stresses or strain history.

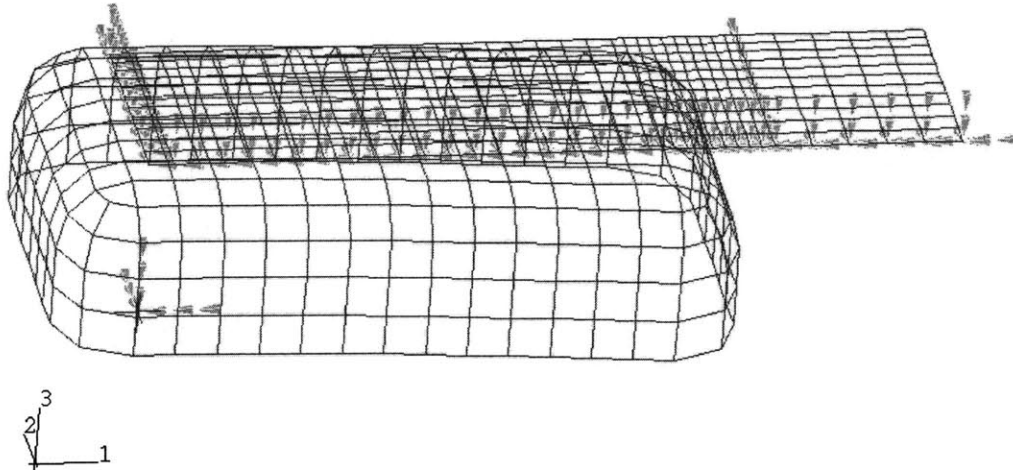


Figure 8-6: Boundary conditions

The primary equipment property is the geometry of the equivalent rigid die, which is provided by the Matlab program. The properties of the interpolator do however need to be accommodated in the model. This is performed through the use of a softened contact interaction between the sheet metal mesh and the rigid die mesh. Figure 8-7 depicts the concept of softened contact. With the softened contact model, the pressure at the sheet to die interface is a function of the clearance between the two surfaces. The softened contact begins when the clearance between the die and the sheet is $\frac{1}{2}$ of the interpolator thickness. For clearances less than this, the pressure exerted across the sheet to die interface is defined by a function like the one shown in Figure 8-7.

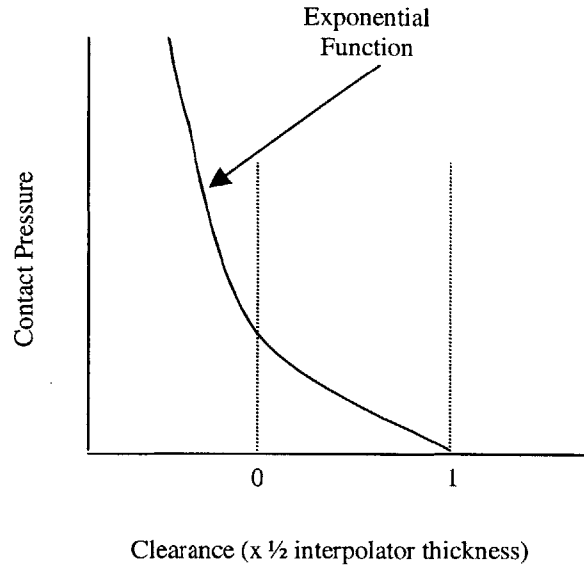


Figure 8-7: Explanation of softened contact

The process states are also pre-defined. These primarily include the pre-stretch force and the wrap angle (which was defined by the Matlab program). This simulation does not currently include a post-stretch segment. The pre-stretch portion of the process consists of applying the pre-stretch force to the material until an equilibrium point is reached. To begin the wrapping phase, the sheet is translated down until it makes initial contact with the die. Once an initial contact force is witnessed, the sheet is incrementally wrapped about the die while the stretch force is maintained constant. The sheet is wrapped around the part until the final wrap angle is obtained.

The part is now released from the stretch forming process by relaxing the constraints on the stretch forming jaw end of the sheet. The sheet is first allowed to relax axially, and then it is allowed to spring back to eliminate any residual moments throughout its cross section.

The ABAQUS portion of the simulation is now complete. A cloud of points that represent the nodal positions part in the sprung back position are now exported to an additional Matlab program that fits the data with a smooth surface. Points are then

interpolated from this surface at points coincident with the pin positions used to define the die surface.

Data now exists for a part and the die that it was formed on. This data is input into one of the algorithms along with the reference part information. This information is used to compute the next die shape, which is then passed along to the Matlab equivalent die program where the entire process is repeated. It should be noted that algorithm #2 cannot be replicated accurately on information from only $\frac{1}{4}$ of a die or part. For this reason, part and die data is quadrupled to create an full die and part representation before it is input into algorithm #2.

Accuracy of Full Part forming simulation

The equivalent die program and the $\frac{1}{4}$ symmetry ABAQUS simulation were created by Dr. Simona Socrate and Dr. Mary Boyce [2000] and subsequently provided to the author. The programs used in this research were only modified to make the sheet metal blank size consistent with the blank size used in Valjavec's [1999] experiments. Further details of these programs are available in the RTFF Annual and quarterly reports from 1996. These reports provide evidence that the modeling methodology used in this simulation provides comparable results to simulations that model the entire discrete die surface explicitly instead of with the equivalent die and softened contact. These reports also provide comparisons between cylindrical shapes formed on MIT's lab scale stretch forming press and parts formed in a simulation that has been calibrated to the specifics of that forming process. The results of this comparison suggest that the simulation provides a very accurate representation of the MIT stretch press.

However, the simulation is not used in this research to replicate experimental results. It is used to provide a deformation model similar to that found in the stretch forming process. The algorithms are then rated on how well they can accommodate this deformation process and predict a die that forms a desired part shape with the provided deformation model. For this reason, while the accuracy of the deformation model is good, it is not crucial to this research.

Simulating the Effects of Variation on FEA Formed Components

There is desire to understand how variation affects the algorithms. To understand this, numerous simulations or experiments must be run to evaluate how variation is amplified or attenuated through each algorithm and how variation affects the convergence of each algorithm. However, the time required to provide a statistically significant sample of simulations is excessive. To achieve results in a much shorter time span, the algorithm performance can be measured on how well it predicts the die that is used to make a part of a known shape.

The equations of each algorithm are typically used to predict a die shape that will form a desired part. The performance of the algorithms is measured on how well these parts conform to the desired parts. However, measuring algorithm performance by how well they predict a die that forms a known part can allow simulations to be performed in much less time. This is because no additional parts need to be formed through simulation, which normally takes a considerable amount of time.

