DEVELOPMENT OF TAILORED PREFORM PROCESSING TECHNOLOGY FOR NET-SHAPE MANUFACTURING OF LARGE MONOLITHIC STRUCTURES

A Dissertation
Presented to
the Faculty of the Graduate School
University of Missouri-Columbia

In Partial Fulfillment
of the Requirement for the Degree

Doctor of Philosophy

---

by
SURENDRA CHITTI BABU

Dr. A Sherif El-Gizawy, Dissertation Supervisor

DECEMBER 2007

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

DEVELOPMENT OF TAILORED PREFORM PROCESSING TECHNOLOGY FOR NET-SHAPE MANUFACTURING OF LARGE MONOLITHIC STRUCTURES

presented by Surendra Chitti Babu,

a candidate for the degree of Doctor of Philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

_____

Dr. A. Sherif El-Gizawy, Dissertation Supervisor

_____

Dr. Uee Wan Cho

_____

Dr. Sanjeev Khanna

_____

Dr. Yuyi Lin

_____

Dr. Hani Salim, Outside Member

_____

Dr. Robert Winholtz

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. A. Sherif El-Gizawy, for providing me the opportunity to work in this research. I am grateful for his expert guidance, continuous support, advice and mentoring during the course of my graduate study in University of Missouri - Columbia. My thanks are also extended to my committee members, Dr. Uee Wan Cho, Dr. Sanjeev Khanna, Dr. Yuyi Lin, Dr. Hani Salim and Dr. Robert Winholtz for their guidance and kind review of my dissertation

I would like to acknowledge the financial support of Advanced Manufacturing R&D, Phantom Works of The Boeing Company for the present research. My appreciations are extended to Mr. Edward Gerding, Mr. Richard Liedrich and Dr. K. K. Sankaran of The Boeing Company for their technical assistance and valuable discussions in the experimental study. Appreciation is also extended to Dr. D. L. Carroll of University of Illinois Urbana-Champaign for the Genetic Algorithms source code and his expert advice.

I would like to thank the staff members of the Mechanical and Aerospace Engineering Department: Rex Gish, Richard Oberto and Brian Samuels for their help with experimental set up; Melanie Gerlach for her help with all the paper work.

Last, but not least, I wish to express my love and appreciation to my family and friends for their continuous support and encouragement through these years.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Tables **Page**

DEVELOPMENT OF TAILORED PREFORM PROCESSING TECHNOLOGY FOR NET-SHAPE MANUFACTURING OF LARGE MONOLITHIC STRUCTURES

ABSTRACT

The aim of the present research effort involves exploration of a new processing approach, 'Tailored Preform Processing (TPP)' technology for net-shape manufacturing of monolithic structures for the transportation industry. The proposed technology combines friction stir welding (FSW) and forming in order to tailor the properties of the preform to meet design requirements and provide net-shape preforms, which can subsequently be turned into finished structural parts through light machining or other standard processes. TPP can provide several advantages including weight reduction, part count reduction, improved damage tolerance, improved material and energy utilization, and cost saving. The research focuses on understanding of the process behavior and the effects of various process parameters on the properties and integrity of the produced structure during different stages of the TPP approach. Objectives of the proposed work also include providing capability for robust process design, prediction and characterization of process-induced damage and properties of the finished structure. These objectives are accomplished through innovative solution to the Tailored Perform Processing problems using the presented experimental and virtual models. Finite Element

Method was used to model 3-D friction stir welding and forming processes. Numerical model of friction stir welding was carried out using Fully Coupled-Temperature Displacement Analysis. Forming of the Tailor Welded Blanks was modeled using ABAQUS/Explicit followed by the Springback simulation using ABAQUS/ Standard. FEA models were used to investigate the process behavior and effects of various parameters on the properties and integrity of produced structure. An optimization scheme based on Genetic Algorithm is integrated with the numerical models to provide the optimal process conditions for quality and cost effective production.

CHAPTER ONE

PROBLEM STATEMENT AND OBJECTIVES

1.1 Sheet Metal Forming

Structural parts for transportation industry requiring tailored properties to meet stiffness and durability specifications are commonly produced by mechanically joining discrete forged or cast components. This processing approach is very tedious and neither cost nor quality effective. Components made by casting or forging are expensive and suffer from inferior properties. Additional costs are also required for assembly and alignments of the individual components which include tooling, fasteners installation and sealing ever more assembled products. Mechanical fasteners used also add weight penalty to the design.

1.2 Process-Induced Damage

Process-induced damage can be categorized into two major groups. The first one involves failure during production and induces wrinkling and tearing (splitting). Wrinkling often occurs in the flange area of drawn products such as cups or pans. During the drawing process, the compressive hoop stresses in the flange area cause the flange to buckle and form wrinkles. Applying sufficient support to the blank in the form of holding pressure during deformation would eliminate this problem. On the other hand, excessive blank holding pressure could also lead to tearing or splitting in some larger deformation area, such as the area adjacent to the punch or die corners. Large deformation combined with high contact friction could lead to excessive thinning in some critical regions, which

can become potential failure spots during deformation. Another problem is the post forming elastic deformation recovery or "springback". This process-induced shape distortion is one of the major problems in sheet metal forming operations. Distortions due to elastic deformation recovery after removing the forming loads alter the product geometry from the original design, and cause difficulties in joining and assembly of sheet products such as in autobody assembly. Distortion due to elastic recovery known as springback in sheet metal forming is a function of material properties, process parameters, and die design. The larger the ratio of yield strength to the elastic modulus, the larger the springback found. Process parameters such as blank holding pressure, ram speed, sheet thickness, and interface friction condition have been observed to affect the springback phenomenon. Die design variables (which include punch radius and die lip radius, and the gap between the punch and the die) were also found to have direct effects on the amount of distortion caused by springback. Simple remedies such as overbending, indentation, and heating have been used to rectify this problem, but these methods are limited to very simple shapes. Thus, the development of an effective and efficient method to predict and control process-induced damage in sheet forming is of great interest to manufacturing industries.

1.3 Tailored Preform Processing

With the increasingly rising costs and material lead times for various metallic product forms, Tailored Preforms Processing (TPP) approach offers an attractive alternative to both processing approaches currently used in industry. The present project will explore a new processing approach involving a combination of friction stir welding

(FSW) and forming in order to tailor the properties of the preform to meet design requirements and provide net-shape parts, which can subsequently be turned into finished structural parts through light machining or other standard processes. TPP can provide several advantages including weight reduction, part count reduction, improved damage tolerance, improved material and energy utilization, and cost saving.

1.4 Statement of Problem

There is little information available about the potentials and application of the proposed approach. Effects of different FSW design on performance and formability of the preforms need to be investigated. Properties and process-induced defects evolved during TPP need to be characterized and managed using both experimental and numerical models. Also techniques and methods for process and tooling design for tailored prefom processing approach have to be developed in order to realize the new TPP in production. The present proposal focuses on exploratory and advanced development of TPP for manufacturing large monolithic parts fabricated out of 2xxx and 7xxx series of aluminum alloys. It addresses material characterization, process-induced damage, process verification and identification of transition pathways. The proposed investigation will provide capability for robust process, prediction and characterization of process-induced damage, and properties of the finished structure. Scaled down prototypes of actual aerospace structures will be used in the feasibility investigation.

1.5 Work Objectives

The followings are the specific objectives of the proposed work:

1. Understanding the Process Behavior during different stages of the TPP Approach.
2. Development of 3D Numerical Model based on Finite Element Method to describe material responses to process environment during Friction Stir Welding (FSW).
3. Development of 3D Numerical Model based on Finite Element Method to determine the effects of different process parameters on the formability and ductility of final product and determine process-induced damages of welded preforms.
4. Develop an Optimization Model to determine optimal process parameters of forming process to enhance the formability of TPP. The link between the simulation schemes and the optimization algorithm will be established.
5. Verify the developed numerical models using experimental measurements of Friction Stir Welding, Stretch Forming and Deep Drawing processes.

1.6 Organization of Dissertation

This dissertation is divided into seven chapters. Following the *Introduction* in this first chapter, here are the brief outlines for Chapters Two through Seven:

Chapter Two is the *Literature Review*. The literature about research on friction stir welding, sheet metal forming and process-induced damage problem in the past years is reviewed in this chapter.

Chapters Three through Five contain the methodology employed in this research, including: *Experimental Investigation* (Chapter Three); *Numerical Modeling* (Chapter Four); and *Optimal Process Design* (Chapter Five). The set up for experiments, the Finite Element Method (FEM) used in simulations, the Genetic Algorithms used in optimization are discussed in these chapters. Also, the integration between the FEM simulations and Genetic Algorithms for developing a process optimization system is described in Chapter Five.

In Chapter Six, results from simulation and optimization are presented and compared with the experimental verification. Besides the springback phenomenon, associated process-induced defects such as thickness distribution and weld line movement are also discussed.

Chapter Seven presents conclusions and recommendations. Final comments on this research are made, followed by recommendations for further studies.

CHAPTER TWO

LITERATURE REVIEW

2.1 Tailor Welded Blanks

The use of new processing concepts and advanced materials is always of interest to aerospace industry because of the continuous interest in weight and cost reduction, better utilization of energy and materials, and endurance of the manufactured products. One such concept is Tailored Preform Processing (TPP) which is introduced in the present research. The new approach is based on the concept of tailored welded blanks (TWB) developed for automotive industry over the last 10 years [1-3]. TWB concept is used in automotive industry to replace multiple stamped parts which are fabricated separately then joined together to represent the final finished structure. TWB is composed of two or more sheets of metal of different gages and /or properties welded together prior to one stamping step to produce the finished panel.

The main advantage of using Tailored Blanks is to have specific characteristics at particular parts of the blank in order to reduce the material weight and costs and increase strength. Some of the other advantages are [4]:

- Improvement of tolerances, since the different parts of a product is welded together before forming using a continuous welding process.
- Reduction of the amount of scrap that arises because of the irregular shapes of the product. Smaller pieces of materials that form an irregular blank shape can be nested easily for better material utilization.
- Reduction of the number of parts to be assembled, which results in simpler logistics
- Reduction of press-handling including dies and installations.
- Improvement of crash durability

2.2 Process-induced damages

Process-induced damages common in conventional sheet forming of uniform blanks also apply to forming of tailored blanks. Process parameters such as blank holding pressure, ram speed, sheet thickness, and interface friction condition have been observed to affect the process-induced damages [5-6]. Die design variables which include punch radius and die lip radius, gap between the punch and the die were also found to have direct effect on the amount of distortion. Wrinkles frequently develop in the walls of sheet formed products as result of high compressive hoop stresses. Ulsoy and Demeri [7] presented an approach to eliminate process-induced damage through an effective on – line process control. Their approach includes modeling of sheet forming process, design of the process controller, and determination of the optimal punch force trajectory. Rui et al [8] used a fuzzy neural network based expert system for optimization of the blank holder forces in order to eliminate process-induced damage (springback, localized thinning, and wrinkles) in sheet forming processes. El-Gizawy [6] presented a numerical model to characterize and manage process-induced damage in sheet forming processes. His experimental results verified the model. Kridli and El-Gizawy [9] investigated hot forming of Aluminum–Lithium alloy under super plastic condition. They presented their results in formability maps that relate formability, and evolved microstructure and defects to the process parameters.

The most common process-induced damages affecting the formability of tailor welded blanks are springback, localized thinning and wrinkling.

2.2.1 Springback

Springback is a process-induced damage, which is basically a deviation from the original shape of the formed product due to elastic recovery of deformation once the forming loads are removed. The mechanics behind this phenomenon is illustrated by a simple bending operation (Figure 2.1, where R_b and R_f are the bending radii before and after springback respectively and α_b and α_f are angles before and after springback respectively). The stress distribution across the thickness of the sheet is not uniform (Figure 2.1a). While the portion near the inner and outer most surfaces of the bend are undergoing stresses larger than the yield stress (producing plastic deformation), parts of the sheet near the neutral line are subjected to much lower stresses below the elastic limit of the material. This leads to recovery of the elastic deformation upon removing the forming load and hence alteration of the shape geometry. After springback, the stress across the thickness (h) is redistributed, and a new residual stress pattern is produced (Figure 2.1b). In simple bending processes, the springback pattern (Figure 2.2) is simple and easy to predict [10-11].

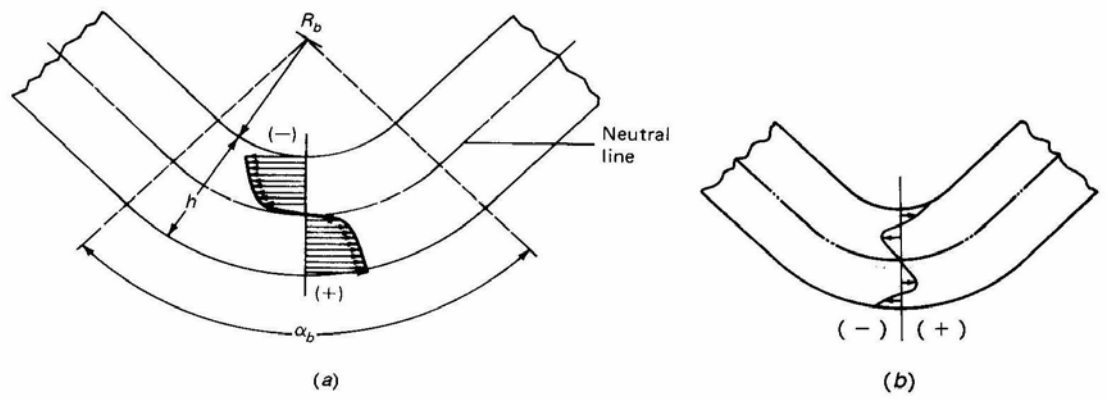


Figure 2.1 (a) The stress distribution across the thickness during bending operation [12]
(b) The residual stress pattern across the thickness after springback [12].

However, in most of the practical productions, forming processes involve a significant number of stretching and bending operations with a wide variety of shapes, which increases the complexity of the springback phenomena and causes essentially uncontrollable shape distortions.

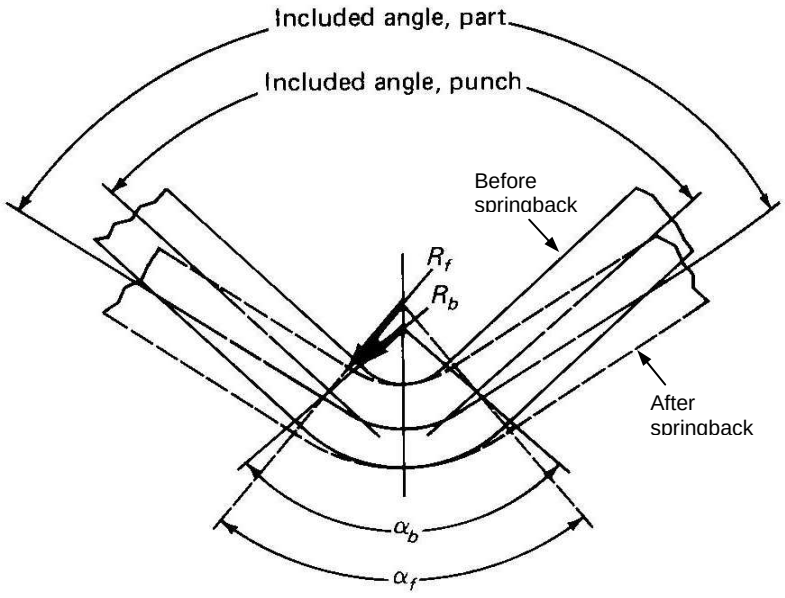


Figure 2.2 Differences in bending angle and radius before and after springback [12].

2.2.2 Sheet thinning

Thickness distribution is one of the major quality characteristics in sheet metal products. Uniform thickness distribution is always desired. However sheet metal thickness at the end of deep drawing process is not always same as of initial thickness. Different sections of sheet undergo considerable thickening and thinning due to various stresses acting at different sections. Excessive sheet thinning is of great importance to the quality of final product since it will ultimately lead to fracture/tearing of final product.

Researches have been carried out to predict the fracture (excessive thinning) of final product in deep drawing [13-14].

2.2.3 Wrinkling

In sheet metal forming process, Wrinkles frequently develop in the walls of sheet formed products as a result of high compressive hoop stresses. The conditions for the onset of wrinkling are controlled by the local geometry (sheet curvature) and the current state of stress. Insufficient blank holder stress leads to the onset of wrinkling. The development of metal sheets with higher yield strengths and lower thicknesses favors the occurrence of wrinkling, which explains that this defect becomes a major problem in sheet metal stamping industries.

Correiu and Ferron [15] proposed a technique which combines finite element simulation and an analytical method to predict the onset of wrinkles. Their results were in fair agreement with the published experimental results. Sheng, Jirathearanat and Altan have developed an adaptive FEM simulation for predicting the magnitude of blank holding pressure during drawing of conical cups that prevent failure due to wrinkling and tearing [16]. Krishnan and Cao [17] proposed a strategy based on FEM simulation and a deterministic Auto Regressive Moving Average (ARMA) model for estimation of blank holding force trajectories needed to suppress wrinkles and avoid tearing. The experimental observations indicate the developments of side wall wrinkling when the strategy was applied to forming a rectangular pan. Therefore, a better optimization search technique needs to consider all variables in the system to make the strategy really robust. Wang, Chen and Li [18] developed a closed loop control strategy based on optimal blank

holding force trajectories for the best formability window to avoid wrinkling and tearing. Mostly recently, Yagami, Manabe and Yamanchi [19] investigated the effect the controlling blank holder motion on elimination of wrinkle. The experimental verification of their approach indicated that wrinkle elimination can only be successful if the developed wrinkles are within certain heights. Ulsoy and Demeri [20] presented an approach to eliminate process-induced damage through an effective on – line process control. Their approach includes modeling of sheet forming process, design of the process controller, and determination of the optimal punch force trajectory.