First, an initial die shape is chosen and then a part is formed over it in simulation. Next, two additional dies are defined by multiplying the initial, spatial die z-coordinate by 1.1 and 0.9 respectively as indicated in Equation 8-1. Parts are then formed on these dies as well. There is now knowledge of three similar die shapes and the parts that have been formed on them as shown by Figure 8-8.

$$d_2 = 1.1d_1$$

$$d_3 = 0.9d_1$$

Equation 8-1

Where:

d_1 = initial die shape (array of vertical z-coordinates)

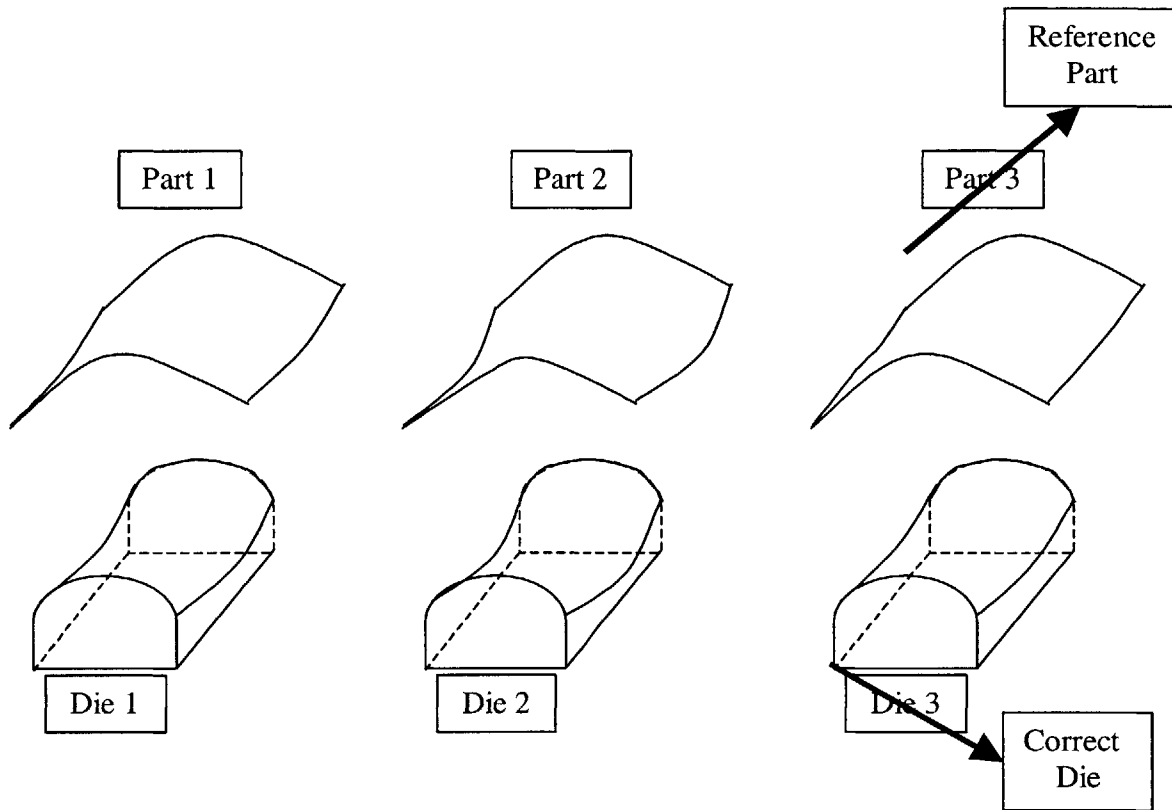


Figure 8-8: Dies and Parts for simulation shortcut

To perform the shortcut, the third part shape is arbitrarily considered the reference part shape. Treating the third part in this manner provides exact knowledge of the die that will form the reference part. The information in the first two die and part shapes is now used by the various algorithms to estimate a die shape that will create the reference part (or alternatively, the third part). Algorithm performance can be measured by how close the die predicted by each algorithm is to the correct die shape. The die error associated with each algorithm is calculated with Equation 8-2.

$$die_error = d_* - \hat{d}_3$$

Equation 8-2

Where:

d_* = die shape used to form the reference part

\hat{d}_3 = estimate of correct die shape
(made from d_1 , d_2 , p_1 , p_2 , & p_{ref})

This 'shortcut' provides a means to evaluate variation effects without making numerous parts, through simulations or experimentation. Representations of any part can be altered by additive noise as shown in Figure 8-9 to introduce variation into an otherwise deterministic system.

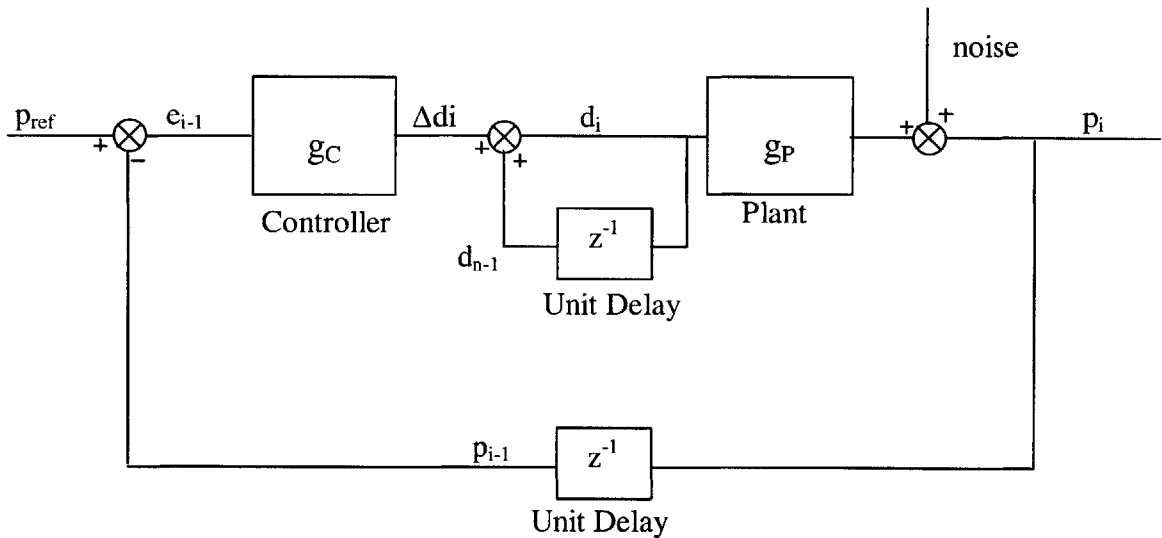


Figure 8-9: Introducing additive noise

The scenario shown in Figure 8-9 is most indicative of how noise exists in an experimental or production environment. Noise in this case is not added to part 3, which now represents a pre-defined reference shape. The noise is, however, included in the calculations performed to arrive at controller gain values (g_C).

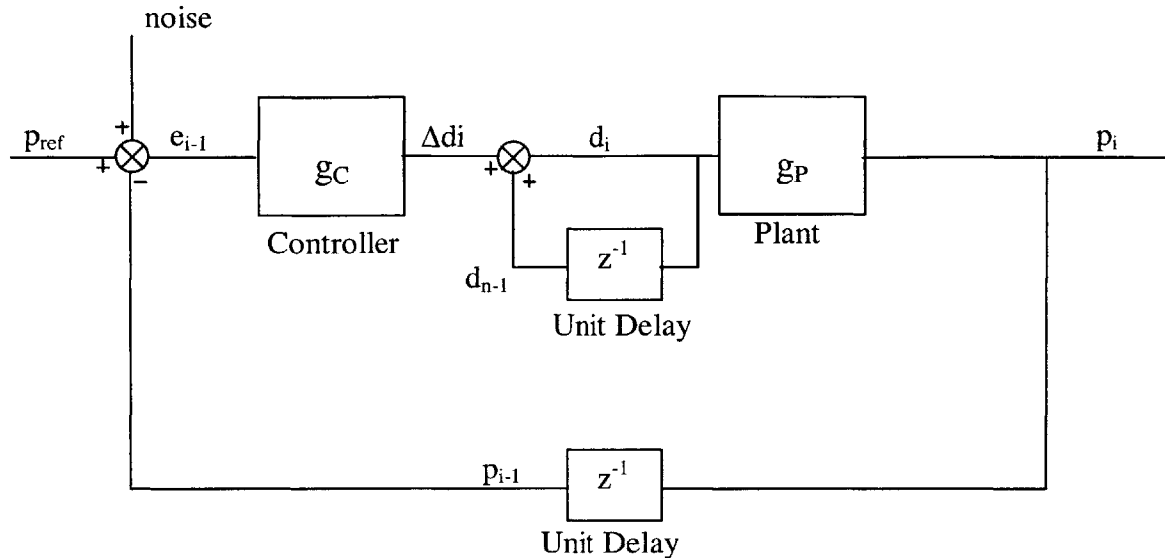


Figure 8-10: Introduction of additive noise to reference part

Alternatively, noise can be added to the reference part as shown in Figure 8-10. Adding noise to the reference part is not indicative of an actual forming scenario. It however introduces noise into the representation of each part without corrupting the part representations used to define the controller gains (g_C). Such simulations help define how sensitive the system identification is to variation.

This ‘short cut’ approach does have some drawbacks. It does not model any effects the actual forming process may have on the variation level. For instance, the interpolating layer has been shown by Valjavec [1999] to act as a low pass filter. The shortcut does not model this effect. This suggests that the simulation will overstate the variation amplification.

Summary

In summary, a process has been developed that allows stretch forming on the MIT press to be simulated through ABAQUS and MATLAB code. Additionally, a shortcut has been developed which allows the effects of noise to be evaluated in a controlled manner without forming numerous components.

Chapter 9 : Simulation Results

In this chapter, the simulations discussed in Chapter 8 are used to evaluate the performance of all of the algorithms. These simulations are capable of quantifying performance with respect to algorithm rate of convergence, and robustness to variation.

Comparison with Experimental Results

Before the algorithms are evaluated in simulation, attempts are made to replicate Valjavec's [1999] lab scale experiments. This is done to understand how closely the full forming simulation represents the lab scale stretch forming process.

There exist several differences between the experimental situation and the simulation. The primary differences include the absence of noise and the greater level of control over process trajectories in the simulation environment. The simulation environment also eliminates the need for a registration routine to orient the parts to their respective dies.

There are additional differences in the algorithms being used in the simulation and in Valjavec's [1999] experiments. The algorithm used in the simulation operates on parts and dies that are defined by a grid of points spaced consistently with the pins in the MIT reconfigurable tool. The experiments performed by Valjavec [1999] use a grid of points that is much more dense. The simulation algorithm uses a two-dimensional hanning window to smooth the part representations used in system identification. The experiments used the Welch method instead to smooth the part representations.

All process parameters are set to values consistent with the experimental parameters. These values are shown in Figure 6-21. Reference parts are defined in an identical manner. These reference parts are shown in Figure 6-22. Initial dies are defined by offsetting the reference die shape by an amount equal to the interpolator thickness plus the pin tip radius. The equations used for the reference parts and first dies are shown in Figure 9-1. The second die shape in the simulation and the experiment is defined by applying algorithm #1. All equations in Figure 9-1 are based on a coordinate system defined in inches.

Shape	Reference Part	Initial Die
Cylinder	$z = (10.65^2 - x^2)^{1/2}$	$z = (10^2 - x^2)^{1/2}$
Saddle	$z = \left(45^2 - x^2 + \left(\frac{32.048}{45} y \right)^2 \right)^{1/2}$	$z = \left(45^2 - x^2 + \left(\frac{33}{45} y \right)^2 \right)^{1/2}$
Spheroid	$z = \left(20^2 - x^2 - \left(\frac{20}{80} y \right)^2 \right)^{1/2}$	$z = \left(19.048^2 - x^2 - \left(\frac{19.048}{79.048} y \right)^2 \right)^{1/2}$

Figure 9-1: Equations used to define reference parts and initial dies

Comparisons between the experimental results and the simulated results are shown in Figure 9-2 through Figure 9-4. Maximum part errors are presented in these figures for each of four simulated forming iterations. All available experimental data is also shown.

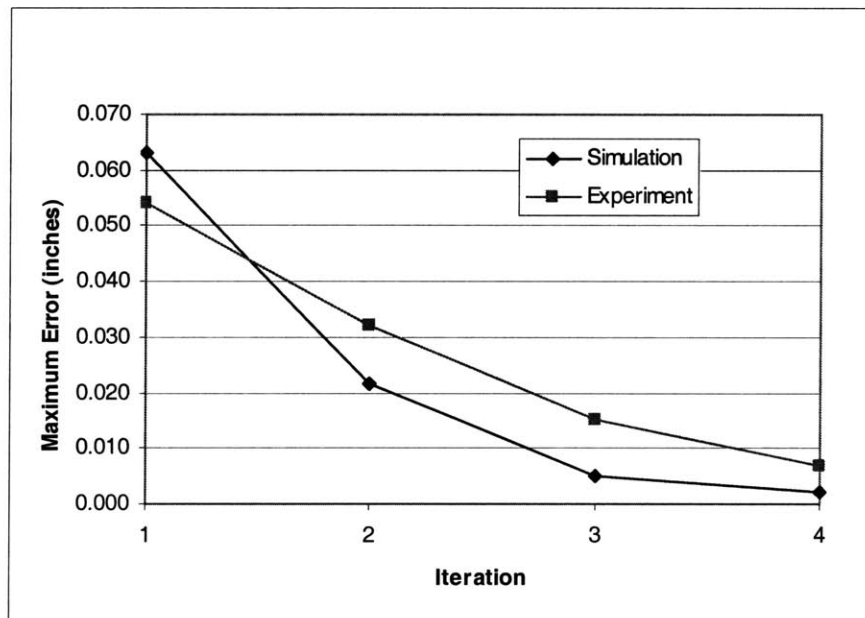


Figure 9-2: Replication of Valjavec's [1999] cylinder experiment

The results from the cylinder experiment are close to the simulated results. Improved results could most likely be obtained if time were spent calibrating the model to the

specifics of the forming situation used to create Valjavec's parts, or if Valjavec's exact algorithm were used.

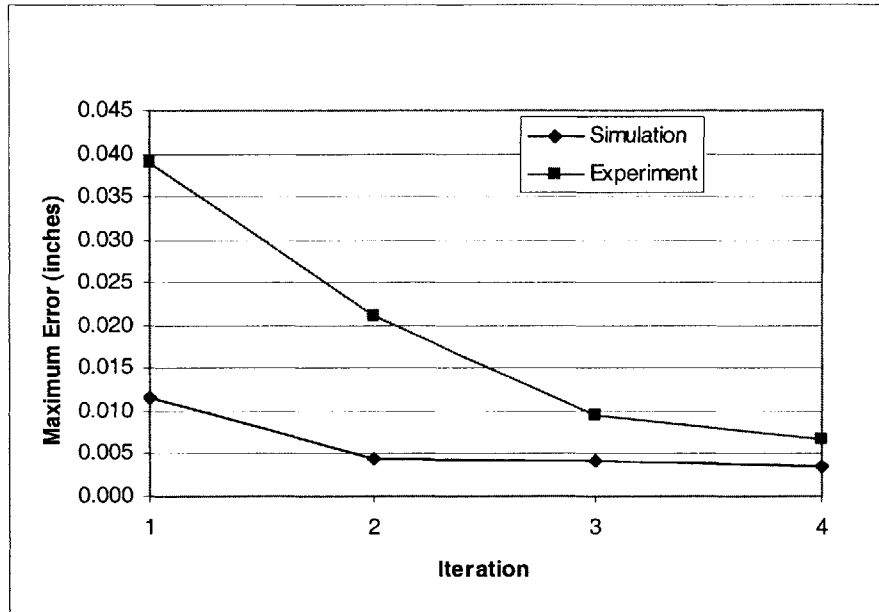


Figure 9-3: Replication of Valjavec's [1999] saddle experiment

The saddle experiment and simulation display significant differences in the first formed part. These differences diminish as the algorithms progress toward a zero error part.