Most of the reported works were based on experimental results. Even though quite a bit of research is done for springback, sheet thinning and wrinkling, not much work has been devoted in Tailor Welded Blanks. Furthermore none of the reported works have considered the possible interaction between the causes of wrinkling, excessive thinning and the springback phenomenon.

The new concept of TPP extends TWB concept to aerospace structure to include combination of friction stir welding (FSW) and forming (stretch forming, deep drawing, and hot forming) in order to tailor the properties of the preform to meet design requirements. The weld design in TWB concept is limited to butt joint type. The design requirements and constraints associated with the TPP concept represent new challenges that do not exist with TWB manufacturing concept and were not addressed before. Large portion of the published work deals primarily with TWB for auto-body structures. Nevertheless, these findings and results could also help in the development of the science-based technology for the new concept of TPP for aerospace applications. The

proposed work is intended to be comprehensive, covering both the friction stir welding (FSW) stage and the forming stage.

2.3 Friction Stir Welding and TPP Concept.

Friction Stir Welding (FSW) is a relatively new welding process developed by The Welding Institute (TWI) in 1991 [21]. Friction stir welding (FSW) is a solid state welding process where localized deformation at the joint interface establishes the bond between the base metals. In this process, a rotating tool generates heat and deformation at the joint interface. The interface temperature never exceeds the melting point of the base metals (maximum 90% of melting temperature). Therefore FSW does not involve liquid phase transformation. This makes the process superior to all other welding processes that result in unfavorable microstructures and properties associated with solidification mechanisms that take place during fusion welding. FSW has many other significant advantages [22] including controlled properties and microstructure, improved material utilization (light weight structures), improved energy utilization (only 2.5% of energy needed for lased welding), and reduced harmful effects on environment.

FSW process set up (Fig 2.3) consists of a rotating cylindrical shouldered tool, two sheets to be welded and a backing plate. The plates to be welded are fixed on top of the backing plate. FSW cycle consists of four stages (Fig 2.3): plunge stage, dwell stage, welding stage and pull out stage.

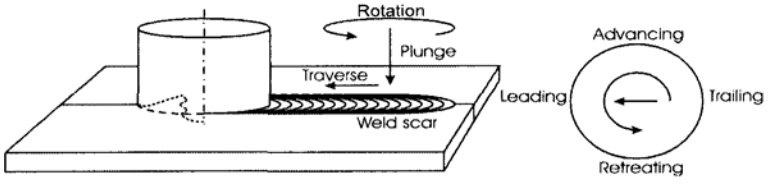


Figure 2.3 Schematic of Friction Stir Welding

During the plunge stage, the rotating tool with shoulder and probe is brought into contact at a point along the joint-line. The process proceeds by rotating the tool at high angular speeds and plunging the pin into the blank until the shoulder makes full contact with blank surface.

In the dwell stage, tool is held in the plunging position while still rotating. The tool is let to rotate at this location for few seconds to locally plasticize the material and increase the temperature.

In the welding stage, the tool then travels along the joint-line, high forging force is applied to maintain full contact between shoulder and work piece surface. In FSW, heat is produced primarily due to friction between the shoulder of the tool and blank. The increase in temperature softens the material, and allows the rotating tool to mechanically stir the softened material flowing to the backside of the pin where it is consolidated to form a metallurgical bond. A process zone is produced by movement of material from the front of the pin to the back of the pin.

2.3.1 Simulation of Friction Stir Welding

Numerical studies of Friction stir welding is still in its infant stage. Two types of simulations of friction stir welding are possible; a) Computational Fluid Dynamics (CFD) and b) Computational Solid Mechanics (CSM). CFD analysis is more suited to analyze the material flow and heat generation due to the viscous dissipation of the softened material. CSM is ideal to study the generated stress during friction stir welding and to analyze the heat generated due to contact friction.

Considerable amount of research has been conducted on various aspects of friction stir welding. Research in FSW has focused on developing experimental, analytical, and numerical models in order to characterize the different zones in FSW. Several techniques have been used to model and analyze friction stir welding process like analytical [23], finite difference method [24], and finite element method [25-27].

Heat generated in FSW can be divided into three parts: frictional heat generated by the tool shoulder, frictional heat generated by the tool pin, and heat generated by material plastic deformation near the pin region.

Chao and Qi [28] used a constant heat flux input from tool shoulder/work piece to increase the temperature and trial and error method was used to adjust the heat input. Frigaard, Grong and Midling [29] developed a model for friction stir welding and heat input is assumed to be the frictional heat and co-efficient of friction and other conditions were adjusted to keep the temperature below the melting point. Zahedul, Khandkar, and Khan [30] modeled the friction stir welding with a moving heat source.

The above researches account only for the heat generated by tool shoulder and sheet interface and doesn't account for heat generated by tool pin and sheet interface and heat produced due the plastic deformation. In order to model the heat transfer in friction stir welding accurately, it is necessary to include heat generated at tool pin and sheet interface and also due to the plastic deformation.

Askari used CTH hydro-code based on finite volume to model the flow of the material on the assumption that the material sticks to the tool surface. [31]. Xu and Deng presented a 3D model in which no separation is allowed between the tool and sheet. This

method doesn't simulate the actual conditions in friction stir welding process as this would lead to unrealistic stress state [32].

Various parameters affect the performance of FSW like tool rotational speed, travel speed of the tool, plunge force, plunge depth, and tool design. These parameters affect the thermo-mechanical and metallurgical changes established during FSW which in turn are related to the evolved properties, microstructure, and process-induced damage in the course of the welding process. Effects of process parameters on the formability of friction stir welded blanks are unexplored. There hasn't been much research devoted to study these process parameters and evaluate them experimentally.

Most of previous works on modeling FSW used finite element based thermo-mechanical models and computational fluid dynamic models to estimate heat generated and material flow behavior during FSW. Many of the reported models included non realistic assumptions about interface friction mechanisms and the dynamics of the entire welding system in order to simplify the solution. Furthermore none of the reported results addressed the relation between FSW parameters and formability of the welded blanks that is crucial to the proposed TPP approach.

Friction Stir Welding is a good candidate for welding tailored blanks because of its capability to weld alloys that are normally not weldable using traditional joining processes. This helps to widen the scope of materials that can be used for fabrication process and thus improve the final product. Several researches have been conducted to study Friction Stir Welding of a wide range of dissimilar materials [33-40]. Dissimilar welds of Al 5083 and Al 6082 showed that the material lower strength should be placed on the advancing side [41]. Cavaliere et. al. investigated the mechanical and micro-

structural properties of 2024-T3 and 7075-T6 aluminum sheets. Fatigue and stress analysis had been performed at a weld velocity of 2.67mm/s, however the rotational velocity has not been disclosed [34]. The compressive flow behavior was studied for the friction stir welding of AA 2024-T3 and AA 7075-T7351 [33]. Friction stir welding reduces the yield strength under high strain-rate and quasi-static loading conditions. The microstructure of a dissimilar weld is distinctly different from the similar alloy weld [40]. The mutual mixing in dissimilar weld is intimate and far from complete even though bonding between the alloys is complete. No information on other welding parameters or the orientations of the alloys has been given.

Ductility is one of the important required characteristics in the final product, but not much research has been devoted to the ductility of friction stir welds. Also, numerical modeling of dissimilar friction stir welding has been unexplored. Numerical modeling can significantly help to reduce the need for repeated and complicated experimental tryouts with different process parameters during friction stir welding.

These limitations are addressed in the proposed work using numerical approach and experimental validation to include all control and uncontrolled (noise) parameters of FSW process. The results will lead to process maps that help in identifying process window for reaching specific properties and microstructure.

2.4 Sheet Metal Forming and TPP Concept.

A schematic of the deep drawing process is shown in Figure 2.4. During deep drawing, an initially flat blank is placed between the die and the blank holder. The punch then moves down to deform the blank into the desired shape. The accuracy of the final

shape of the product depends on the geometry of the tools, sheet material properties and the process parameters.

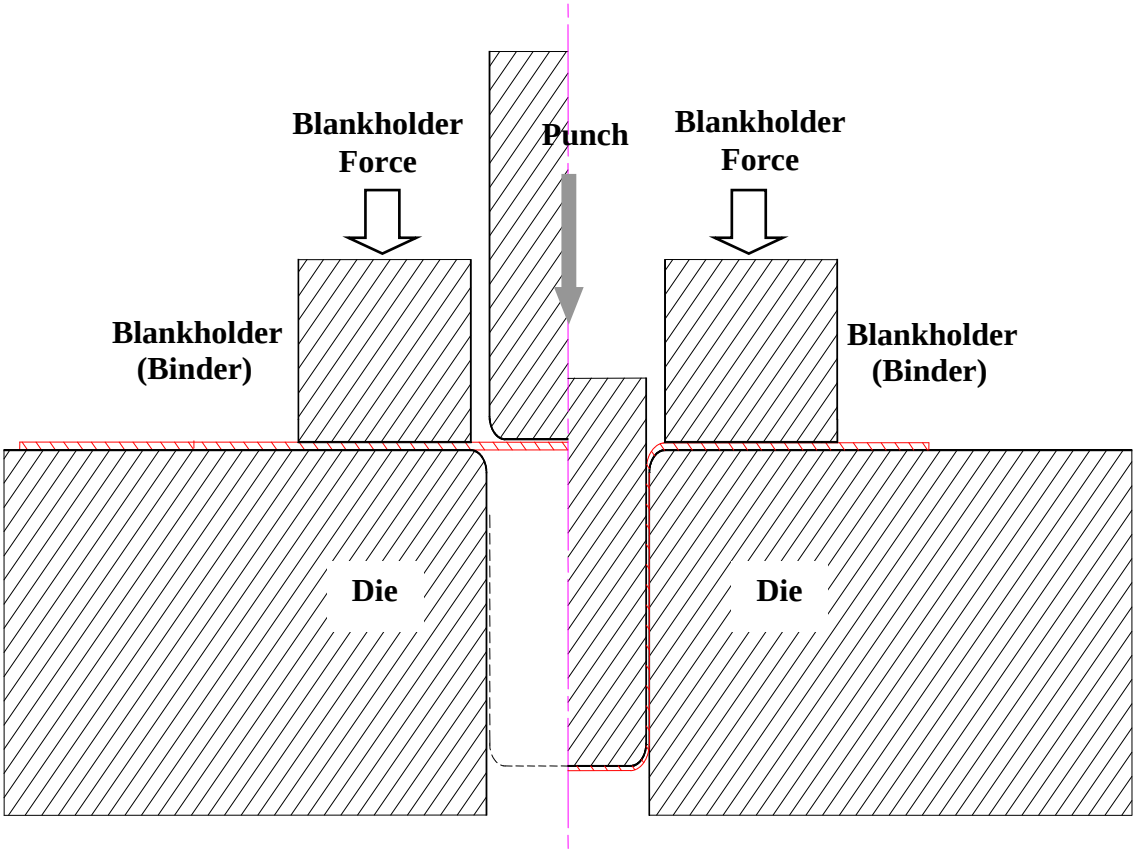


Figure 2.4 Initial tool setup for Sheet Metal Forming Process

2.4.1 Simulation of Forming Process

Modeling of Sheet metal forming processes have been carried out for several years; however forming process of tailor-welded blanks is different from same material blanks due to the different behavior in Tailor Welded Blanks.

A flat blank holder cannot prevent wrinkling in thinner base material. Possible adaptations to blank holder are stepped rigid, deformable blank holder, and segmented blankholder which allows individual force per segment [42].

Punch will need adapting only if the weld is located under the punch. In that case, a flat punch cannot prevent wrinkling. A stepped punch can be the solution to this problem. However, the displacement of weld line during the forming process can cause problems. Die can also be adapted for forming tailor-welded blanks, but only either die or punch and blank holder needs to be adapted [43].

Difference in thickness between two material gauges would normally cause the entire clamping load to fall on the thick gauge. A shim can be placed on the thin gauge to account for the thickness difference so even clamping pressure is distributed over the blank. [44-45].

Difference in the material properties would also lead non-uniform forming of the blank. Blank with lower strength material will deform more than the higher strength material and will lead to the weld line movement into the higher strength material. This would require the blank holder to segmented and apply higher force on thinner material and lower force on the thicker material.

2.4.2 Formability of Tailored Preforms

The material flow and formability are highly affected by the orientation of the weld line [45]. Due to the differences in the sheet thickness and strength between blanks, the blank may not deform uniformly during the forming process.

Tool and process design have to be done taking into consideration the orientation of the weld line. Shi [46] showed that the amount of weld movement towards the stronger material is a measure for the strain localization in the weaker base material. This weld movement strongly depends on the strength ratio and location of the base material.

Chan et al [47] evaluate formability based on three measures: failure mode, forming limit diagram, and minimum major strain. They found that the larger the gage mismatch of TWB, the lower the forming limit curve which is indicative of lower formability of the blank.

Saunders [48] shows the performance of a tailored blank to depend on the amount of weld movement. Prediction and control of weld movement is very important to predict the press formability. Mustafa A. Ahmedtoglu et.al [45] has employed a multi point pressure control technique to vary the pressure around the periphery of the blank holder. By increasing the BHF on the thinner side and reducing BHF on thicker side, better flow characteristics were obtained and weld line was kept on symmetry axis. Due to lower pressure, thicker side of blank was drawn in easily and a uniform metal flow was achieved.

Kinsey, Krishnan and Cao [49] used the methodology of segmented binder force to reduce wrinkling. Segmented binder, where the force applied to the tailor welded blank varies, has been successfully used to reduce the process induced damages and also ratio of binder force has been attempted to develop systematically.

Sijun [50] developed a simplified 2-D sectional analytical model that estimates the stress/strain distribution and BHF required for each side of the flange for dissimilar materials. Sijun later extended this model to 3-D analysis by superimposing the 2-D

analysis results and determined the required BHF for the whole panel. Using this technique, strain level is reduced significantly because of thicker material drawing in and compensating the stretching.

2.5 Optimization Methods

In recent years, the integration of FEM simulation with optimization techniques has become a major interest in manufacturing processes design. In process optimization, the FEM program serves as part of the objective function; the optimization techniques serve as the backbone of the system to evaluate the design and search for new values for various design parameters. A non-gradient optimization technique was applied by Kusiak [51] in conjunction with the FEM code FORGE2 for forging process optimization. In applications for springback control, Shu and Hung [52] and Chou and Hung [53] combined the FEM code, ABAQUS, with an optimization code, MOST, to minimize the springback in a channel bending operation. They applied the double-bend technique [54] to reduce the springback, which is actually evolved from the over bending method. The design variables were the bending die gap and plate opening. The objective function was defined as the angle difference between the two strikes. The MOST program used a gradient-based scheme in optimization, which is potentially to be trapped into a local minimum in a complicated design space. For more common shapes, the double-bend technique will not be useful, and a new definition of objective function is necessary. In research by Oral and Darendelier [55], a gradient-base optimization technique was used with FEM to design the die profile for a cylindrical bending process to compensate for

springback. The objective was defined by the "norm" of the deviations of the plate after springback from the desired shape as

$$f = \sqrt{\sum_{m=1}^{M} d_m^2} \qquad (2.1)$$

where M is the number of plate nodes and the d_m is the normal distance from the m^{th} node on the distorted product to the corresponding node on the desired shape. This appears to be a convenient way to describe the shape deviations for most general shapes, especially for those from which irregular distortions were created, but it might change with the number of nodes measured. Another method of quantifying the shape deviation based on the volume included between the shapes before and after springback proposed by Kutt *et al* [56] could be suitable for general three-dimensional shapes, but it may have difficulty for those cases with heavy curves and distortions. In recent years, a new optimization technique using Genetic Algorithms [57-59] was developed and has become popular in many fields of science and engineering research [60]. The characteristics of this algorithm (non-gradient based approach, group search, and random selection) help avoid the drawbacks of traditional methods, make it suitable for non-linear problems involving lots of design variables and constraints, such as the problems of sheet metal forming and springback. A study by El-Gizawy and Surendra Chitti Babu [61] showed the feasibility and effectiveness of combining Genetic Algorithms with FEM for the optimal process design to minimize the springback for a general sheet forming process.

Summary

From the literatures discussed above, various studies have been conducted on tailored welded blank and process-induced damages including springback, localized

thinning and wrinkling phenomenon. However, none of them have considered all factors involved in the process. Most of the models are limited to simple 2-D cases with only one or two parameters involved. The gradient-based searching technique applied for optimization could leave out the real optimum process conditions if more parameters are considered.

In this research, a new approach that combines FEM of Friction Stir Welding and Forming process and optimization in process design of sheet metal forming is proposed. Process parameters in both Friction Stir Welding process and Sheet Metal Forming process will be investigated from the formability point of view. The nonlinear FEM code, ABAQUS [62-64], will be used to simulate Friction Stir Welding, Stretch Forming and Deep Drawing processes and Springback analysis following Forming process. Genetic Algorithms will be integrated with FEM as a system so the process designs can be updated, analyzed, and improved during the evolution. Multiple process parameters will be considered in the design problem to improve the formability of the final formed product. A square pan drawing model will be used in validation of the developed methodology.

CHAPTER THREE

EXPERIMENTAL INVESTIGATION

The experimental phase of this research was conducted as part of the development for current research methodology. The objectives are: (1) to build a knowledge base of friction stir welding and sheet forming process with the case studies, and (2) to serve as a verification tool for the numerical results. A detail description and explanation about the tooling and procedures will be provided in the following sections.

3.1 Friction Stir Welding

3.1.1 Material

The materials investigated in this study are Al-Zn alloy 7075-T6 and Al-Cu alloy 2024-T3, which were provided by Boeing, St. Louis. The Al 7075 is among the highest strength alloys available in 7000 series. From ASM Metals Handbook [65, 66], the major chemical composition is shown in Table 3.1. Zinc is the major alloying element for 7075-T6, coupled with a smaller percentage of magnesium, which results in heat-treatable alloys of very high strength. For Al 2024-T3, Copper is the major alloying element with combination of chromium, Magnesium, Manganese, Silicon, Titanium, and Zinc (Table 3.2). With T6 temper, the 7075 is solution heat treated and artificially aged to a fully-hardened condition. A combination of light weight and high strength makes the 7075 suitable for aircraft structural parts and other highly stressed structural applications where exceptional strength and resistance to corrosion are required. It is also considered a good replacement for low carbon steel sheet since it matches the strength for half the weight of

a steel sheet. In sheet metal forming, due to its high tensile strength and low elongation, 7075-T6 is difficult to deform into complex shapes under both room temperature and elevated temperature; and causes more post-forming shape distortion problems than many common used metals. On the other hand, Al 2024-T3 has relatively low strength compared to 7075-T6, however has higher ductility which is advantageous for sheet metal forming operations. This offers better formability but less strength than 7075-T6. The larger elongation makes it more capable for deforming into complex shapes, but it may not be suitable for structural material where high strength is demanded.

Table 3.1 Major Chemical Composition (%) of Al 7075-T6.

Al	Cu	Mg	Zn	Cr	Mn	Si	Fe	Ti
Bal.	1.20-2.0	2.1-2.9	5.1-6.1	0.18-0.28	0.30(Max)	0.40(Max)	0.50(Max)	0.20(Max)

Table 3.2 Major Chemical Composition (%) of Al 2024-T3.

Al	Cu	Mg	Zn	Cr	Mn	Si	Fe	Ti
Bal.	3.8-4.9	1.2-1.8	0.25(Max)	0.1(Max)	0.3-0.9	0.50(Max)	0.50(Max)	0.15(Max)

3.1.2 The Material Properties

The ASTM Standard [67] tensile test specimens, with the dimensions shown in Figure 3.1, were machined from a 2mm thick sheet. The tensile specimens are designed to measure stress and strain with one-inch gage length.

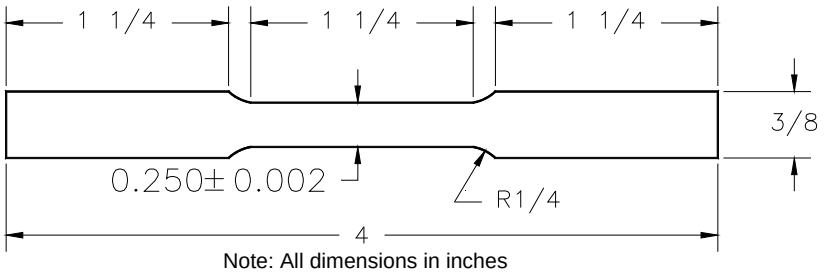


Figure 3.1 ASTM Standard tensile test specimen.

An MTS servo-hydraulic universal testing machine was used to conduct the uniaxial experiments. The machine was controlled by a 458.20 MicroConsole, and a 418.91 MicroProfiler^{TM} which was used to program the strain rate. The specimen was clamped between upper and lower grips, and a 1-inch extensometer was attached on the middle section to measure the strain. The MicroConsile was connected to a Data Acquisition System, with the process software LabVIEW, data such as load, displacement, and strain can be recorded in real time. All tests were conducted at room temperature.

From the test results, the major properties for both materials are listed in Table 3.3 below.

Table 3.3 The material properties of Al 7075-T6 and 2024-T3.

Property	7075-T6	2024-T3
Young's Modulus (GPa)	71.7	73.1
Poisson's Ratio	0.33	0.33
Density (Kg/M3)	2795.72	2780
Initial Yield Stress (MPa)	503	345
Ultimate Tensile Strength (MPa)	578	483
Elongation (%)	11	18
R value	0.68	0.65
K	756.6	430.2
n	0.0782	0.2288

3.1.3 Experimental Setup and Procedures

Description of the Welding Apparatus

Friction Stir Welding process was conducted experimental using the adapted milling machine (Figure 3.2). Fixtures (Figure 3.3 and 3.4) for Friction stir welding and the machine were designed and set up by a capstone design group under the supervision of Dr. El-Gizawy and I (Teaching Assistant of Senior Capstone Design).

Figure 3.2 Adapted Milling Machine for Friction Stir Welding Process

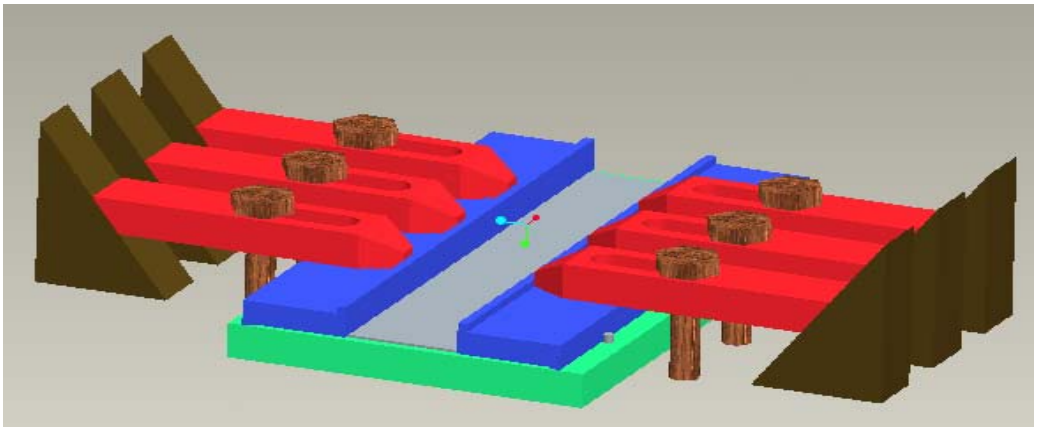


Fig.3.3 Model of Final Design for Friction Stir Welding Fixture

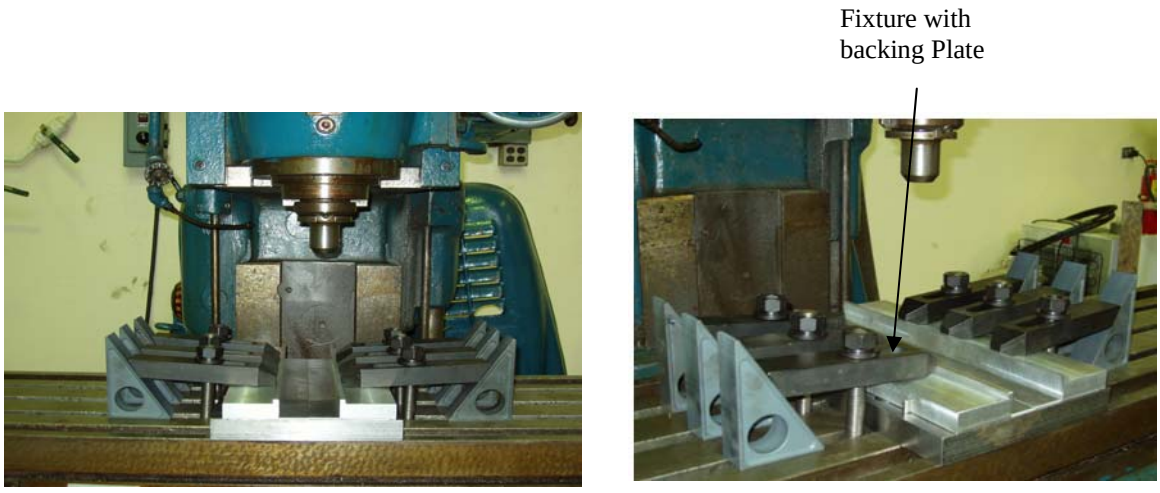


Figure 3.4 Adapted Milling Machine with fixture for Friction Stir Welding

The fixture consists of three main parts: the base plate, parallel bars, and clamps. The purpose of the base plate is to provide a strong and smooth surface on which the aluminum sheet sits. It needs to be capable of withstanding a 1500 pound load into it by the tool bit. A 1.25" thick slab of steel caped with three sheets of 0.040" titanium was chosen as the base plate materials. The steel was faced with a mill on all sides then surface ground on the welding surface to insure perfect smoothness with accurate distances, which are especially critical for this process. In order to align the aluminum sheet consistently and easily with the y-axis motion, pins were placed in the base plate. The base plate was keyed into the slots on the mill surface to align it with the table and make sure that it was completely square. The base plate was then bolted via t-nuts to the table to prevent it from shifting. The titanium sheets were added for the fact that they have low heat conduction properties for a metal and would not bond to the aluminum under high heat and pressure [8]. Low heat conduction is a very desirable quality in the

z-axis for friction stir welding. This allows more heat to pool around the weld, and not be pulled away, this allows for a faster welding process.

The parallel bars or parallels are used to distribute the load from the clamps evenly over the length of the aluminum sheets, and constraining the work pieces from separating in the case of a butt joint. The parallels were milled out of aluminum because of lightweight, to keep from marring the aluminum sheets, and to lightly pull heat away from the weld in the x-axis. The parallels were made to cover the entire length of the longest possible welding procedure, which is limited by the table's length. The width of the parallel provides a large amount clamping area, but will not interfere with the tool or the spindle of the machine. A lip of 0.060 inches, minimum workpiece thickness, was milled into the bottom of the parallels, to be able to handle separation force during welding. The lips on the parallels counteract the separation force that occurs during butt welds. Pulling heat away from the weld in the x-direction help distribute the heat evenly along the weld area, and helps from keeping heat to build up in the clamping system.

The clamps used are standard serrated end clamps with step blocks that are designed mills. This clamping system was easily available and very adjustable to account for different sized sheeting. The studs used to hold the clamps to the table are ¾" grade 7b steel and thread into t-nuts in the tables' existing channels. It should be noted that by using this clamping system, all block heights used in clamping must be equal to ensure equal forces [6].

The Tool Bit

The tool bit, Figure 3.5, contacts the metal and creates friction and stirs the material. Each tool has a shoulder whose rotation against the substrate generates most of the heat required for welding. The pin on the tool is plunged into the substrate and helps stir the metal in the solid state. The bits do show very good life when working with the aluminum, about 30 runs of 12" a piece for our applications. They are cleaned with a sodium hydroxide and water solution that removes all the aluminum and allows inspection of any wear of the bit.

Figure 3.5. Close-Up of New Tool Bit

Friction Stir Welding Process

Two sheets to be welded were butted up against each other and clamped down using the fixture. Rotational velocity and translational velocity of the tool are set in the adapted milling machine. The tool is then rotated and then slowly plunged into a work piece along the interface of the sheets. The tool creates frictional heat in the work piece until the material becomes plasticized. Heat generated by the mechanical mixing process and the adiabatic heat within the material cause the stirred materials to soften without reaching their melting point. This is a major advantage of friction stir welding. Once the

material becomes plasticized the tool traverses along a weld line to bond the two materials together as seen in Figure 2.3.

Plasticized material is deformed around the tool and is forged into place by the substantial downward axial force of the tool shoulder. Material then consolidates into the weld joint at the trailing edge of the tool leaving a solid phase bond between the two pieces [2].

Statistical Design of the Experiments

Design of Experiments (DOE) approach using the Taguchi Method was used to study the effects of process parameters simultaneously. Taguchi Method was used to reduce the number of tests to find optimal welding conditions and to create a knowledge base on the effects of process parameters. A set of experiments were designed to investigate the effects of the control process parameters that include rotational speed and feed rate (welding speed) on the important quality characteristics of the joint. Friction stir welding of two blanks of Al 2024-T3 was investigated with the Orthogonal L9 array in Table 3.4. Other process conditions used during the friction stir welding process are given in Table 3.5.

Table 3.4 L 9 Orthogonal Array for Al 2024-T3

#	Rot. Vel. (RPM)	Weld Vel.	
		(mm/sec)	(IPM)
2024-1	1300	0.9	2.125
2024-2	1300	1.53	3.625
2024-3	1300	2.43	5.75
2024-4	1300	3.23	7.625
2024-5	1045	2.43	5.75
2024-6	1045	1.53	3.625
2024-7	1045	3.23	7.625
2024-8	840	1.96	4.625
2024-9	840	2.43	5.75

Table 3.5: Process Parameters for FSW

Blank Dimensions (per blank): 100mm X 300mm X 2mm
Type of Weld: Butt Weld
Co-efficient of Friction: 0.3
Effective Plunge Depth: 2 mm
Plunge Time: 15 seconds
Dwell Time: 20 seconds

Friction stir welding of Al 7075-T6 was conducted to determine the optimal condition that produce a solid weld (Table 3.6).

Table 3.6 Process parameters for FSW of Al 7075-T6

#	Rot. Vel. (RPM)	Weld Vel.	
		(mm/sec)	(IPM)
7075-1	1300	0.9	2.125
7075-2	1045	3.23	7.625
7075-3	840	2.43	7.625

Table 3.7 L9 Orthogonal Array for Al 7075-T6 & Al 2024-T6

#	Rotational Velocity (RPM)	Welding Velocity (IPM)
Bi-A 1	675	3.625
Bi-A 2	675	5.75
Bi-A 3	675	7.625
Bi-A 4	840	3.625
Bi-A 5	840	5.75
Bi-A 6	840	7.625
Bi-A 7	1045	3.625
Bi-A 8	1045	5.75
Bi-A 9	1045	7.625

A systematic experimental study was conducted to understand the influence of process parameters during welding of dissimilar alloys on the ductility and formability of the friction welded blank. The three main process parameters in friction stir welding, namely, rotational velocity of the tool and weld feed of the tool are studied to determine their effects on the ductility and formability of the weld.

A few set of welds were created using various rotational speeds and weld feed to determine the window of feasible process parameters. Once the process window has been determined as shown below, the orthogonal L-9 table of test matrix (Table 3.7) with two extremes of rotational speed and welding feed were used in Friction stir welding of Al 7075-T6 and Al 2024-T6.

$$675\text{RPM} < \text{Rotational Velocity} < 1045\ \text{RPM}$$

$$3.625\ \text{IPM} < \text{Weld Velocity} < 7.625\ \text{RPM}$$

Tensile Testing

An MTS servo-hydraulic Universal Testing Machine was used to conduct the uni-axial experiments to determine the material properties of the weld. The machine was controlled by a 458.20 MicroConsole, and a 418.91 MicroProfiler^{TM} which was used to program the strain rate. The specimen (Figure 3.6) was clamped between upper and lower grips (Figure 3.7), and a 1-inch extensometer (Figure 3.8) was attached on the middle section to measure the strain.

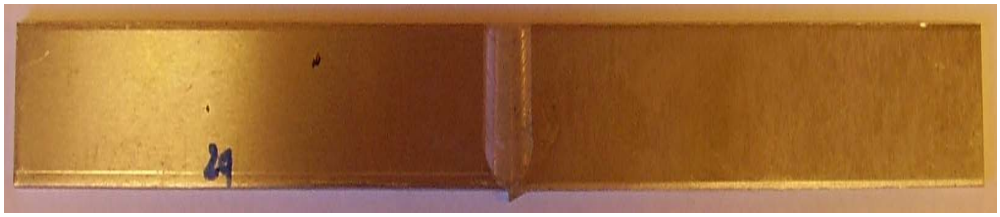


Figure 3.6 Sample used during Tensile Testing

The MicroConsole was connected to a Data Acquisition System, with the process software LabVIEW, data such as load, displacement, and strain can be recorded in real time. All tests were conducted at room temperature. This testing was conducted on all the samples that did not have any visible imperfections that deemed it unsatisfactory.

Mechanical properties of the friction stir welded joints were evaluated using standard tensile test procedures. The gage dimensions of the specimens were 25.4mm (1”) long and 19mm (3/4”) wide and with the weld zone running across the gage length. The test speed was kept constant at 1 cm /min for the duration of the test.

Figure 3.7 The hydraulic Linear Displacement Machine for Tensile Testing, with Extensometer in Place on the Clamped Sample.

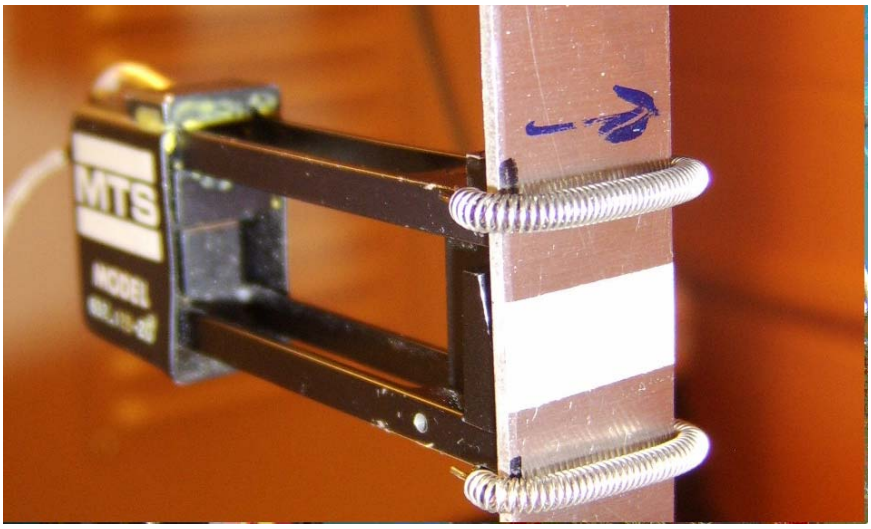


Figure 3.8 Close-up of Extensometer Covering the Welded Area That Records Displacement During Tensile Testing.

3.2 Stretch Forming

Stretch forming is conducted experimentally to determine the ductility and formability of friction stir welded blanks and determine the effects of rotating and weld velocity on ductility. Tool setup for stretch forming process with die, punch and blank holders are shown in Figure 3.9. The welded blank (Figure 3.10 & 3.11) is placed on the die and clamped rigidly by tightening the bolts on the blank holders. This in turn applies the force on the blanks resulting in blank holder force. High blank holder is applied so as to prevent the blanks from being drawn into the die when punch moves in to deform the blank. As a result, the blanks are stretched rather than drawn resulting in sheet thinning of the material and this might eventually lead to fracture.