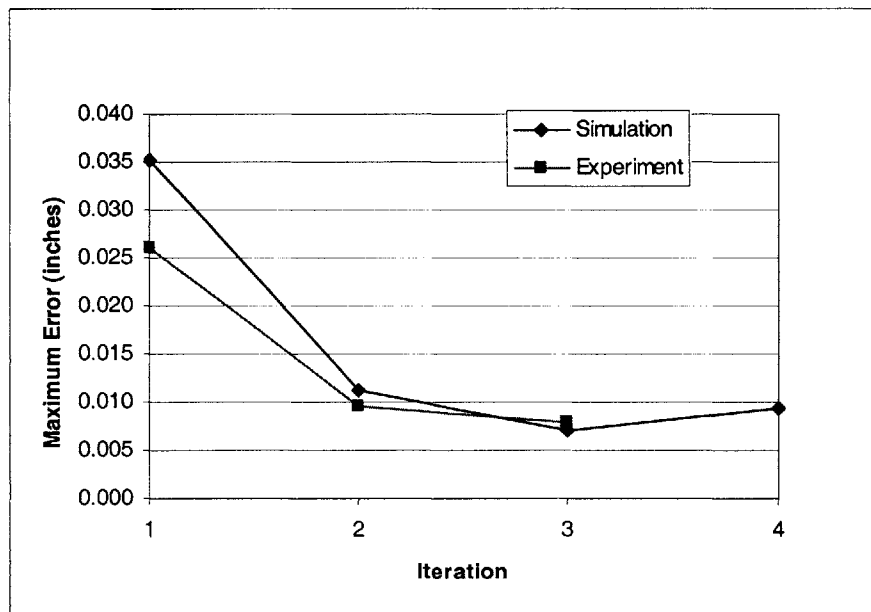


Figure 9-4: Replication of Valjavec's [1999] spheroid experiment

Results from spheroid experiments compare closely to one another. It should be noted that the experiment was terminated after three iterations as an acceptable part was achieved. Maximum error values below 0.010 inches were defined as an acceptable error levels for the lab scale process.

In summary, the cylinder and the spheroid simulations replicate the values and trends of Valjavec's [1999] forming experiments within the expected range of uncertainty. While the trend of the saddle experiment and simulation are similar, the absolute values of the errors are not. This disparity may be explained by any of the previously mentioned differences. The difference in absolute value is however not of great concern for this research as the ABAQUS model is used to provide a deformation model with properties similar to an actual stretch forming process, not to explicitly model the MIT lab scale stretch forming process.

Algorithm Comparisons with Full Forming Simulations

The various algorithms are now tested on three of the reference shapes used in Valjavec's [1999] experiments. Again, the test shapes are shown in Figure 6-22.

These experiments begin with the pin positions adjusted to create a net shape die at the interpolator to sheet metal interface. The equations used to define these initial dies and the reference parts are shown in Figure 9-1. The second part shapes are defined by applying algorithm #1 to the first die and part shape. The third and fourth die are each defined by four different methods, including algorithm #1, algorithm #2 with a hanning window used on parts in system identification, algorithm #2 without windowing, and algorithm #3.

The minimum RMS errors for each simulation are shown in Figure 9-6 through Figure 9-8. Figure 9-5 provides a summary of the forming trials in tabular form. Valjavec's [1999] experimental results are included for reference. The algorithm with the best performance in every category is highlighted in the table.

	% of Initial Min RMS Error					
	Cylinder		Saddle		Spheroid	
	Iteration #3	Iteration #4	Iteration #3	Iteration #4	Iteration #3	Iteration #4
Alg #1	12%	6%	25%	19%	12%	6%
Alg #2 (no window)	4%	27%	116%		17%	35%
Alg #2 (window)	7%	3%	27%	26%	16%	22%
Alg #3	4%	2%	22%	18%	6%	4%
Experiment	28%	13%	24%	17%	31%	

Figure 9-5: Summary of full simulation results

The unequivocal best performer in this simulation is algorithm #3. Its error convergence outperforms all other algorithms in every shape classification. It should be noted that a fourth iteration of the saddle was not attempted for algorithm #2 (no window) as the predicted die was determined to be invalid.