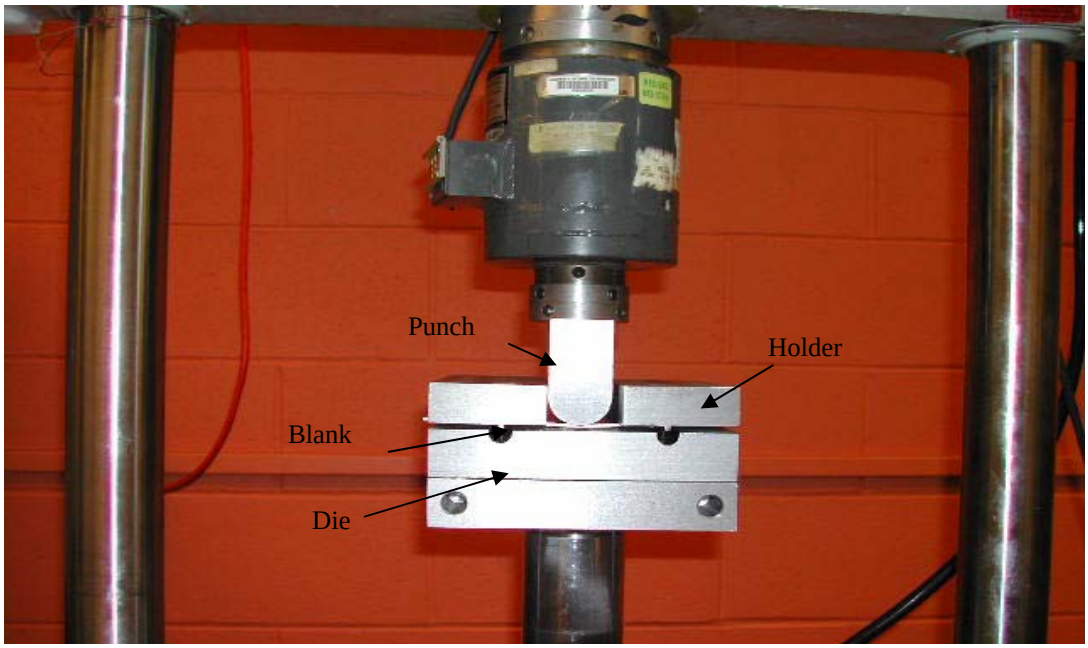


Figure 3.9 MTS Machine with Stretch Forming Fixture

Figure 3.10 Tool Set Up of Stretch Forming Process

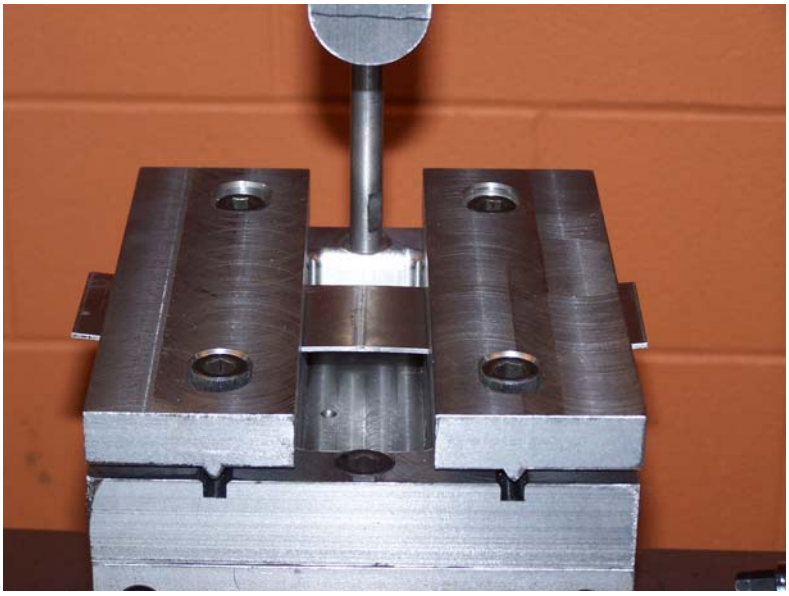


Figure 3.11 Tailor Welded Blank Set Up on Forming Die Before the Start of the Process

3.2.1 Experimental Setup and Procedures

Experimental procedures for stretch forming process are the same as deep drawing. The same MTS servo-hydraulic universal testing machine was used to conduct the stretch forming experiments. All tests were conducted under room temperature, and no heating equipment was used.

The experimental program included procedures for workpiece preparation, adjustment of parameters, the DAQ (Data Acquisition) system, execution of the drawing test, and post-process examination. The following discussion reviews the experimental plan and procedures for this research.

Workpiece Preparation

Sheets from friction stir welded blanks of Al 7075-T6 and Al 2024-T3 were used for stretch forming. Each workpiece was sheared to a 50.8mm by 203.2mm (2in×8in) size from 431.8mm by 203.2mm (12in×8in) blanks.

Adjustment of the Parameters

In this step, the prepared workpiece was set on forming die and center of the weld was aligned with the center of the die. The blank holder and spring plate were then carefully aligned and placed upon the workpiece. Bolts for blank holder were gradually and evenly tightened to rigidly clamp the work piece. The magnitude and the profile of the punching speed were specified by programming on the MicroProfiler.

Setup DAQ System

Before the execution of the process, the DAQ system was checked. By using the processing software LabVIEW, the signals from the MicroConsole were processed while recording. In the mean time, the selected data (load, displacement, and voltage) were plotted on the monitor to help evaluate the whole process, and identify the potential problems. The data file names and sample rate were properly set before each test was conducted.

Execution of the Drawing Process

Several stretch-forming experiments are carried to determine the optimal conditions among the different cases in Table 3.7. Stretch-forming experiments were

conducted with the final objective of improved ductility and formability and reduced process-induced damages. Each forming process began at the position where the punch touched the upper surface of the blank. During the stretch forming process, the whole die set (including the blank and the holder set) moved up towards the stationary punch. The process was terminated once the fracture occurred. The force on the punch was sensed by the load cell to which the punch was attached. The acquired data (including the load, displacement, and time) were recorded by the data acquisition system throughout the process at 0.005 second intervals. The whole die set was returned to "zero" (starting position) before being cleared from the punch.

All stretch forming processes were conducted with a constant punch velocity of 10mm/min and the blanks are rigidly clamped using the blank holder. The 'V' shape notch in the blank holder helps to lock the blank into the groove in the die, thus preventing the blank from slipping as the punch moves in to deform the blank.

3.3 Deep Drawing

A square pan model modified from the NUMISHEET'93 [22] benchmark (Deep Drawing of a Square Box) was used in current study. The major dimensions and tooling setup are shown in Figure 3.12. Similar models have also been used in [20, 21, 26, 68-72]. The square box drawing process is the most basic process among the nonsymmetrical deep drawing processes with which deformation mechanics of other similar processes can be qualitatively assessed. While most analytical studies of the deep drawing processes concentrate on the circular shapes, it is extremely difficult to develop analytical models for noncircular deep drawing processes. Due to the geometrical

difference, the drawability index, limit drawing ratio, and associated theoretical developments for the circular cup, the drawing process appears to be inappropriate to apply to the square cup drawing process [68]. In order to assess the basic process behavior as well as the process limits, it was necessary to conduct an experimental study.

The purpose of our experiments was to study the drawing ability and forming quality of friction stir welded Al 7075-T6 and Al 2024-T3. The drawing ability test is designed to find the forming limit and operating range of process parameters without any type of failure. In forming quality tests, the effects of the process parameters on shape distortion and thickness distribution of the product are examined.

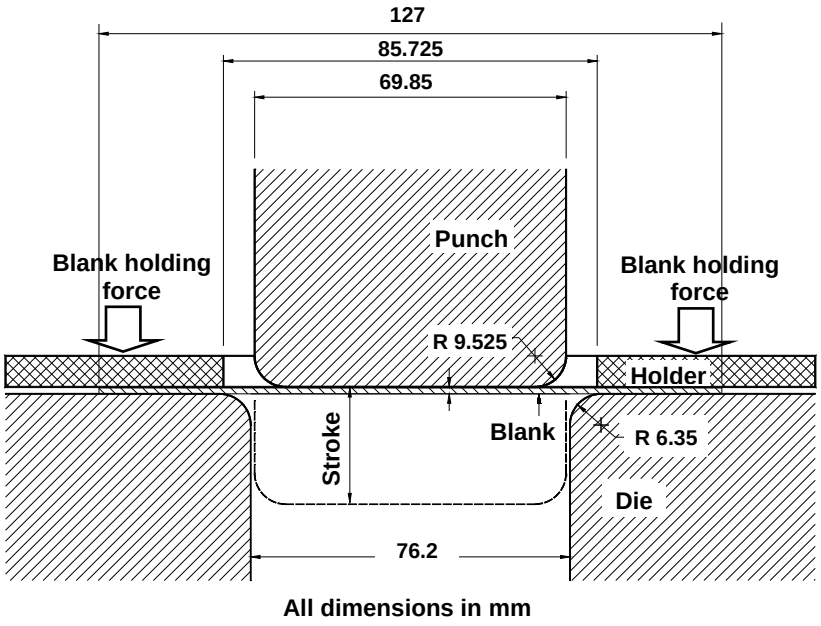


Figure 3.12 The Dimension and Set Up for the Square Pan Drawing Problem.

3.3.1 Experimental Setup and Procedure

MTS servo-hydraulic universal testing machine (Figure 3.13) was used to conduct the forming experiments. Figure 3.14 shows the tooling setup with the eight-point adjustable blank holder on the MTS universal testing machine. A close look of the assembled die set and punch can also be seen in Figure 3.14. Figure 3.15 shows an exploded view of the components composing the forming tool. The square hole in the center of the spring plate is designed to have a tight tolerance with the punch, the purpose of which is to guide the die set and prevent any rotation during the drawing process. The spring plate also plays an important role in exerting and adjusting the blank holding pressure. By tightening the bolts against the die, the eight die springs between the spring plate and holder were compressed, which generates reaction forces on the holder and becomes the blank holding force. The blank holder force was controlled by adjusting the

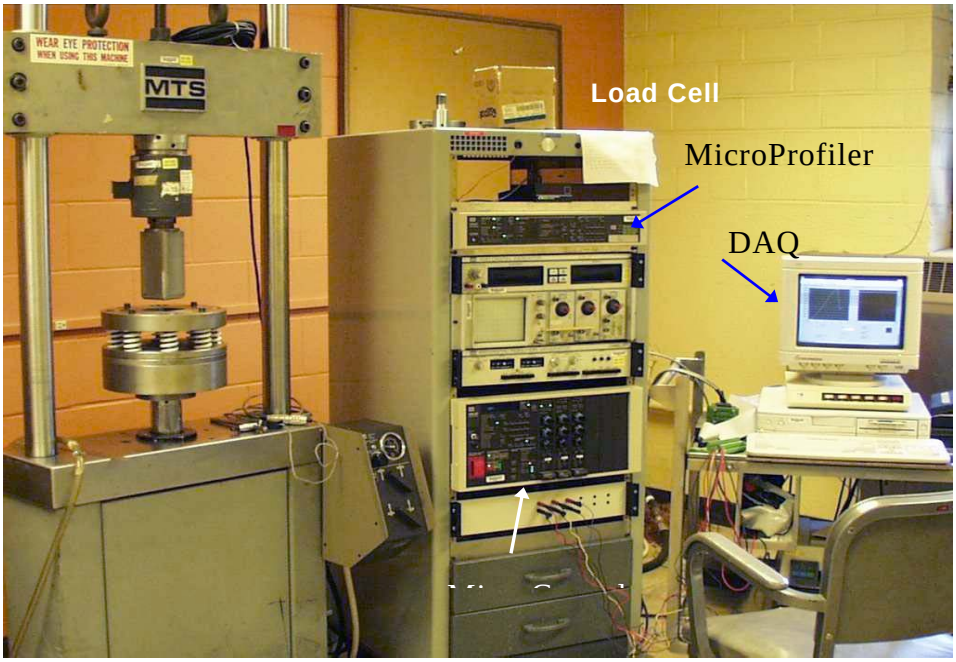


Figure 3.13 Testing Facility with MTS Machine and Data Acquisition System

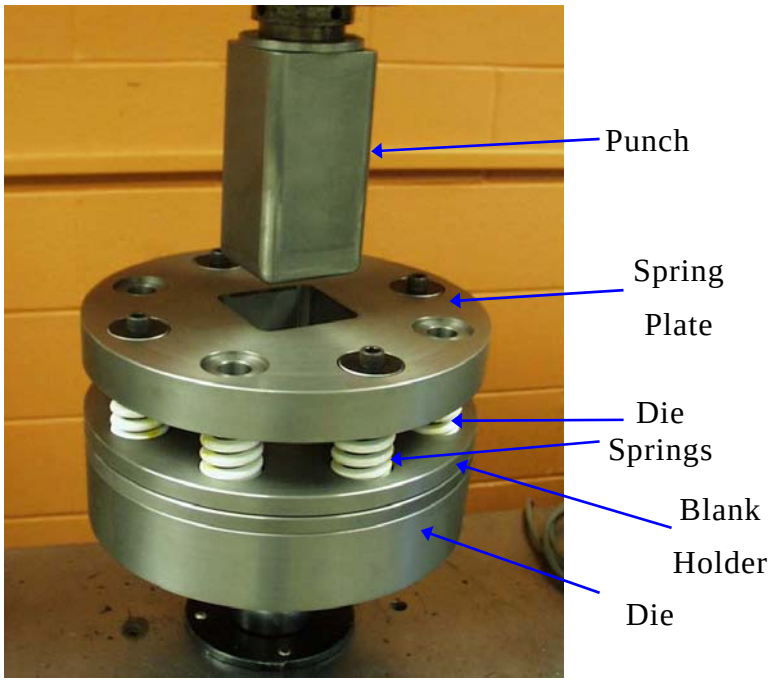


Figure 3.14 Closer Look at the Forming Tool.

height of the die springs, and remained constant throughout the drawing process. Gage blocks were applied to ensure a uniform height of all eight springs. With eight die springs, the holder was capable of carrying out a total blank holder force of 40kN with the springs in a fully compressed condition. The programmable MicroProfiler controls the hydraulic force and movement of the actuator which the die set sits on. The complete testing facility with data acquisition system is shown in Figure 3.13. All tests were conducted under room temperature, and no heating equipment was used.

The experimental program included procedures for workpiece preparation, adjustment of parameters, the DAQ (Data Acquisition) system, execution of the drawing test, and post-process examination. The following discussion reviews the experimental plan and procedures for this research.

Workpiece preparation

Sheets from friction stir welded blanks of Al 7075-T6 and Al 2024-T3 were used for deep drawing. Each workpiece was sheared to a 127mm by 127mm (5in×5in) size from 431.8mm by 203.2mm (12in×8in) blanks.

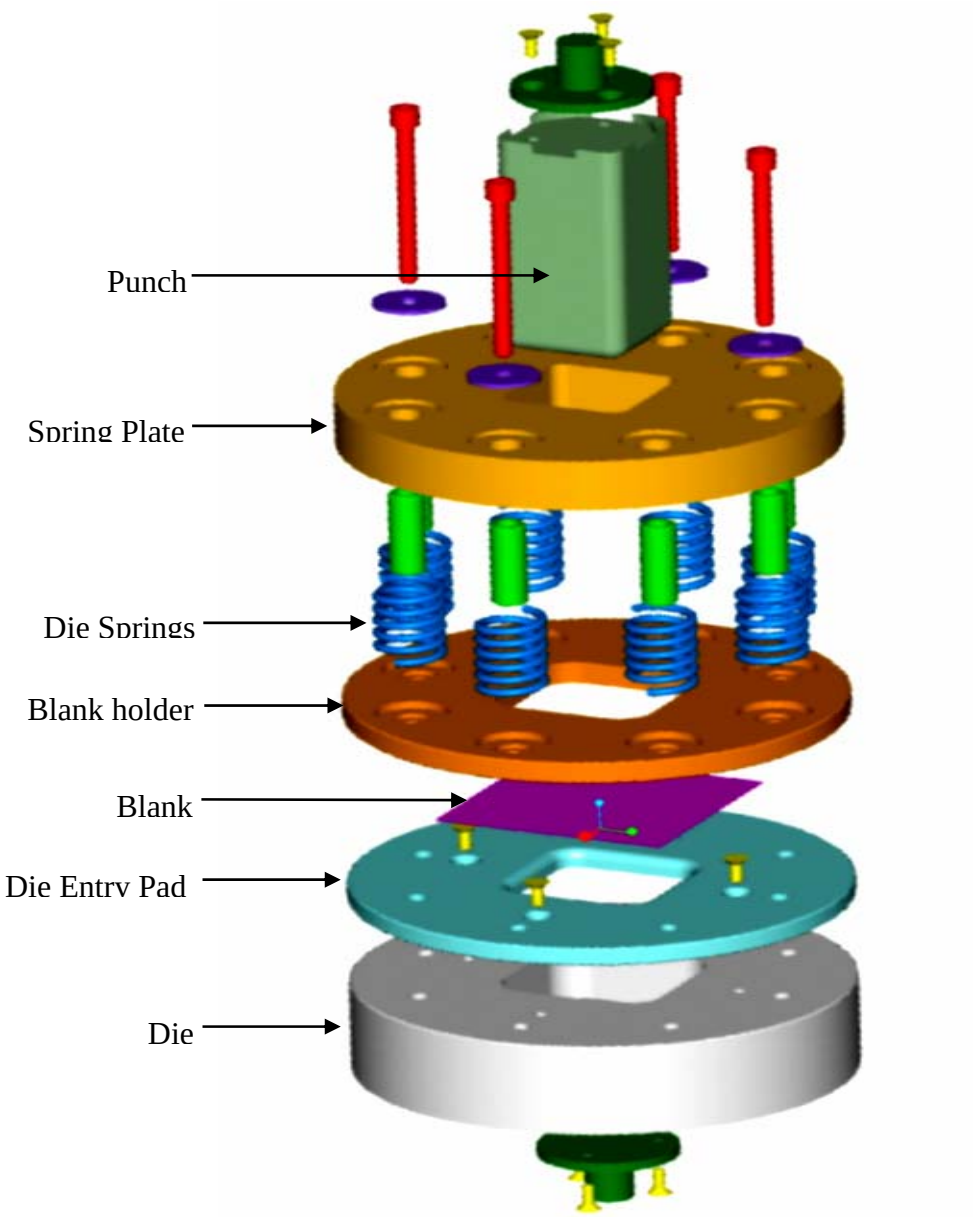


Figure 3.15 Exploded View of the Forming Tool.

Adjustment of the parameters

In this step, the prepared workpiece was mounted onto the forming die, and the blank holding pressure was set. When the workpiece was placed on the die, its alignment with the center and each side of the square die cavity was checked. The blank holder and spring plate were then carefully aligned and placed upon the workpiece. Bolts were gradually and evenly tightened against the die to compress the die springs to achieve the expected blank holding pressure. Gage blocks were used to ensure the uniform amount of compression for all springs. The magnitude and the profile of the punching speed were specified by programming on the MicroProfiler.

Setup DAQ system

Before the execution of the process, the DAQ system was checked. By using the processing software LabVIEW, the signals from the MicroConsole were processed while recording. In the mean time, the selected data (load, displacement, and voltage) were plotted on the monitor to help the user evaluate the whole process, and identify the potential problems. The data file names and sample rate were properly set before each test was conducted.

Execution of the drawing process

Square pan drawing process is a very good representation of several manufacturing process since it involves stretching, bending and drawing processes in more than one axis. Several deep drawing experiments were conducted to determine the optimal conditions among the different cases in Table 3.7. Deep drawing experiments

were conducted with the final objective of improved ductility and formability and reduced process-induced damages. Each forming process began at the position where the punch touched the upper surface of the blank. During the process, the whole die set (including the blank and the holder set) moved up towards the stationary punch. The process was terminated once the fracture occurred. The force on the punch was sensed by the load cell to which the punch was attached. The acquired data (including the load, displacement, and time) were recorded by the data acquisition system throughout the process at 0.005 second intervals. The whole die set was returned to “zero” (starting position) before being cleared from the punch.

The spring plate plays an important role in exerting and adjusting the blank holding pressure. By tightening the bolts against the die, the springs between the plate and holder are compressed, which generates reaction forces on the holder thus becoming the blank holding force. The actual blank holding force can be controlled by adjusting the height of the springs. The programmable Microprofiler controls the hydraulic force and movement of the actuator on which the die sits on.

In this study, formability of the blanks under different cases is investigated. All the deep drawing processes were conducted for all 9 cases and optimal cases for Al 2024-T3 and Al 7075-T6 with a constant velocity of 10mm/min and blank holder force of 3.75KN. Gage blocks were used to ensure a constant height thereby a uniform blank holding force on the welded blanks. Deep drawing processes were carried out until the fracture of the blanks.

3.4 Post-Forming Evaluation

The post-process examination included the observations of wrinkle, sheet thinning, fracture, or distortion of the final parts; and the measurement of the quality characteristics. The post-forming shape distortion (springback) was observed as soon as the part was released from the holder. In order to have a rigorous evaluation of springback associated with each case, Probing function in Centroid controller using CNC machine was used for scanning each square pan (Figures 3.16 and 3.17). Sheet thicknesses variations across the part were followed by measuring the thickness at different locations. The location of the weld line after the forming processes was measured carefully to determine the amount of weld line movement. The combined information from the above procedures provided a complete practical evaluation of the forming ability and quality of the friction stir welding 7075-T6 and 2024-T3 alloys in this square pan forming process.

Figure 3.16 The Coordinate Measuring Machine (CMM).

Figure 3.17 The Measurement of the Square Pan on the CMM.

CHAPTER FOUR

NUMERICAL ANALYSIS

Finite Element Method was used as an analytical tool in this research. Finite Element methods was used to model, simlate, analyze and optimize Friction Stir Welding and Sheet Metal Forming process. Gnereal purpose ABAQUS [62-64] with both explicit and implicit algorithms was used in the modeling of Friction Stir Welding and Sheet Metal Forming processes and to predict process behavior and the process-induced damages. The square pan drawing was used as the case study in order to verify the simulation. After modeling the sheet metal forming process, Optimization using Genetic Algorithm was integrated with the simulation in order to explore optimal process conditions that reduce springback, excessive sheet thinning and weld line movement and hence manage the process-induced damage to its minimum level.

4.1 Modeling of Friction Stir Welding

Friction stir welding processes are inherently nonlinear because of the large strains, high temperature, and plastic behavior of the material in the welding zone. Complex nature of the interface friction conditions between the material and tooling surfaces adds to the difficulty of modeling the process. The standard “implicit” finite element formulation is a true quasi-static solution. However in applications such as FSW, the standard method would yield very large linear matrix equation which must be solved for each load step [59]. The irregular nature of the interface friction and severe contact

conditions will add to the complexity of the solution and make convergence extremely difficult and time consuming for each load step. The explicit method on the other hand [59] is basically a dynamic solution procedure through the application of explicit time integration to the discrete equation of motion. FSW requires simulation of both thermal and mechanical analysis. ABAQUS provides several choices to perform thermal and mechanical analysis. Un-coupled temperature displacement and Sequential temperature displacement analysis will work for the simulation of FSW. In order to simulate actual condition during FSW, its better to perform thermal and mechanical analysis simultaneously. This is necessary because increase in temperature leads to additional stress and expansion of material that needs to be accounted for. Coupled temperature displacement is best suited for simualation of FSW and is used in this research.

4.1.1 Coupled Temperature-Displacement Analysis

A coupled temperature-displacement procedure is used to solve simultaneously for stress/displacement and the temperature fields. In ABAQUS/Explicit, the heat transfer equations are integrated using an explicit forward-difference time integration rule, and the mechanical solution response is obtained using an explicit central-difference integration rule.

Coupled Temperature-Displacement analysis is used to model friction stir welding using ABAQUS-Explicit [63]. In this analysis, temperature and displacement and mechanical responses are determined simultaneously. Heat generated during friction stir welding is produced by friction between tool's shoulder and probe with sheet material and also due to plastic deformation energy. Heat loss from sheet is due to convection

from exterior surfaces of sheet and conduction from bottom surface of sheet to the backing plate.

Governing Equations for Mechanical Analysis

ABAQUS/Explicit solves for a state of dynamic equilibrium at the start of current time increment *t* [63]:

$$M^{NM}\,\ddot{u}^{N}\Big|_{t} = \left(P^{M} - I^{M}\right)\Big|_{t} \tag{4.1}$$

where M^{NM} is the mass matrix, u^{N} is the acceleration vector, P^{M} is the external force or applied load vector, and I^{M} is the internal force vector (the 'internal force' created by stresses in the elements). In the explicit procedure, a diagonal mass matrix is used for efficiency. Thus the nodal acceleration can be easily obtained:

$$\ddot{u}^{N}\Big|_{t} = \left[M^{NM}\right]^{-1}\cdot\left(P^{M} - I^{M}\right)\Big|_{t} \tag{4.2}$$

The central difference integration rule is used to update the velocities and displacements:

$$\begin{aligned}
\dot{u}^{N}\Big|_{t+\frac{\Delta t}{2}} &= \dot{u}^{N}\Big|_{t-\frac{\Delta t}{2}} + \left(\frac{\Delta t|_{t+\Delta t} + \Delta t|_{t}}{2}\right)\ddot{u}^{N}\Big|_{t} \\
u^{N}\Big|_{t+\Delta t} &= u^{N}\Big|_{t} + \Delta t|_{t+\Delta t}\,\dot{u}^{N}\Big|_{t+\frac{\Delta t}{2}}
\end{aligned} \tag{4.3}$$

No iterations are required in the equation solver to update the accelerations, velocities, and displacements; so it is computationally economical for big model size like the cases addressed in this research. The

$$\Delta t \approx \frac{L_{\min}}{c_{d}}, \tag{4.4}$$

stability of the solution depends on the time increment size, which is approximated as the smallest transit time of a dilatational wave across any of the element in the mesh

In which L_{min} is the smallest element dimension in the mesh and c_d is the dilatational wave speed.

$$c_d = \sqrt{\frac{\lambda + 2\mu}{\rho}} \tag{4.5}$$

Where λ and μ are Lame's constants, and ρ is the material density.

Governing Equations for Thermal Analysis

In ABAQUS/Explicit the heat transfer equations are integrated using the explicit forward-difference time integration rule

$$\theta^N_{(i+1)} = \theta^N_{(i)} + \Delta t_{(i+1)} \dot{\theta}^N_{(i)} \tag{4.6}$$

where θ^N is the temperature at node N and the subscript i refers to the increment number in an explicit dynamic step. The forward-difference integration is explicit in the sense that no equations need to be solved when a lumped $\dot{\theta}^N_{(i)}$ capacitance matrix is used. The current temperatures are obtained using known values of from the previous increment.

4.1.2 Heat Generated During Friction Stir Welding

During Friction Stir Welding, heat is generated from two sources. They are from the plastic deformation of the material and from the friction generated between the tool and sheet blank. In order to simulate friction stir welding process accurately, heat generation by both friction and plastic deformation are modeled rather than artificially adding heat flux to the tool.

Heat generated by Friction

In this research, it is assumed that 90% of the frictional energy dissipated is converted into heat energy and this heat is distributed equally between the tool and blank

$$q_A = f\eta P_{fr} \quad \text{and} \quad q_B = (1 - f)\eta P_{fr} \tag{4.7}$$

Where q^A is the heat flux into the sheet and q^B is the heat flux into the tool.
f is the percent of heat flux that flows into the blank (0.5)
η is the factor of converting mechanical to thermal energy (0.9)
P_{fr} is the frictional energy dissipation and is given by

$$P_{fr} = \tau \cdot \dot{\gamma} \tag{4.8}$$

Where τ is the frictional stress and $\dot{\gamma}$ is the slip rate.

Inelastic Energy dissipation is other source of heat in the system. In this research, it is assumed that 90% of plastic deformation energy is converted into heat and added to the system. Plastic straining giving rise to a heat flux per unit volume is given by

$$r^{pl} = \eta\sigma : \dot{\varepsilon}^{pl} \tag{4.9}$$

Where r^{pl} is the heat flux that is added into the thermal energy balance,
η is the factor for percent of heat converted,
σ is the flow stress of the material,
$\dot{\varepsilon}^{pl}$ is the rate of plastic straining.

4.1.3 Heat Lost During Friction Stir Welding

During the friction stir welding process, heat is lost continually due to conduction and convection.

Heat Loss by Conduction

Conductive heat loss is modeled as the heat lost to the backing plate from the bottom of the blank. Conductive heat loss is given by

$$q = k(\theta_A - \theta_B) \tag{4.10}$$

Where q is the heat flux
K is the conductivity
θ^A and θ^B are the temperatures of bottom of blank and backing plate.

Heat Loss by Convection

Heat loss due to convection is considered from all exterior surfaces of the sheet. Heat flux due to convection is given by:

$$q = -h(\theta - \theta^O) \tag{4.11}$$

Where q is the heat flux
h is the film co-efficient
θ is the temperature of the blank
θ^0 is the sink temperature

The developed model deals with characterization of the responses of the material to the mechanical and thermal loading environment generated by friction stir welding. This model provides guidance for selecting the appropriate process conditions that result in desirable properties of the weld.

4.1.4 Material Constitutive Equation

The Johnson-Cook Strain Rate dependent equation is used as the constitutive equation to model the flow stress behavior of the sheet material.

$$\bar{\sigma} = \left[A + B\left(\bar{\varepsilon}^{pl}\right)^{n} \right]\left[1 + C \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right) \right]\left(1 - \hat{\theta}^{m}\right) \tag{4.12}$$

Where is the effective flow stress, is the effective plastic strain, is the normalizing strain rate, A, B, C, n, m are the material constants **(Table 1)**.

$$\hat{\theta} = \begin{cases} 0 & for \quad \theta \le \theta_{transition} \\ (\theta - \theta_{transition})/(\theta_{melt} - \theta_{transition}) & for \quad \theta_{transition} \le \theta \le \theta_{melt} \\ 1 & for \quad \theta \ge \theta_{melt} \end{cases} \tag{4.13}$$

where θ is the current temperature, θ_{melt} is the melting temperature and $\theta_{transition}$ is the transition temperature below which there is no temperature dependence on flow stress.

Table 4.1. Material Properties of Al 2024-T3 and Al 7075-T6

Material Properties	**Al 7075-T6**	**Al 2024-T3**
Density (Kg/cc)	2810	2780
Modulus of Elasticity (GPA)	71.7	73
Poisson's Ratio	0.33	0.33
Ultimate Tensile Strength (MPa)	572	483
Yield Stress, A (MPa)	503	345
Strain Factor, B (MPa)	350	684
Strain Exponent, n	0.4	0.73
Temperature Exponent, m	1.5	1.7
Strain Rate Factor, C	0.12	0.0083
θ_{melt} (°C)	477	502
$\theta_{transition}$ (°C)	25	25
Thermal Conductivity (W/m-K)	130	121
Specific Heat Capacity (J/Kg °C)	960	875

Table 4.2: Material Properties of Titanium

Density: 4510Kg/cc
Modulus of Elasticity; 117.21GPa
Poisson's Ratio: 0.32
Ultimate Tensile Strength: 1030 MPa
Tensile Yield Strength: 930 MPa
Thermal Conductivity: 21 W/m-k
Specific Heat Capacity: 520 J/Kg °C

4.1.5 Interface Conditions

Interaction between tool and sheet is defined by modified Coulomb friction law with a friction co-efficient of μ. The Coulomb law is modified in the sense that there exists a maximum critical frictional stress, say τ_{max} above which the frictional stress stays constant and is no longer equal to the product of friction co-efficient and contact pressure. This modification is necessary in order to model the plastic shear flow behavior of the material when applied shear stress is near the material's shear failure stress. In this study, τ_{max} is set to equal to τ_u, where τ_u is the related to ultimate strength through the relation $\tau_u = \sigma / \sqrt{3}$

4.1.6 Adaptive Meshing

Three types or surfaces used in finite element analysis are Lagrangian, Eulerian and Sliding. Displacement in normal and tangential directions will follow the material in Lagrangian Surface. In Eulerian surface, the material is allowed to flow through the mesh. The mesh is fixed in all direction for eulerian surface. For sliding surface, mesh will follow the material in normal direction, while in tangential direction the mesh is fixed. Thus in this research, top and bottom surfaces are modeled as sliding surface and other surfaces are modeled as eulerian surface.

In Friction Stir Welding, there is considerable amount of material flow around the tool pin and contact forces. This will lead severe element distortion and ultimately lead to premature termination of the analysis. In order to overcome this mesh distortion during large plastic deformation, Arbitrary Lagrangian-Eulerian (ALE) technique has been used. Adaptive meshing with re-meshing has been employed to reduce large element distortion in modeling friction stir welding. Remeshing improves the quality of the mesh as the analysis sweeps the mesh for every preset increment and the results are remapped to the improved mesh.

4.1.7 Modeling Techniques

Total computation time in this method can be reduced by scaling up velocity and mass. Scaling velocity beyond certain limit might introduce nonrealistic dynamic effects that can result in inaccuracy of the solution. Limiting punch speed in the simulation to less than 1% of the wave speed in the workpiece material would not significantly affect the solution accuracy [59]. In the present case of aluminum Alloys with wave speeds of 5600 m/s, a welding speed up to 50 m/s will not affect the accuracy of the solution.

Two panels each with dimension of 50mmX50mmX2mm is modeled as Aluminum 2024-T3 and 7075-T6 alloy with properties given in Table 4.1. The backing plate is modeled as Titanium. The tool is modeled as rigid tool to reduce the computation time. Total of 19608 nodes 12508 elements are used to model the sheet. C3D8RT has been found as an appropriate element for our analysis since it gives both temperature and displacement results. Node biasing technique has been used to increase mesh concentration near the tool/sheet interface and coarser mesh in other areas. This

technique has been found to greatly reduce the computation time. Area where the tool plunges into the sheet can cause severe element distortion and result in premature stop of the analysis. In order to overcome this concern, an artificial hole about the diameter of the probe is introduced where the tool plunges into the sheet.

Figure 4.1 shows the set up of the finite element model of friction stir welding. Tool is modeled as rigid analytical part. Isometric view of tool and sheet before the start of plunge step is show in Figure 4.2.