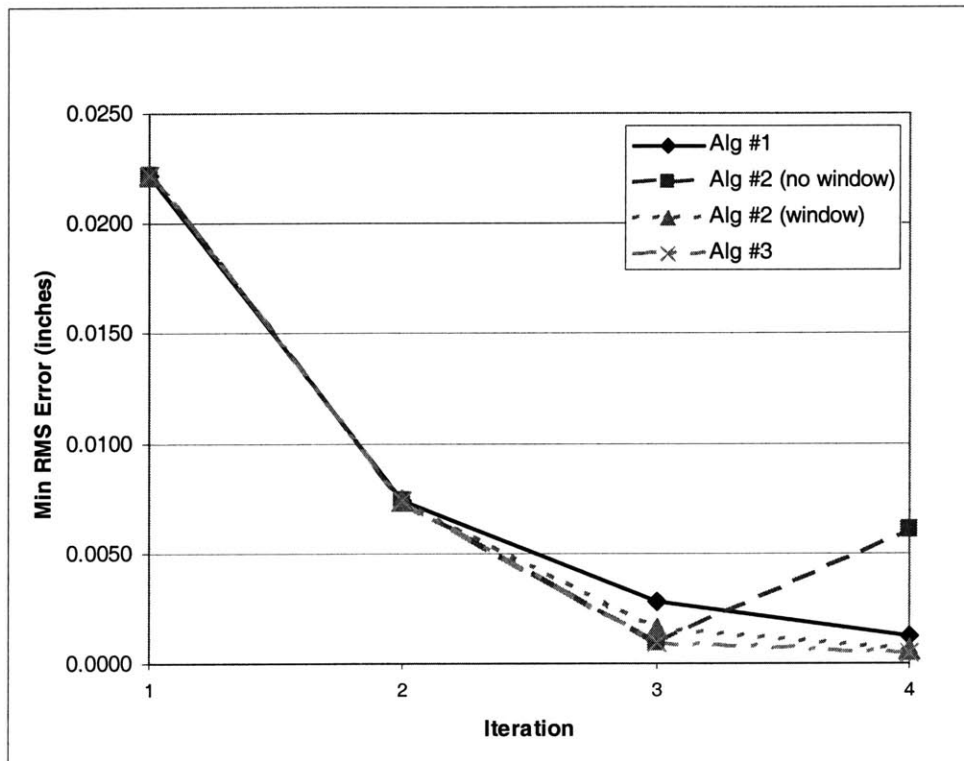


Figure 9-6: Simulations of cylinder forming trial

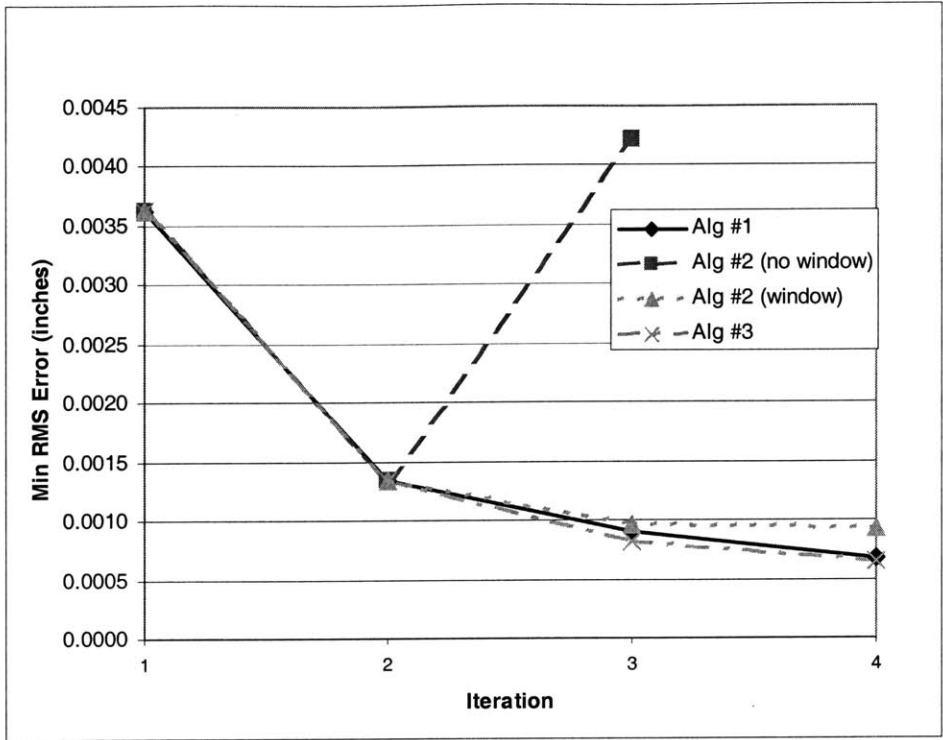


Figure 9-7: Simulations of saddle forming trials

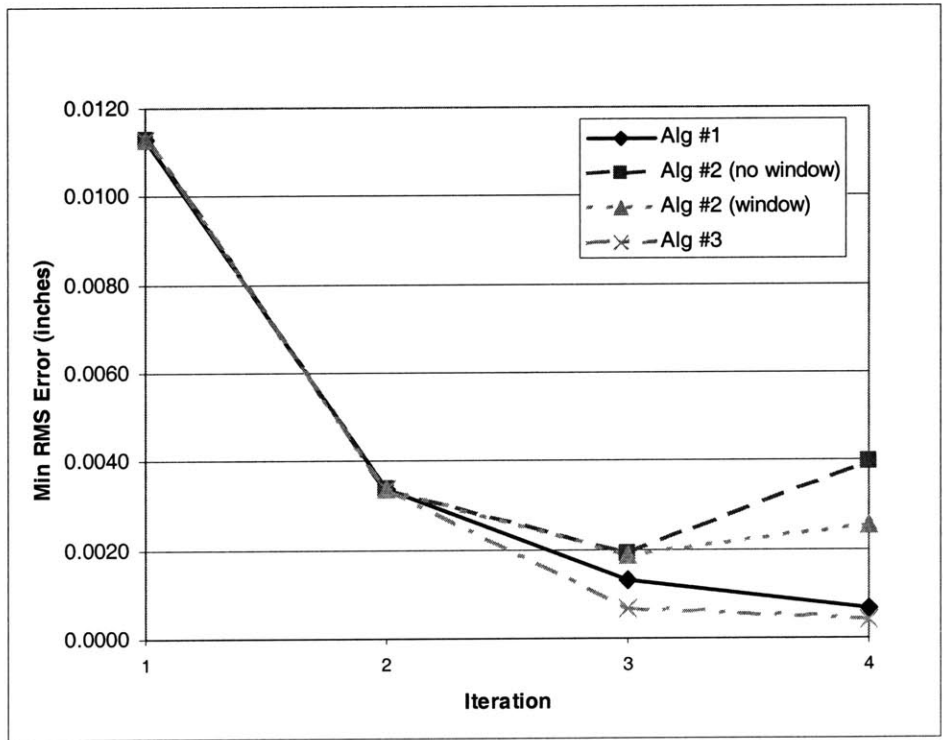


Figure 9-8: Simulations of spherical forming trials

Variation Investigations, Noise included in Identification

The techniques mentioned in Chapter 8 are now used to evaluate algorithm robustness to variation. Noise is input to the system by adding a normally distributed value to each z-coordinate data point for both part 1 and part 2 as suggested by Figure 8-9. A predicted die is calculated for each algorithm and then the die error is measured. This is repeated fifty times to complete each simulation.

To calculate how variation is amplified through the algorithm, the minimum RMS error associated with part 2 is first calculated for each of the fifty trials to establish a baseline. This baseline is calculated with Equation 7-18. This value serves as the denominator for Equation 9-1.

Equation 7-18 is then used to calculate the noise level of the predicted die shape errors. This value serves as the numerator of Equation 9-1.

$$\text{Amplification} = \frac{\text{stdev}(e_d)}{\text{stdev}(e_2)}$$

Equation 9-1

Where:

e_d = error between each predicted die and the correct die
 e_2 = error between the reference part and part 2

The variation amplification experienced by the three different test shapes in fifty different simulated forming trials is presented in Figure 9-9. The amplification is mapped for three different process variation levels.

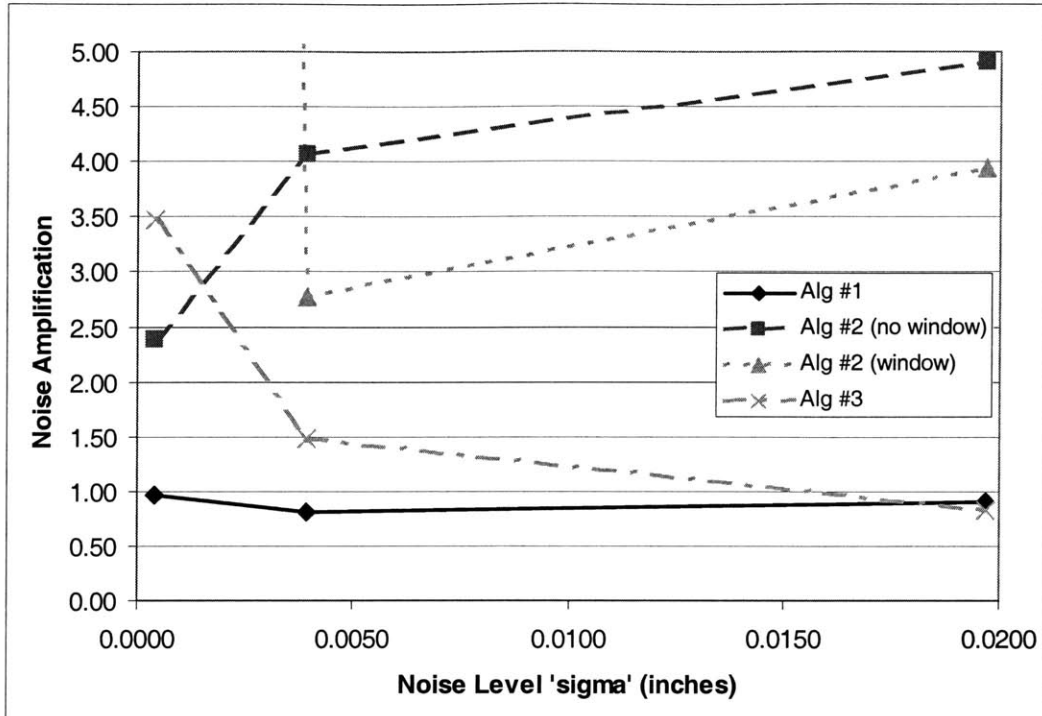


Figure 9-9: Variation amplification, noise included in identification

The lowest level of variation amplification is shown by algorithm #1. This is expected, as it makes no attempt at system identification and thus has less chance of amplifying the system noise. Algorithm #3 has the next consistently lower amplification level, which exceeds algorithm #2 (no window) only at very low noise levels. Algorithm #2 (window) experiences excessive variation amplification at low noise levels. Its performance does improve at the higher levels.