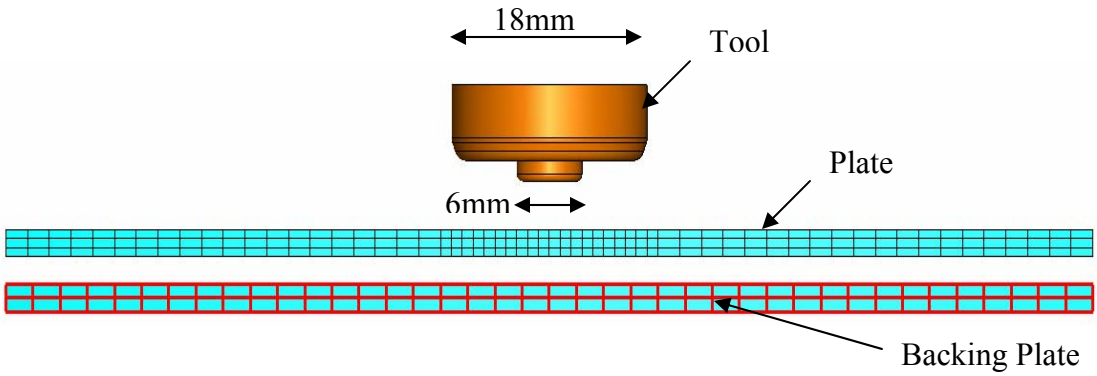


Figure 4.1. Front View of Tool and Plate Set up

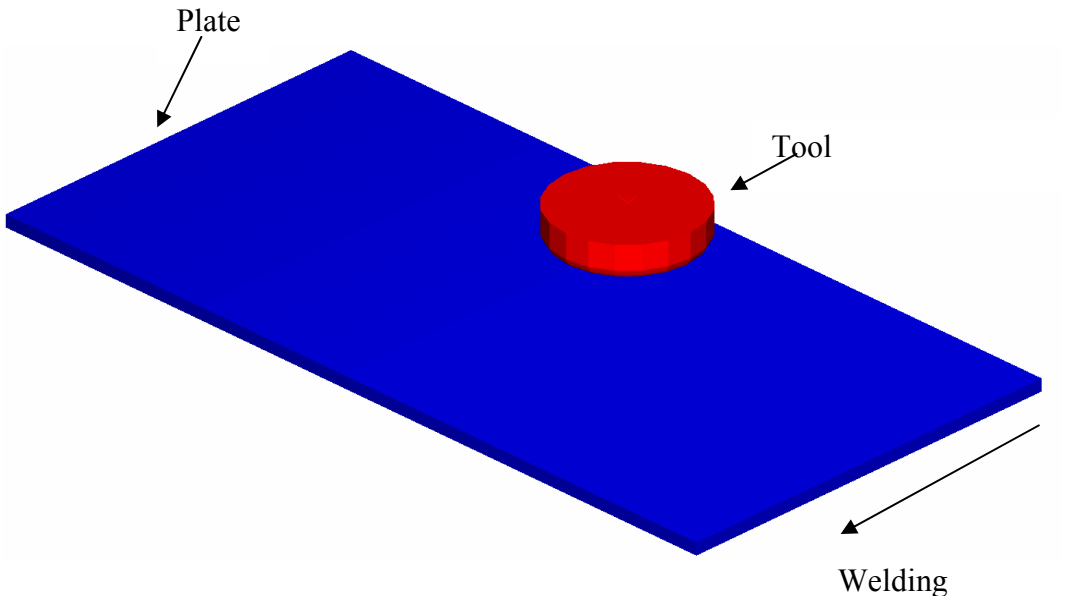


Figure. 4.2. Isometric View of FSW Set up

4.2 Forming of Tailor Preforms

Sheet metal forming processes are inherently nonlinear due to the large strains, plastic behavior of the sheet material, and the complex nature of the interface friction conditions between the sheet and tooling surfaces. The standard "implicit" finite element formulation is a true quasi-static solution. However, in applications such as sheet forming the standard method would yield a very large linear matrix equation, which must be solved for each deformation load step. The irregular nature of the interface between the sheet and tooling surfaces adds to the complexity of the solution and makes convergence extremely difficult and time consuming for each load step. Hence the application of the standard method to simulate large strain deformation in our research was not practical.

The explicit method on the other hand is basically a dynamic solution procedure through the application of explicit time integration to the discrete equation of motion. Total computation time for this method can be reduced by scaling up velocity and mass. The interface friction problem can be simulated in a much simpler way than in the standard method. Furthermore, there is no convergence problem associated with the explicit approach. For prediction of the post deformation shape distortion (springback) in sheet products, the explicit method could suffer from the free vibration motion associated with sheet elastic recovery and it would take long time to vanish even with the velocity and mass scaling techniques. In such situations, the standard (implicit) method is more suitable for predicting the post deformation elastic recovery. Since all deformations are elastic and interface friction does not exist, the convergence time for solution is very reasonable.

Therefore a combination of explicit method for simulating deformation stage and standard (implicit) method for post deformation elastic recovery stage was used in this research. Both methods exist in the commercial FEM code ABAQUS. The forming process is usually simulated using ABAQUS/Explicit [63]. The governing equations are the same as procedures used for Mechanical response of Friction Stir Welding.

The springback analysis is modeled efficiently using ABAQUS/Standard [64], which is based on the nonlinear equilibrium characteristics of the springback phenomenon. The deformed mesh of the blank and its associated state at the end of the forming process from ABAQUS/Explicit are imported into the ABAQUS/Standard at the beginning of the analysis as the initial conditions. At this stage, the model will not be in a static equilibrium condition because of the contribution from two sets of forces: dynamic forces (initial and damping forces) and boundary interaction forces (resulted from the fixed boundary conditions and contact conditions). A set of artificial internal stresses are applied which equilibrate the imported stresses so that static equilibrium was obtained at the beginning of the analysis, then removing gradually while the blank deforms due to redistribution of internal forces, during the springback calculation steps. The implicit procedure uses the full Newton iterative solution method to solve for static equilibrium. For each iteration, an incremental displacement is solved with the standard form equation

$$\Delta u_{(i+1)}^{N} = \Delta u_{(i)}^{N} + \left[K^{NM}\right]^{-1}\left(P^{M} - I^{M}\right) \tag{4.14}$$

where $\Delta u_{(i)}^{N}$ and $\Delta u_{(i+1)}^{N}$ are the incremental displacement for current (i^{th}) and next ($i+1^{\text{th}}$) iteration step of the N^{th} degree of freedom, and K^{NM} is the system's current tangent stiffness matrix. The springback configuration is achieved after complete removal of the

artificial stresses or initial out-of-balance forces. Iterations were repeated in each increment until convergence is achieved. This means that the force and moment equilibrium are attained at every node; and displacement corrections are small compared to incremental displacements.

ABAQUS uses a high-performance, sparse, multi-front equation solver to solve both symmetric and unsymmetrical systems of equations and automatically uses the unsymmetrical solution scheme when the physics of a problem demands it. For Eigen value problems, ABAQUS/Standard uses a Lanczos Eigen solver that is designed for efficiency in large models.

In nonlinear problems the challenge is to provide a convergent solution at minimum cost. This challenge is addressed by automatic control of the time incrementation, which is provided for all relevant analysis procedures. The user defines a "step" (a portion of the analysis history, such as a thermal transient, a stage in a manufacturing process, or a dynamic event). ABAQUS then automatically selects the convergence tolerances and the increments required for the step. This approach is highly effective for nonlinear problems because the model's response may change drastically during an analysis step. Automatic control allows nonlinear problems to be run with confidence without extensive experience with the problem. This capability is a good example of the many features in ABAQUS that make it a production-oriented analysis tool and distinguish it from other finite element programs.

4.2.1 Geometric Modeling

Modeling of Weld

There are two strategies to model tailored blanks in finite element program. The first one is to model the weld accurately. The dimensions, shape and material properties of the weld are taken into account. This requires a fine element mesh in the weld region and a thorough knowledge of the material properties and the heat affected zone. The second one is to neglect the weld type, only taking into account the place of the weld. The second method has been widely used to model welds in welded blanks since it is safe to ignore the weld in deep drawing without high local deformations in the weld area [73]. However to accurately model the tailored welded blanks, dimensions, shape and material properties of the weld zone are modeled in this research.

Modeling of the TWB

As the accuracy of numerical simulation is concerned, 3-D solid finite element model is able to describe exactly the geometry of typical TWBs, namely the middle layers of base metals are not in the same plane after welding. On the other hand, 3-D shell element model is popular in simulation of sheet metal forming due to its cost effectiveness.

The computing time for 3-D solid element model is 15 times more than that of shell elements. Also, 3-D shell elements have advantages of cost effectiveness and accuracy in simulation of TWBs [74]. Zhao et. al. show that shell elements can be used accurately and more cost effective than solid elements. In this research, shell elements are

used to model the tailor welded blanks since shell elements do not sacrifice any accuracy in the results.

Modeling of Tools

Forming of tailored blanks is different from forming of uniform blanks, and hence the tools have to be adapted, notably when base materials are of different thickness.
If the materials of the blanks are different, they deform to a different degree thus causing non-uniform drawing or stretching of the blanks. In order to allow for a uniform forming of the tailor welded blanks, segmented blank holders will be needed with different blank holders applied.

Unlike uniform forming process, flat rigid blank holder cannot be used for tailored blank, since the wrinkling in the thinner base material during stretch forming cannot be prevented and the blank holder force cannot be varied over the product flange. In this case, stepped blank holder is used to reduce the wrinkles and to avoid the un-uniform thickness. A shim is used to offset the change in thickness of the sheet. All the tools used in stretch forming process like die, punch, blank holder and shim are modeled as rigid tool.

In this research, all tools except for the blank are modeled as rigid tools since they undergo negligible deformation compared to the blank and also this technique will help to save the computation time. Segmented blank holders are used since the tailor welded blank is made of two different materials. Higher blank holder force will be applied on the weaker material while lesser blank holder force will be applied on the blank with the stronger material.

4.2.2 Material Constitutive Equations

In sheet metal forming, strains are large and may evolve along complex loading paths. To obtain correct results of the stress, strain, or thickness distribution, the material constitutive behavior must be properly modeled for the workpiece. The Johnson-Cook Strain Rate dependent equation (Eq 4.12) is used as the constitutive equation to model the flow stress behavior of the sheet material. Material properties in Table 4.1 are used to model the blanks used for forming process.

4.2.3 Interface Contact and Friction Modeling

As described above, the blank was considered to be a deformable body, while the punch, holder, and die were rigid surfaces. Therefore the tool/workpiece contact was treated as contact between a deforming body and several arbitrary shape rigid bodies that one or more of them may move during the process being modeled. Based on this characteristic, a pure master-slave algorithm was used on all contact surfaces for the purpose of computational efficiency. The contact surfaces on the punch, die and holder were defined as master surfaces, while the contact surfaces on the blank were slave surfaces. Based on the definition of the algorithm, the slave surface nodes could not penetrate into the master surface; but the master surface nodes were allowed to penetrate into the slave contact boundary surfaces. This was avoided by using finer meshes for the blank surfaces to ensure the computational accuracy.

A classical isotropic Columb friction model, which relates the critical shear stress to the contact pressure and friction coefficient, was used to describe the interface shear stress. Also a kinematic method was used alone with the critical shear stress calculated by

the Columb friction model to impose frictional constraints, and determine the sticking /slipping condition. The force required to maintain a node's position on the opposite surface in the predicted configuration was calculated using the mass associated with the node, the distance the node has slipped, and the time increment. The shear stress at the contact node τ_{eq} and the critical stress τ_{crit} [76] were defined by

$$\tau_{eq} = \sqrt{\tau_1^2 + \tau_2^2} \tag{4.1}$$

$$\tau_{crit} = \mu p \tag{4.2}$$

where μ is the coefficient of friction and p is the contact pressure, τ_1 and τ_2 are shear stresses in two principal directions. If $\tau_{eq} < \tau_{crit}$, the node is considered to be sticking and this force is applied to each surface in opposing directions. If $\tau_{eq} > \tau_{crit}$, the surface is slipping and the force corresponding to τ_{crit} is applied.

4.2.4 Deep Drawing and Stretch Forming Analysis

The finite element meshes for the blank and each component of the deep drawing and stretch forming are shown in Figure 4.3 and 4.4 respectively. The geometry of the blank and forming tool was designed to match the experiment in order to compare results. Since the whole model was symmetrical to y axis, only 1/2 of the blank and forming tool was modeled to facilitate computational efficiency. Symmetrical conditions were imposed through appropriate constraints on boundary locations.

The blank was modeled with four-node, bilinear, quadrilateral shell elements with reduced integration (ABAQUS type S4R). Six degrees of freedom were available at each

node, except for those restricted by the boundary conditions. The stable time increment was determined by the behavior of the membrane (not the membrane element) at the mid-plane of the shell. The small thickness of the shell element had no effect on the stable time increment, which made it very effective for sheet forming simulations. The change in element thickness was calculated by integrating the strain in the thickness direction through the thickness of the shell. The strain in the thickness direction was determined by enforcement of the plane stress condition at the section points through the shell's thickness. The shell was assumed to be homogeneous across the thickness. To calculate the cross-section behavior of a shell, 5 integration points were defined through the

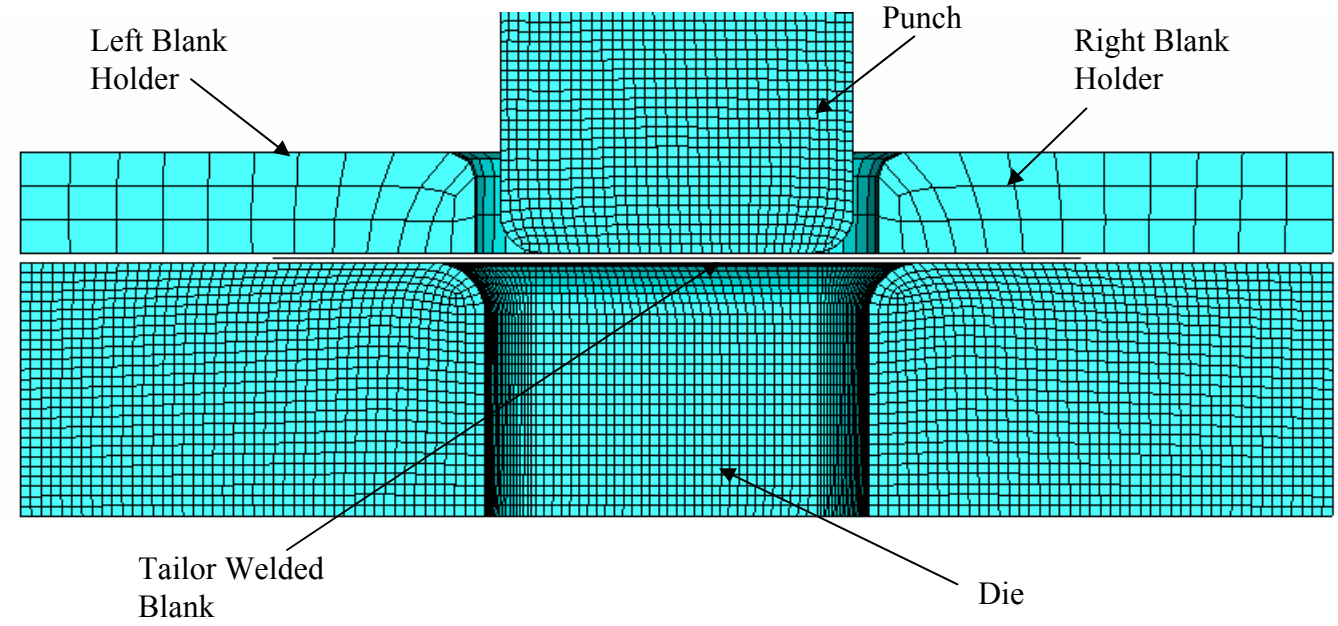


Figure 4.3 Initial Tool Set Up for Deep Drawing Process

thickness. The strain and stress were calculated at these points at one section located at the center of the element. The section points through the thickness of the shell were numbered consecutively, staring with point 1. By using Gauss Quadrature, section point 1

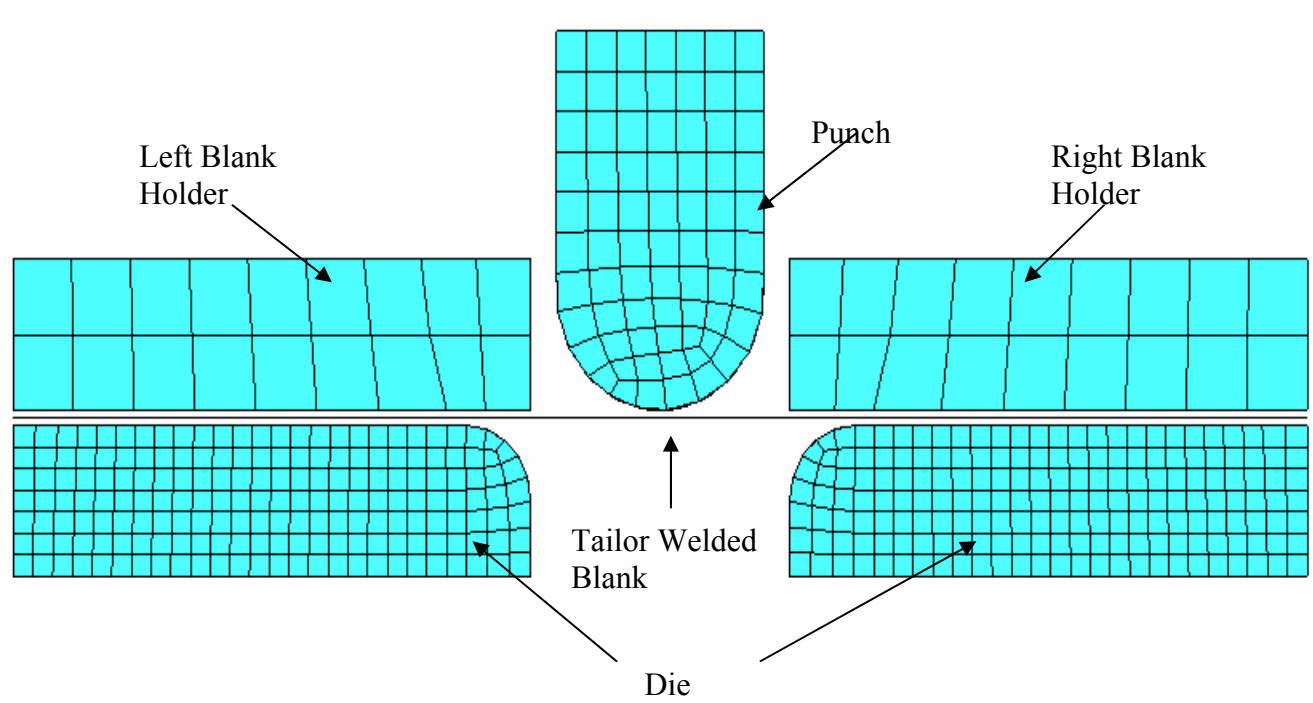


Figure 4.4 Initial Tool Set Up for Stretch Forming Process

was the point closest to the bottom surface. In general, Gauss Quadrature [63] provides greater accuracy than Simpson's rule when the same number of integration point is used. For same level of accuracy, Gauss Quadrature requires fewer integration points than Simpson's rule does, and thus requires less computational time. Reduced (lower-order) integration was used to compute the element internal force vector, while the mass matrix and distributed loads were still integrated exactly. Reduced integration usually provides more accurate results (provided the elements are not distorted) and significantly reduces running time, especially in three dimensions. Based on first-order shear deformable shell

theory, the transverse shear stiffness was computed by matching the response for the case of the shell bending about one axis, using a parabolic variation of transverse shear stress in each layer. Transverse shear strains and transverse shear forces per unit length were calculated at the center (centroid) of the element. The welded blank was modeled using the constitutive equation given in Section 4.1.4 and material properties given in Table 4.1

In modeling the forming tool, four-node three-dimensional rigid elements (ABAQUS type R3D4) were selected to model the punch, die, and holder surfaces. For our research, the forming tool was massive and stiff compared to the blank, which was very unlikely to produce any strain during the process; thus the assumption was made that the tools were perfectly rigid throughout the process. A reference node was specified for each set of elements to define the motion of rigid body. The mass was assigned by attaching a mass element on the reference node of the holder, in order to apply the blank holder force. Only translational degrees of freedom were allowed on the nodes of the rigid elements except the reference nodes, which also have rotational degrees of freedom. During the drawing process, the die and holder element sets were constrained in all directions except the z-direction translational degree of freedom on the holder elements, which was free to accommodate the thickening on the flange area. The downward motion of the punch was defined by its reference node, which also detected the reaction force on the punch during the process.

Process Parameters for Stretch Forming

In the stretch forming model, the tailor welded blanks in case Bi-A 2 has been used. Tailor welded blank is modeled with a total 10302 nodes and 10000 elements. The

tailor welded blank is made of Al 7075-T6 on the left and Al 2024-T3 on the right and the weld.S4R shell elements are used to model the tailor welded blanks.

Stretch forming process is completed in two steps; during the first step, blank holder is brought down with a displacement to come in contact with the blank. Blank holder force and punch velocity is applied during the second step thereby stretching the blank with the movement of blank. A constant velocity of 10mm/min is applied to the punch until its reaches a depth of 16.5mm with a friction c-efficient of 0.125 while the blank is rigidly clamped at the ends to prevent any slipping and be subjected to pure stretching.

Process Parameters for Deep Drawing

Deep drawing for case Bi-A 2 was simulated using ABAQUS/Explicit followed by the import of the deformed mesh into ABAQUS/Standard to simulate the springback process. Tailor welded blank was modeled with a total of 11550 nodes and 10800 elements. The process parameters for the deep drawing process are given below.

Punch Velocity: 10 mm/min

Blank Holder Force: 3.75KN

Co-efficient of Friction: 0.125

Punch Travel: 12.5 mm

CHAPTER FIVE

OPTIMAL PROCESS DESIGN

Optimization is the process of determining the best possible decision for a set of circumstances without having to enumerate all the possibilities. [29]. Two major types of optimization methods are gradient-based optimization method and non-gradient optimization method. The gradient-based optimization methods like steepest descent, simplex method, and two phase method are efficient only for simple problems. However in case of real-time engineering, problems are highly non-linear and the gradient-based methods are not robust to handle them. Hence the non-gradient non-linear methods are under extensive research recently. The non-gradient optimization techniques include simulated annealing (SA) and genetic algorithm (GA). In the past decade, the study of how to apply genetic algorithms to problems in the industrial engineering world has been a subject engaging the curiosity of many researchers. The major reason for this interest is that genetic algorithms are powerful and broadly applicable stochastic search and optimization techniques that really work for many problems subject to complex constraints. [23]

5.1.1 GA vs. Conventional Optimization

In the field of optimization, Genetic Algorithm has been a considerably new development and has been used more than any other optimization technique. The Genetic Algorithms are stochastic global search methods that mimic the metaphor of natural biological evolution [30]. Genetic algorithm is based on the principles of

Darwinian theory of survival of fittest. Genetic algorithms are different from traditional optimization and searching procedures in four aspects:

1. GAs work with a coding of the parameter set, not the parameters themselves.

   During the search of the parameter set which could lead to the optimal result, traditional methods adjust the values of the parameters gradually using certain rules, such as the gradient information of the objective function. In GA, a chromosome string represents a set of parameters. For example, a set of two design parameters x_1 and x_2 for the objective function $f(x_1, x_2, x_3)$, with values x_1 = 7.783 , x_2 = 5.673 and x_3 = 0.057, can be represented by the following binary string:

   00001110011111110010000000001100

   with first 13 bits as x_1 , 12 bits as x_2 and 8 bits for x_3. The total length of the string is controlled by the number of variables, domain (upper and lower limits), and precision requirement (decimal points) of each variable. By randomly changing any one or more digits in the string, the values of the parameters can be easily changed once the binary string is decoded; thus a new search point is located. Detail coding and decoding procedure is shown in Appendix D.

2. GAs search from a population of points, not a single point.

   In searching for the optimum, traditional methods always try one point at a time. In GAs, a “group” of search points are created in each “generation,” which covers a larger area in design space, and leads to a better probability of finding the optimum design more efficiently.

3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.

   In searching for the global optimum, traditional methods use the gradient information of the objective function value in deciding which way to go, and how far it should go, in locating the next search point. For a single-peak function, this method is easy to use. However, for a multi-peak function, this method could "trap" the search point into the local optimum (see Figure 3.4 below). The GAs use random selection in creating new groups of search points, which greatly decreases the chance of the search point being trapped into the local maximum or minimum. For complicated non-linear functions, this is particularly important.

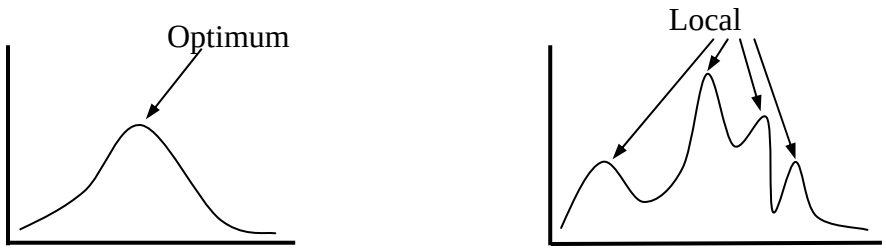


Figure 5.1 The Single-Peak and Multi-Peak Functions.

4. GAs use probabilistic transition rules, not deterministic rules.

   In transitioning from one generation to the next, the GAs use probability in choosing the population in the current generation to create the next one. Those populations with better performance (objective function value) will have higher probability to be selected, a process which resembles the "survival the fittest" concept in nature. This scheme depends totally on the performance of each

generation, not by any one certain rule. It is also the key to how performance improves by itself generation after generation

5.2 Working of Genetic Algorithm

The major part of GAs process contains four steps: encoding, decoding, evaluation, and regeneration [20~22]. In **encoding**, the number of variables, domain of each variable, constraints, and decimal places needed for each variable are used in conjunction with information from the last generation to create a group of finite-length chromosome strings. These chromosome strings with 0 and 1 are then **decoded** to be several sets of parameters with numerical values. Because of the information used in encoding, these sets of parameters should meet all the requirements and constraints. Each set of parameters is then put into an objective function in the **evaluation** step to get the fitness value. The best, total, and average fitness in the generation also can be determined in this step. In the **regeneration** step, a “roulette wheel” is constructed for the selection process. The probability that each member is selected in creating next generation is calculated. The new generation is created with two operations: "mutation" and "crossover." In "mutation", one or more bits of the selected individual (binary) string are randomly selected and “flipped over”, i.e., 0 changed to 1, and 1 changed to 0. In "crossover", any two selected strings are cut at a certain point and exchanged with each other. In addition, there is an "elitist" strategy, which directly reproduces the best individual in the current generation in the new generation in order to ensure that the current optimum individual will never be lost during the selection process. This strategy can also prevent the process from diverging into worse results even there is no

improvement in the best result from one generation to the other. Once the new generation of chromosome strings is created, the decoding process turns these binary strings into sets of values, and proceeds to the step of evaluation.

Generally, the results (total and best fitness values) will gradually improve in the long run (although worse results may appear occasionally in some generations). There is no certain number of generations to get the optimal results. A stopping criterion such as design tolerance or number of generations could be set to stop the iteration and claim the "optimum".

5.3 Integration of GA with Finite Element Analysis

In this research, the Genetic Algorithms was integrated with finite element simulation. The objective of the integrated model was to minimize the amount of springback, sheet thinning and weld line movement. Figure 5.2 illustrates the architecture of the integrated model and the flow of information between the finite element analysis using ABAQUS and optimization routine with GAs. The integrated system contained three major parts: optimization, input/output data processing, and FEM analysis. The nonlinear Genetic Algorithms (GAs) were used as an optimization tool, and the nonlinear FEM programs, including both ABAQUS/Explicit (forming) and ABAQUS/Standard (springback), were used as an objective function module. An interface program is developed in linking and integration of the two systems [73-74].

The integrated process starts with the initial input of the optimization. Major steps are:

Step 1: Specify the initial value and the constraints of each design parameter. In addition, the major parameters of GAs - population size, probability of mutation and cross over, number of maximum generation, etc., are specified in this step. Using GAs, the initial design group (“first generation”), with the number of members equal to the population size, is generated and sent to the interface program.

Step 2: The interface program writes input file for each member of the design group and sends it to ABAQUS for FEM analysis. For each case, major output – maximum, average shape deviation, sheet thickness and the objective value - are reported to GAs program and stored. The GAs program finds the best design among the “first generation” and completes the initial run.

Step 3: By GAs operation, the “new generation” is created with information from the previous generation and the parameters controlling GA. A process similar to Step 2 is executed for this new design group and results fed back as new outputs to the main (GAs) program.

Step 4: The best solution in the new generation is found and compared with the best one from the previous generation, and the current optimum is updated if an improvement is made. The process then repeats Steps 3~4 until the stopping criteria (such as design tolerance or number of generation) is met.

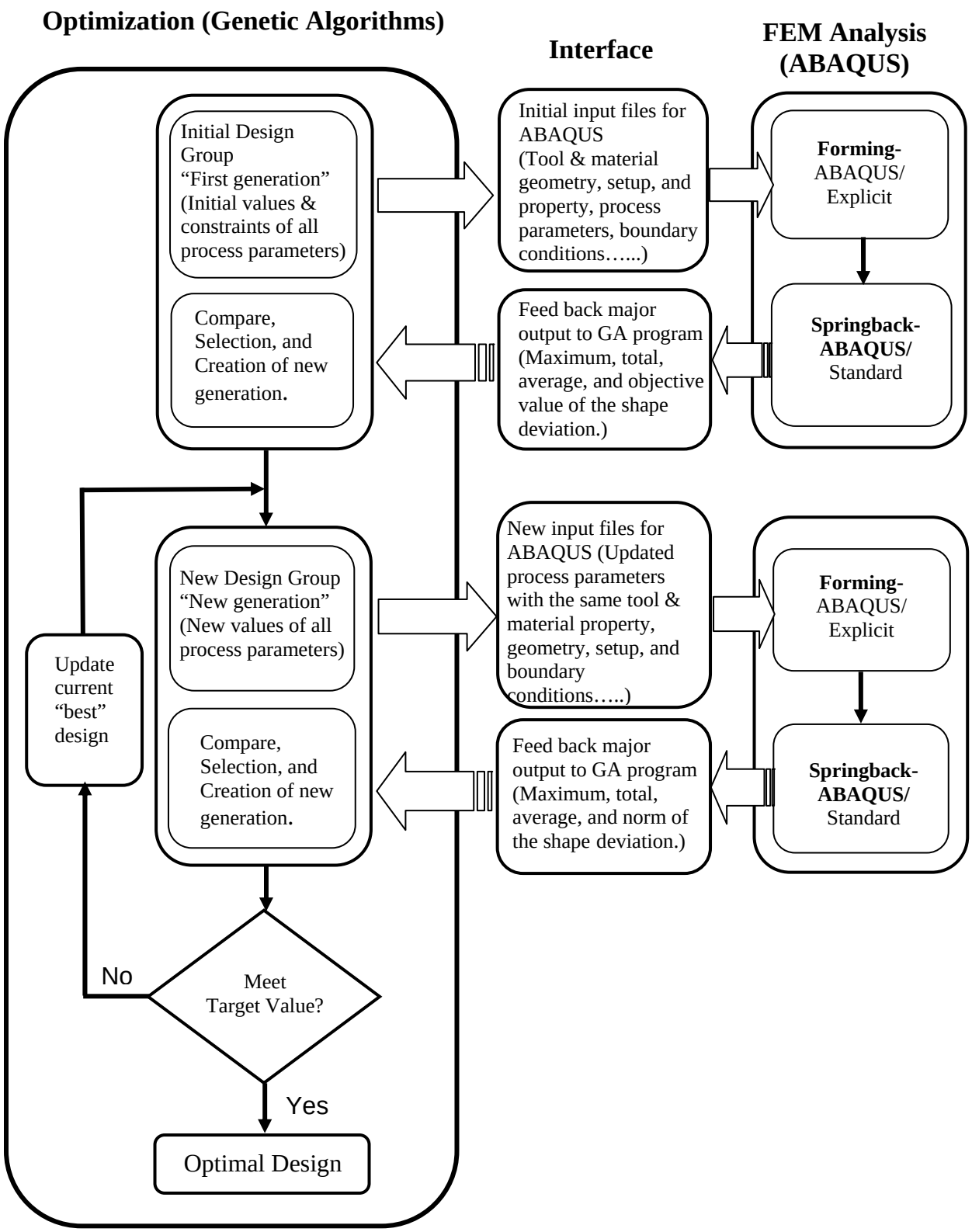


Figure 5.2 Integration of Optimization and Finite Element Analysis.

5.4 Optimization of Sheet Metal Forming

5.4.1 Formulation of Simulation Problem

Finite Element Analysis of deep drawing process acts as the objective function in the integrated model. The square pan model used for earlier study was selected for optimization also. The same material properties, tool geometry, and sheet thickness were used. Constant punch velocity of 10 mm/sec was selected for the optimization process. Two different blank holder forces profile were applied on left and right blank holder. he purpose of optimization here is to define the best configuration of this profile. The effects of the interaction between friction co-efficient and blank holder force on process-induced damages are also analyzed.

5.4.2 Formulation of Process Optimization Problem

One of the main aspects of formulating a problem in Genetic Algorithm is to define the parameters of the GA for the present investigation. Several parameters like maximum number of generations, population number, cross over probability, mutation rate are to be defined prior to running the program.

- Maximum Number of Generations:

The maximum number of generations is the number of times the program has to run. Maximum generation of 200 was found to give reasonable results at acceptable computation time [37]. Twenty generations were selected for the present study because of the small variable range used.

- Population Size:

The population size is the total number of parameter chromosomes in each generation. A population size of 5 was reported to yield good results [37]. Therefore a population size of 5 was set for our algorithm.

- Cross over probability:

Crossover probability determines the recombination of chromosomes among the existing parameters. For single point crossover, 0.6 or 0.7 is best and for uniform crossover, 0.5 is a better choice [37].

- Mutation rate:

Mutation determines the probability of mutation that will occur. Mutation is employed to give new information to the population and also to prevent the population from being saturated with the same chromosomes. Large mutation increases the probability that healthy individuals might be eliminated. In this research, the cross over and mutation probability was set at 0.5 and 0.02 respectively.

The optimization programs used in this research are show in Appendix C.