The average error value over the fifty simulations is also of interest. It provides insight into how noise affects the convergence of each algorithm. This value is calculated with Equation 9-2. It is the average of the minimum RMS error values produced by the 150 trials of any given algorithm (50 trials per shape). The resulting values are shown in Figure 9-10.

$$\text{average_error} = \text{average}(e_d)$$

Equation 9-2

Where:

e_d = error between each predicted die and the correct die

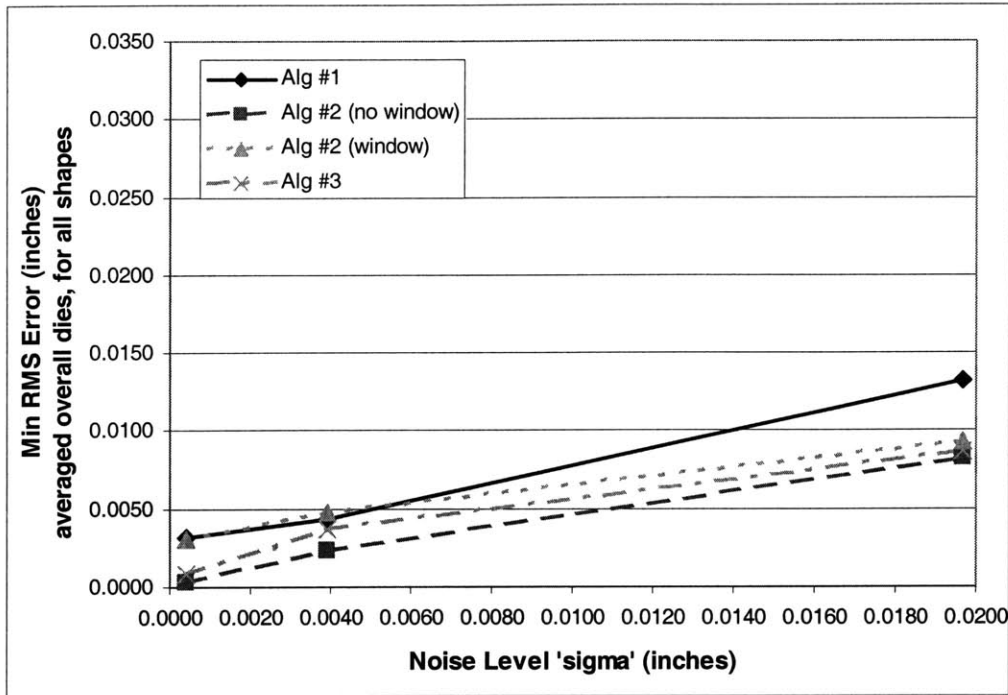


Figure 9-10: Error convergence during variation investigation simulation, noise included in identification

Algorithm #2 (no window) consistently outperforms the other algorithms with respect to error convergence in this test. Algorithm #3 performs the next best. Algorithm #2 (window) is the consistent poor performer, and is even outperformed by algorithm #1 at low variation levels.

The results for both forms of algorithm #2 contradict results from the full forming simulation. There, algorithm #3 consistently converged in the fewest iterations. Algorithm #2 (window) also performed better than algorithm #2 (no window) in some situations. These discrepancies likely result from measuring algorithm performance on how well a known die shape is predicted. The full forming simulation measures performance on how well the reference part is formed. Many of the dies predicted by

algorithm #3 have random variation that will be filtered out by the interpolator during the forming process. The errors witnessed by the algorithm #2 dies are often periodic in nature and are less likely to be eliminated by the interpolator during forming. Figure 9-11 provides some examples of these die errors.

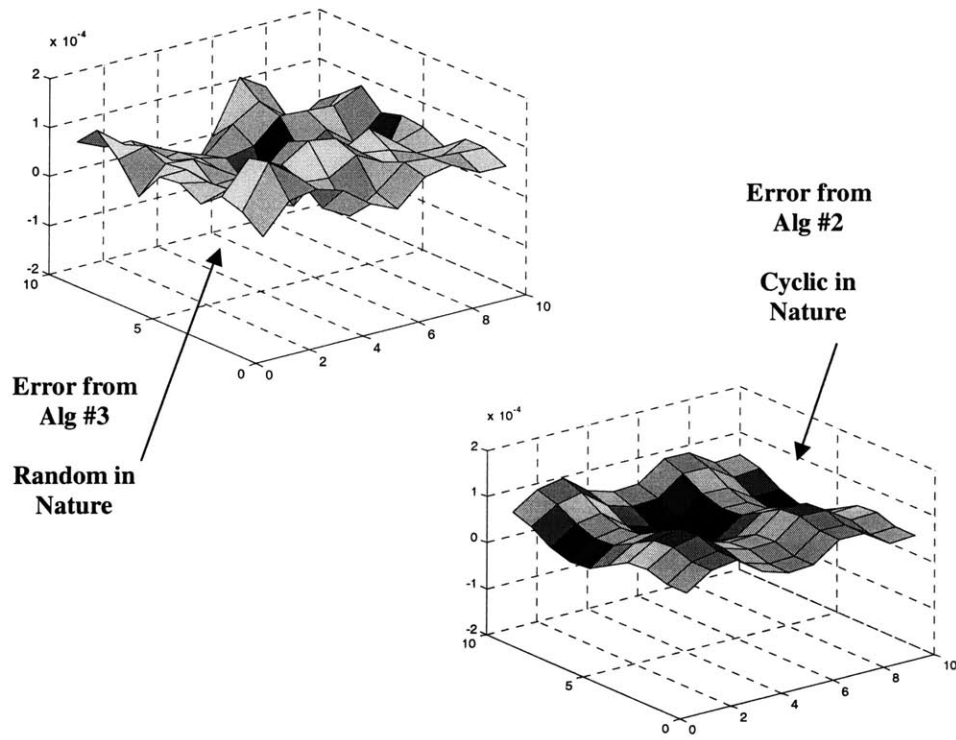


Figure 9-11: Die errors for algorithm #2 (no window) and algorithm #3

In summary, algorithm #1 is affected least by noise. This is expected, as it does not employ system identification. Algorithm #3 proves to amplify noise less than either form of algorithm #2 except at very low noise levels. The error convergence rates seen in this simulation contradict the outcome of the full forming simulations. The full forming simulation proved algorithm #3 consistently converged to a lower error value in the same amount of iterations. This simulation suggests that algorithm #2 without windowing surpasses algorithm #3 in this regard. This discrepancy may result from comparing algorithm performance on how well die shapes are predicted instead of part shapes. Comparing die shapes excludes the low-pass filtering effects of the interpolator from the simulation.

Variation Investigations, Noise Excluded from Identification

The simulation is then repeated with noise excluded from the system identification process. This is done by computing a controller gains (g_c) on noise free parts and dies and then adding noise to the part 2 error term. The schematic in Figure 8-10 shows how this is performed. The noise amplification and error convergence rate are computed in the same way as the previous experiment

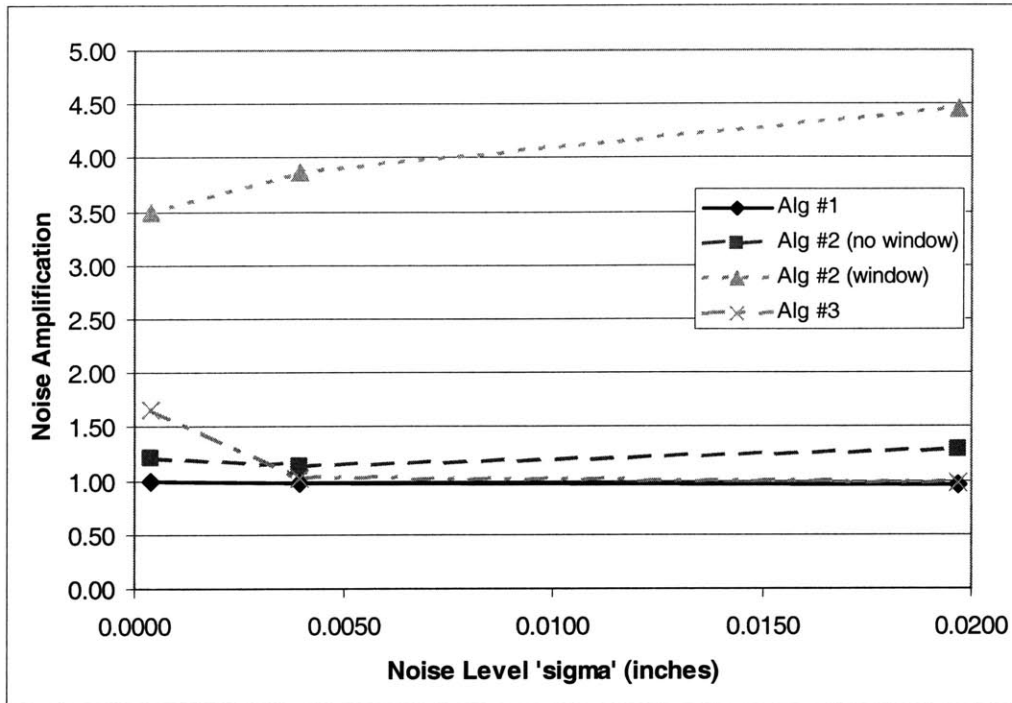


Figure 9-12: Variation amplification, noise excluded from identification

Figure 9-12 provides the variation amplification results and Figure 9-13 details the error convergence rates for this simulation.

Algorithm #1 obtains the lowest variation amplification level as expected. Again, in this scenario algorithm #3 achieves the next lowest variation amplification at all levels except the smallest process noise level. The trends for both forms of algorithm #2 are similar to results from the prior simulation, except that algorithm #2 (no window) always performs better than algorithm #2 (window). The noise amplification values are generally lower

than those associated with noise included in the system identification. This suggests that a noisy system identification may worsen algorithm performance.

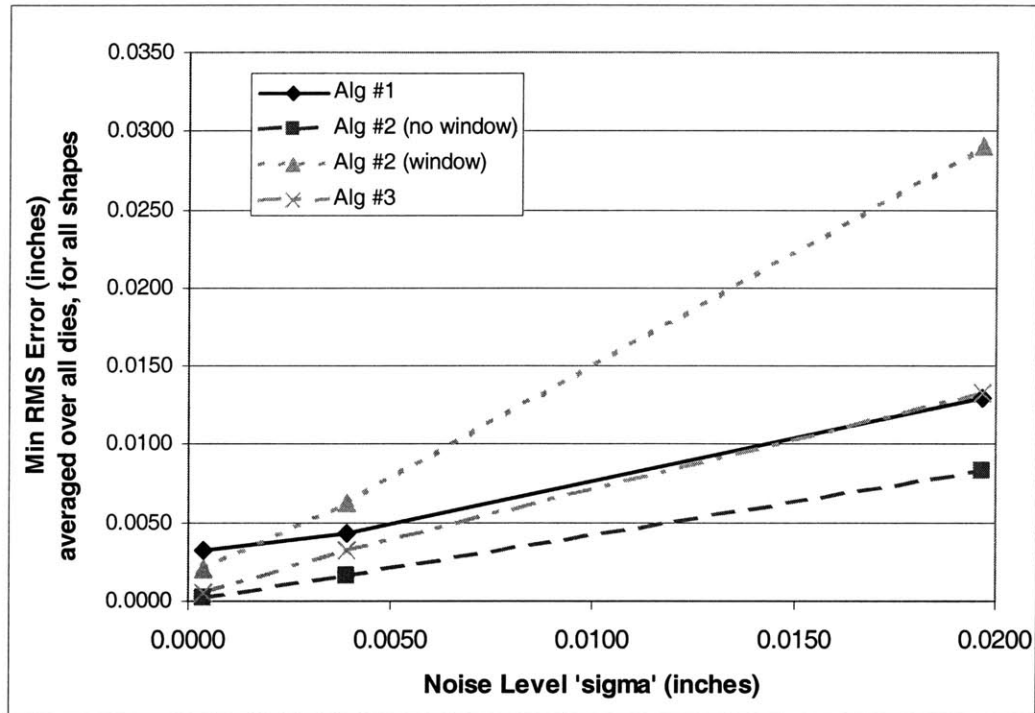


Figure 9-13: Error convergence during variation investigation simulation, noise excluded from identification

The error convergence trends for all algorithms displayed in Figure 9-13 are similar to those from simulations with noise included in identification. Again, algorithm #2 (no window) achieves the highest convergence for all noise levels. This is inconsistent with the full simulation results. This discrepancy is believed to result from comparing predicted dies to know dies versus comparing formed parts to desired parts. It is surprising to note that the values in Figure 9-13 are very close to those in Figure 9-10, which represents error values when noise is included in system identification. The close proximity of these values suggests that a noisy system identification may not affect the error convergence rate significantly.

Summary

In summary, this chapter presents simulations results that show algorithm #3 to converge to the lowest error value in a fixed number of trials. It also shows algorithm #3 to

amplify the effects of variation less than any other algorithm except #1, which has poor converges very slowly. The Monte-Carlo simulations did, however, provide some conflicting results for algorithm #2. These conflicts are most likely the result of the comparing dies to measure algorithm performance instead of comparing parts.

Chapter 10 : Conclusions - Guidelines for Future Work

Conclusions

This research has provided a greater understanding of the algorithms used to control shape in the stretch forming process. A new algorithm has been synthesized and the performance of all algorithms has been quantified.

No Coupling	<u>Algorithm #1</u> - No System ID - No Coupling	<u>Algorithm #3</u> - System ID - No Coupling
		<u>Algorithm #2</u> - System ID - Coupling via Convolution
Coupling	No Identification	Identification

Figure 10-1: Algorithm Framework

Greater understanding is achieved by viewing the algorithms through the principles of control theory and linear system theory. This analysis leads to a common framework under which all algorithms can be categorized. This framework is summarized in Figure 10-1. Here each of the three algorithms are described by whether they employ system identification and coupling via a convolution.

Algorithm #2 was previously viewed as a process with operations solely in the frequency domain. It is analyzed in this research from the spatial domain perspective. The

algorithm's operations are much easier to interpret from this perspective. This provides greater insight, which allows users of the algorithm to quantify the system coupling modeled by the algorithm. This also allows users to quality check the algorithm before resources are spent forming parts on dies improperly predicted by the algorithm.

Analysis and simulation of algorithms #1 and #2 also led to the synthesis of algorithm #3. This algorithm has the benefits of system identification without the difficulties associated with the convolution and blind de-convolution process of algorithm #2. Some difficulties are encountered with algorithm #2, specifically the issue of choosing reference points for dies and parts. However, a heuristic is developed which successfully addresses this issue.

Several ABAQUS and Matlab programs were developed to help quantify algorithm performance through simulation. Simulations were run of the lab scale stretch forming trials to investigate how close each algorithm converges to the desired shape in the first four forming iterations. Additional Matlab, Monte-Carlo simulations provide insight into how process variation affects the performance of each algorithm.

The simulations conclude that algorithm #3 has the best all-around performance. It most often produces parts that are closer to the desired part shape. It amplifies the effects of process variation less than algorithm #2 and is much easier to implement as well.

Algorithm #1 is arguably simpler than algorithm #3, and it does actually amplify variation effects less. However, its performance lags significantly behind that of algorithm #3. To achieve the benefits of both, a strategy that uses algorithm #1 to predict the second die and algorithm #3 to predict the third and fourth die is suggested.

Future Work

The fact that average z-coordinates are arbitrary in defining the surface of a shape poses some difficulties to Algorithm #3. A heuristic has been developed to accommodate this issue. However, other improved methods may exist. One concept involves characterizing parts and dies by the mean and/or Gaussian curvature they possess at discrete locations about their surfaces. Algorithm #3 could then be applied to iterate on one or both of these quantities instead of z-coordinate values. In addition to addressing the referencing issue of algorithm #3, Gaussian curvature may prove to be a better indicator of shape quality than z-coordinate positions. In practice, having to deform a part (change its Gaussian curvature) so that it conforms to a surface may prove to be the cause of most production problems. Developing a part about a die surface (change its mean curvature) is arguably easier to do.

Algorithm #3 may also be improved by eliminating the biasing that is introduced during the current referencing heuristic. Some alternate approaches may include smoothing techniques applied to the surface that represents controller gain values. Such techniques could be used to eliminate 'spikes' in the surface that represents controller gains.

Blind de-convolution performed in algorithm #2 is highly sensitive to noise. However, de-convolution of a system that is known specifically does not have the same problems. A strategy could be pursued that fits the empirical response of the stretch forming system to that of a pre-defined system for which there exists a closed form solution. This may allow a de-convolution process with less sensitivity to noise.

References

- Boas, R.C., **Sequential Setup Mechanism Design for a Reconfigurable Sheet Metal Forming Die**, SM Thesis, Department of Mechanical Engineering, MIT, Feb.1997.
- Bloomenthal, Jules, **Introduction to Implicit Surfaces**, Morgan Kaufmann Publishers, Inc., San Francisco, 1997
- Bracewell, Ronald Newbold, **The Fourier Transform and its Applications**, McGraw-Hill, New York, c1978
- Brigham, E. Oran, **The Fast Fourier Transform and its Applications**, Prentice Hall, New Jersey, 1998
- Casey, James, **Exploring Curvature**, Braunschweig, 1996
- Crandal, S.H., Dahl, N.H. and Lardner, T.J., **An Introduction to the Mechanics of Solids**, 2nd Edition with SI Units, McGraw-Hill, Inc. London, 1987
- Eigen, G., **Smoothing Methods for Discrete Die Forming**, SM Thesis, Department of Mechanical Engineering, MIT, May 1992.
- Farin, Gerald E., **Geometric Modeling: Algorithms and New Trends**, Society for Industrial and Applied Mathematics, Philadelphia, c1987
- Gallier, Jean H., **Curves and Surfaces in Geometric Modeling : Theory and Algorithms**, Morgan Kaufmann Publishers, San Francisco, c2000.
- Goh, John, **Deep Drawing Performance of a Discrete Die Surface**, S.M. Thesis, Department of Mechanical Engineering, MIT, May 1984.
- Grodzinsky M., **Testing a Closed Loop Forming Algorithm on a Part Created by Discrete Die Stretch Forming**, M.S. Thesis, Department of Electrical Engineering; MIT, 1996
- Hardt, D. E., Webb, R. D., **Sheet Metal Die Forming Using Closed-Loop Shape Control**, Annals of the CIRP, Vol 31/1/1982
- Hardt, D.E., Boyce, M. C., Osterhout, K. B., Karafilis, A., **A Flexible Forming System for Sheet Metal**, 18th Annual NSF Conference on Design and Manufacturing Systems Research, Atlanta, GA, Jan. 1992
- Hardt, D.E., Norfleet, W.N. Parris, A., Valentin, V.M., Valjavec, M., **In Process Control of Strain in a Stretch Forming Process**, Proc. ASME Symposium on Advances in Metal Forming, Orlando, Nov. 2000.

Hardt, D.E., Norfleet, W.N. Parris, A., Valentin, V.M., Valjavec, M., **In Process Control of Strain in a Stretch Forming Process**, to be published. Trans ASME Journal of Engineering Materials and Technology, 2001.

Jennings, Walter, **First Course in Numerical Methods**, Macmillan, New York, c1964

Kalpakjian; **Manufacturing Engineering and Technology**, 3rd Ed. , Adison Wesley , New York, 1995.

Karafillis, A.P., **Tooling design for three-dimensional sheet metal forming using finite element analysis**, Ph.D. Thesis, Department of Mechanical Engineering, MIT, 1994

Karu, Zohar Z., **Signals and Systems Made Ridiculously Simple**, ZiZi Press, Cambridge, MA, 1999

Knapke, J., **Forming with a Variable Configuration Die**, S.M. Thesis, Department of Mechanical Engineering, MIT, Sept. 1988

Montgomery, Douglas C., **Introduction to Statistical Quality Control**, Wiley, New York, c1985

Ogata, K.; **Modern Control Engineering** 3rd Ed, Prentice Hall, New Jersey, 1997

Olsen, B.A., **Die Forming of Sheet Metal Using Discrete Surfaces**, S.M. Thesis, Department of Mechanical Engineering, MIT, June 1980.

Osterhout, K.B., **Design and Control of a Flexible Process for Three Dimensional Sheet Metal Forming**, Ph.Dd Thesis, Department of Mechanical Engineering, MIT, 1991

Proakis, Jahn G. and Manolakis, Dimitris, **Digital Signal Processing, Principles, Algorithms and Applications**, Prentice Hall, New Jersey, 1996

Papazian, John M., David Hoitsma, Lembit Kutt, John Melnichuk, Jerrell Nardiello, Allan Pifko, and Robert C. Schwarz, "Reconfigurable Tooling for Sheet Metal Forming," in **Sheet Metal Forming Technology** edited by Mahmoud Y. Demeri, The Minerals, Metals & Materials Society, Warrendale, PA, 1999, pp. 23-39.

Parris, Andrew, **Precision Stretch forming for Precision Assembly**, Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1996.

Robinson, R., **Design of a Variable Configuration Forming Die**, S.M. Thesis, Department of Mechanical Engineering, MIT, Jan. 1987.

Siu, George Tsz-Sin, **Cycle to Cycle Feedback of Manufacturing Processes**, S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 2001.

Socrate, Simona, and Boyce, Mary C., “**A Finite Element Based Die Design Algorithm for Sheet Metal Forming on Reconfigurable Tools**”, Proc. ASME Symposium on Advances in Metal Forming, Orlando, Nov. 2000.

Socrate, S., Narisarunukul, N., and Boyce, M. C., **Reconfigurable Tooling for Flexible Fabrication**, ARPA flexible Fabrication Program (ONR Cooperative Agreement N0014-95-003), First Annual Report and Quarterly Reports, 1996.

Strang, Gilbert, **Introduction to Applied Mathematics**, Wellesley-Cambridge Press, Mass. 1986.

Trefethen, Lloyd N. and Bau, David, **Numerical Linear Algebra**, Society for Industrial and Applied Mathematics, Philadelphia, 1997

Valentin, V., **Implementing Strain Control in Stretch Forming**, SM Thesis, MIT Department of Mechanical Engineering, Jan. 1999

Valjavec, M. **Closed-Loop Shape Control in Aerospace Panel Forming Systems**, Ph.D. Thesis, MIT Department of Mechanical Engineering, Jan. 1999.

Valjavec, M. and Hardt, D.E., **Closed-loop Shape Control of the Stretch Forming Process over a Reconfigurable Tool: Precision Airframe Skin Fabrication**, Proc. ASME Symposium on Advances in Metal Forming, Nashville, Nov. 1999.

Walczyk, D., **Machine Design for Flexible Sheet Metal Forming**, Ph.D. Department of Mechanical Engineering, MIT, May 1995.

Walczyk, D. and Hardt, D.E., **Design and Analysis of Reconfigurable Discrete Dies for Sheet Metal Forming**, SME Journal of Manufacturing Systems Fall 1999.

Walczyk, D. and Hardt, D.E., **A Comparison of Rapid Fabrication Methods for Sheet Metal Forming Dies**, ASME Journal of Manufacturing Science and Engineering 121, May 1999, pp 214-224.

Webb, Richard Davis, **Adaptive Control of a Flexible Die System for Forming Sheet Metal Parts**, MS Thesis, Department of Mechanical Engineer, MIT, 1981.

Webb, Richard Davis, **Spatial Frequency Based Closed-Loop Control of Sheet Metal Forming**, Ph.D. Thesis, Department of Mechanical Engineering, MIT, 1987