The main purpose of this research is to reduce the process-induced damage, which constitutes sheet thinning, shape deviation and weld line movement. For a general three-dimensional shape such as the current square pan model, a new strategy to evaluate the process-induced damages (which includes different categories of the shape deviation, sheet thinning and weld line movement) quantitatively. The objective function used in our case involves the weighted-sum, multi-objective type [23]. The objective function used here in was defined as:

$$Obj = w_1 O_1 + w_2 O_2 + w_3 O_3 \tag{4.3}$$

$$\Sigma w_i = 1,\ w_i /0 \tag{4.4}$$

where w_1, w_2, w_3 are weight factors for the three addressed objective functions. O_1, O_2 and O_3 are the objective functions corresponding to the three main process-induced damages studied in this research, Springback, Sheet Thinning and Weld Line Movement respectively. Thus, the Objective function in this study is

$$\text{Obj, } O=0.33*O_1+0.33*O_2+0.33*O_3 \tag{4.5}$$

Springback Damage, $O_1=0.6*O_{11}+0.4*O_{12}$, where

$$O_{11} = \text{Max } (ds_i),\ i = 1 \sim M \tag{4.6}$$

where O_{11} is the objective function referring to maximum shape deviation

$$O_{12} = \frac{1}{M}\sum_{i=1}^{M} ds_i^2, \text{where} \tag{4.7}$$

$$ds_i = \sqrt{dx_i^2 + dy_i^2 + dz_i^2} \tag{4.8}$$

where O_{12} is the objective function referring to average shape deviation

Sheet Thinning, $O_3 = (\text{Max}(sth_{el\_i}) - \text{Min } (sth_{el\_i}),)$ (4.9)

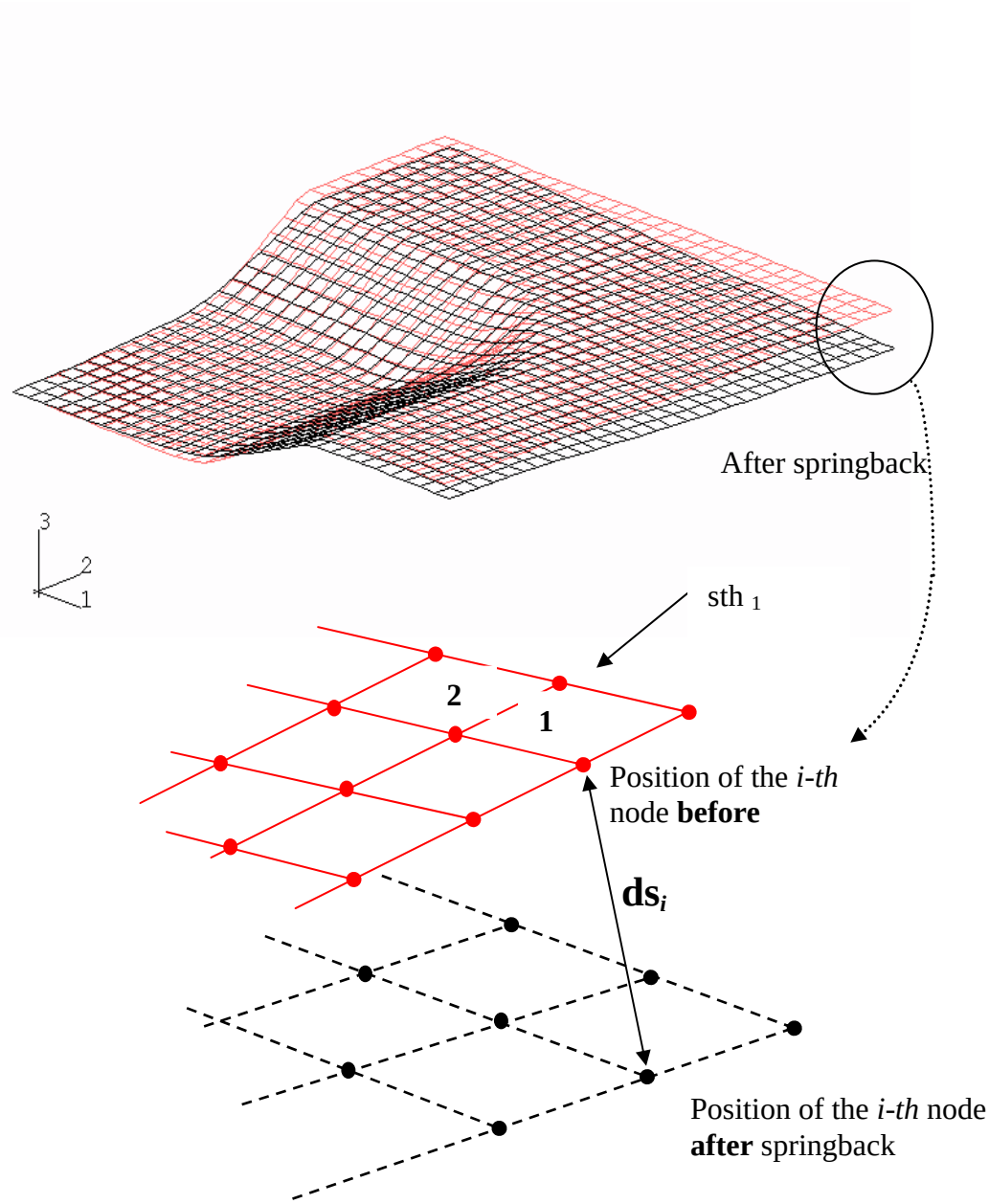
Weld Line Movement, $O_3 = 0.6*O_{31}+0.4*0_{32}$, where

$$O_{31} = \text{Max } (dw_i),\ i = 1 \sim M \tag{4.10}$$

where O_{31} is the objective function referring to maximum weld line movement

$$O_{32} = \frac{1}{M}\sum_{i=1}^{M} dw_i \tag{4.11}$$

where O_{32} is the objective function referring to average weld line movement



ds_i : position deviation of the *i-th* node

dx_i : x-component of the position deviation of the *i-th* node

dy_i : y-component of the position deviation of the *i-th* node

dz_i : z-component of the position deviation of the *i-th*

Figure 5.3 Evaluation of Shape Deviation, Sheet Thinning and Weld Line Movement.

ds_i is the position deviation of the ith node on the workpiece model, and dx_i, dy_i, dz_i are the three components of the ds_i; M is the total number of nodes; j, k are the node numbers on the flange tip along the transverse and diagonal directions (Figure 4.15); sth_{el} is sheet thickness of elements, N is the total number of elements. dw_i is the movement of the nodes along the weld line. Since the process-induced damages come directly from the evaluation of the springback analysis results, sheet thinning and weld line movement (which are functions of the process design parameters), the objective function is also related to the design parameters.

CHAPTER SIX

RESULTS AND DISCUSSIONS

This current study includes four different stages. In the first stage, friction stir welding of single material (Al 2024-T3) was performed numerically and verifying the model by conducting the experimental evaluation. In the second step, a systematic study of friction stir welding for bi-alloy material (Al 2024-T3 and Al 7075-T6) was conducted with different process parameters to determine the optimal quality weld and formability. In the third step, stretch forming and deep drawing processes were conducted to determine the formability and ductility of the friction stir welded blanks. Finally in the fourth stage, Genetic Algorithm was used as an optimization tool to determine the optimal process design for minimal process-induced damages.

6.1 Friction Stir Welding

6.1.1. Numerical Results

Two panels each with dimension of 100mmX300mmX2mm cut out of Al 2024-T3 were butt welded together using FSW. Total of 19608 nodes 12508 elements are used to model the sheet. C3D8RT has been found as an appropriate element for our analysis since it gives both temperature and displacement results. Node biasing technique has been used to increase mesh concentration near the tool/sheet interface and coarser mesh in other areas. This technique has been found to greatly reduce the computation time. Area where the tool plunges into the sheet can cause severe element distortion and result

in premature stop of the analysis. In order to overcome this concern, an artificial hole about the diameter of the probe is introduced where the tool plunges into the sheet.

The blank is modeled as Aluminum 2024-T3 alloy with properties given in Table 4.1. The backing plate was modeled as Titanium with properties given in Table 4.2. The tool was modeled as rigid tool to reduce the computation time.

The entire friction welding process was achieved by four different steps: Plunge (15 seconds), Dwell (20 seconds), Weld (50 seconds) and Pull (5 seconds).

Two different case studies were performed for numerical analysis of friction stir welding. Process parameters used in these analyses are given in Table 6.1 All parameters are kept constant for both case studies except for rotational velocity and plunge depth applied on the tool.

Table 6.1. Process Parameters for Case studies

Process Parameters	**Case 1**	**Case 2**
Welding Velocity (mm/sec)	2.43	2.43
Rotational Velocity of Tool (rpm)	1045	445
Co-efficient of Friction	0.3	0.3
Effective Plunge Depth (mm)	2	1.9

Figure 6.1 displays temperature distribution fields along the weld line at the end of welding process. Temperature distribution across the width of entire blank is shown in Figure 6.2. The temperature in area adjacent to the tool is the highest (538 °C). The evolved temperature field is extended to cover the entire blank because of the active heat transfer by convection due to the flow of heated material and conduction to the neighboring zones. The lowest temperature in the field during FSW is 45°C. The close

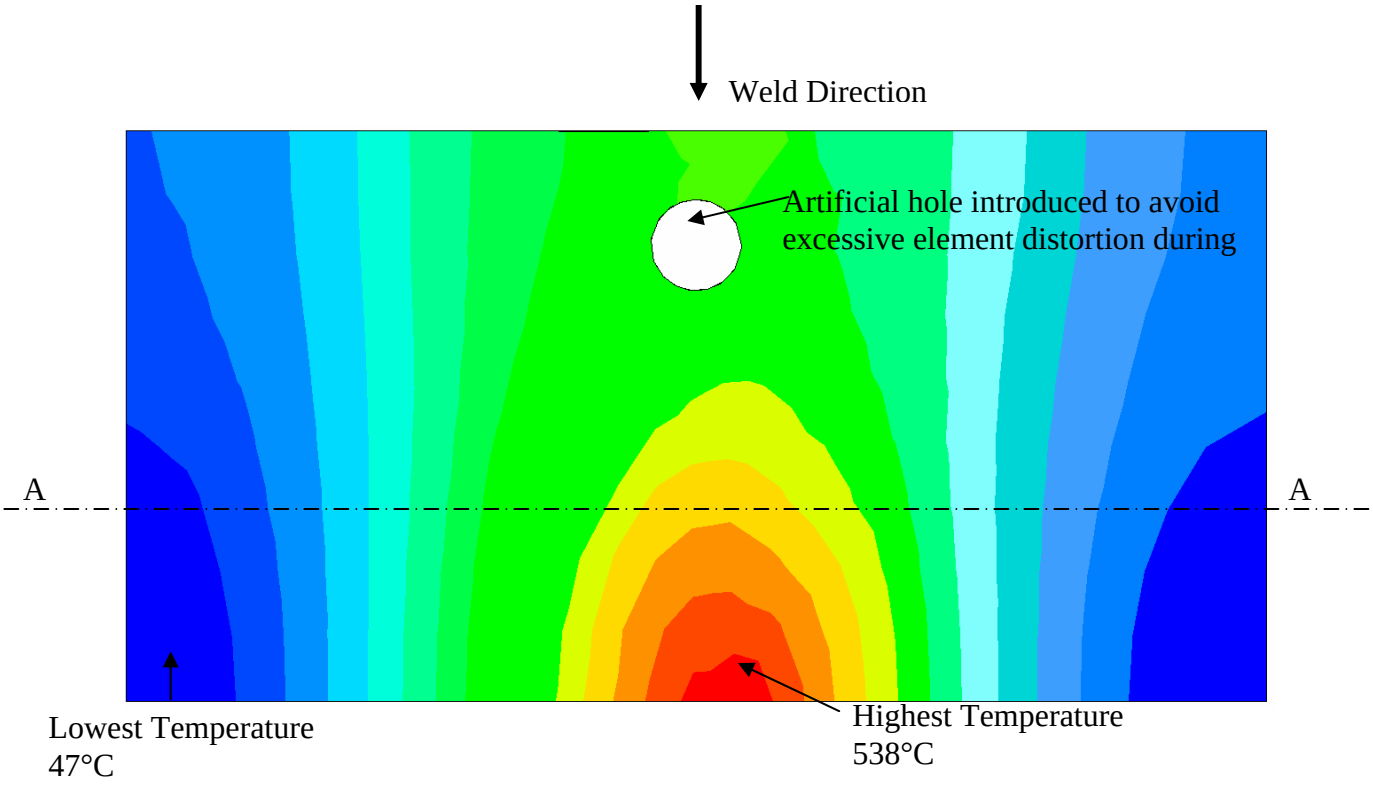


Figure 6.1 Temperature Distribution from Top View

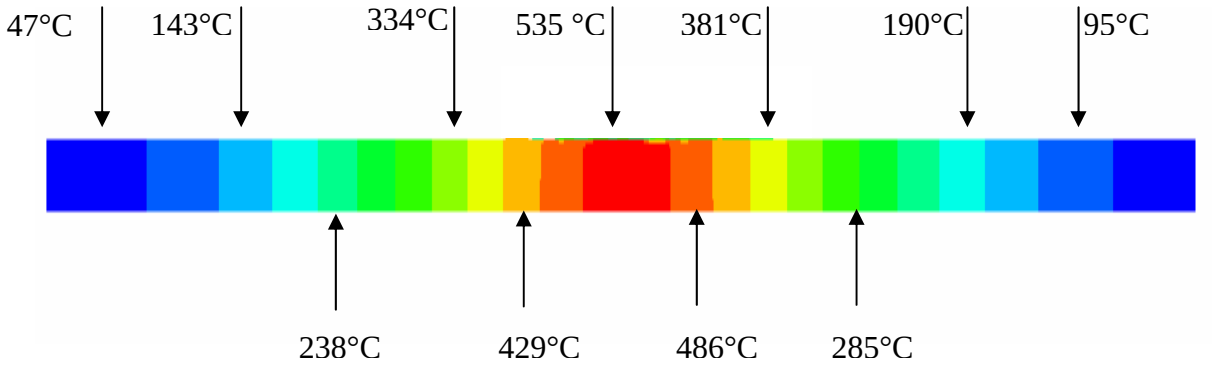


Figure 6.2 Temperature distribution along A-A from Fig 6.1

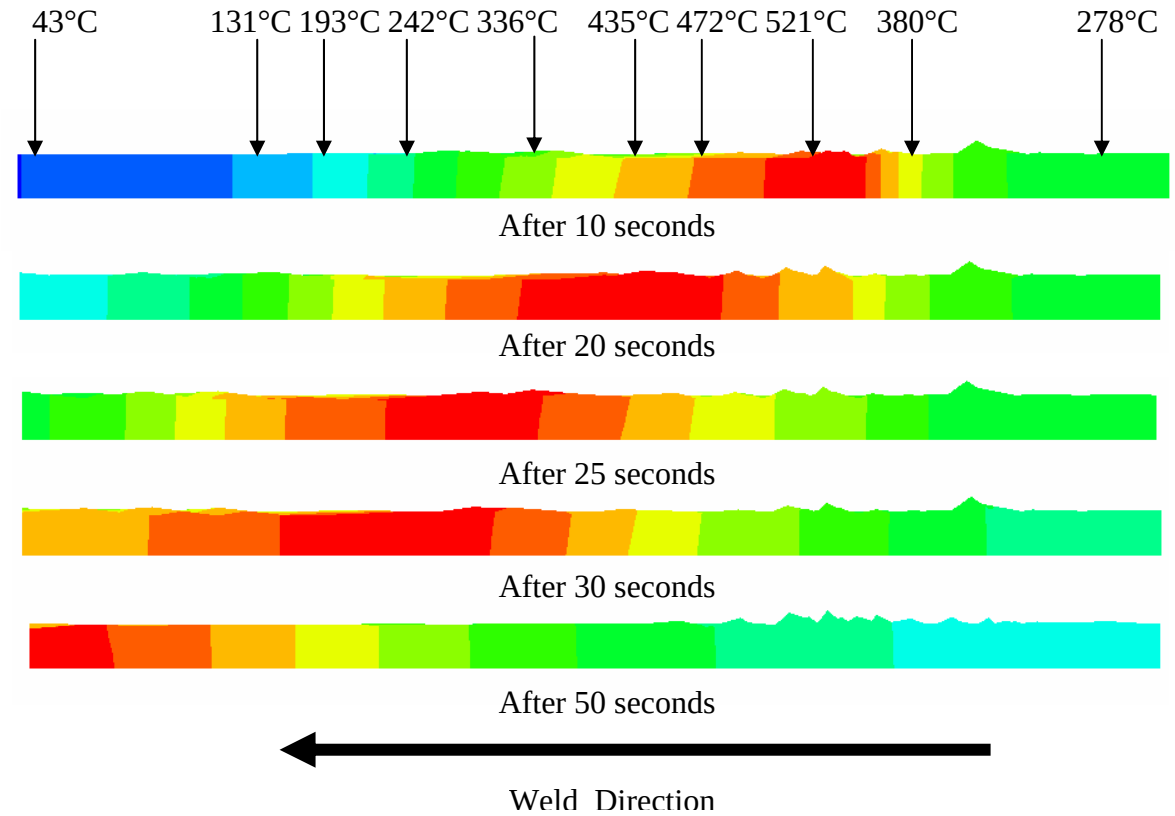


Figure 6.3 Temperature Distribution Along Weld Line as Tool Progress During Welding

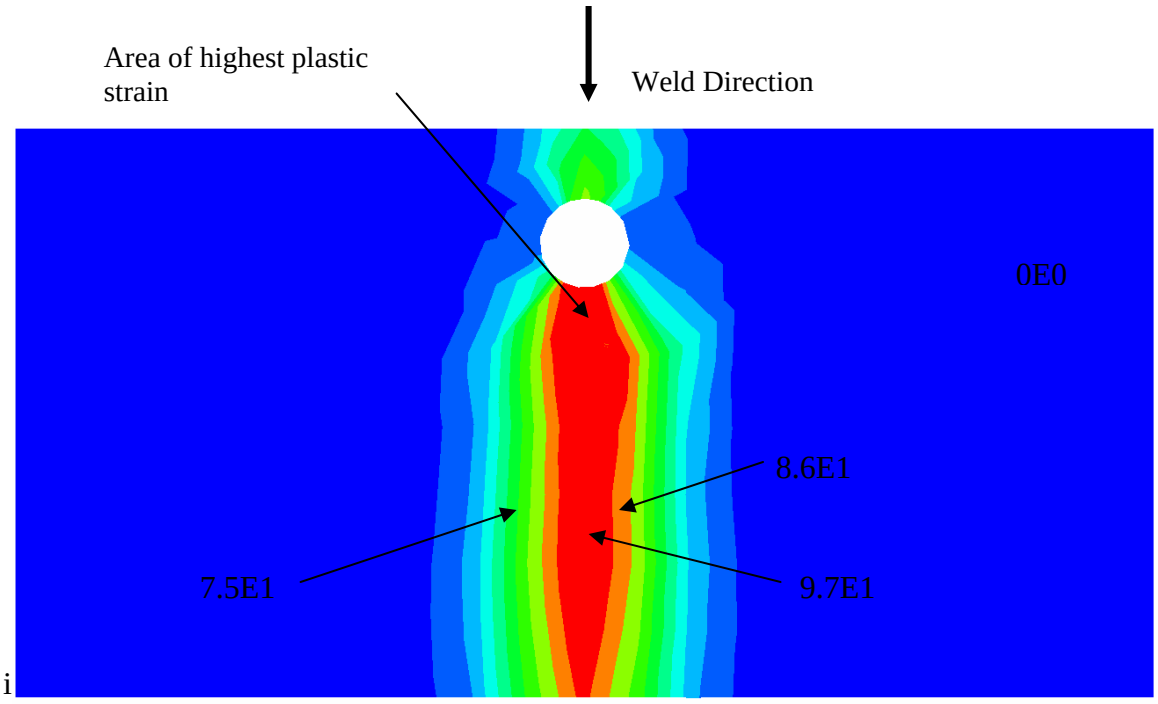


Figure 6.4 Equivalent Plastic Strain

conduction to other zones in the panel. Figure 6.3 shows how the blank is heated as the tool travels along the weld line. It can be seen that once the steady state condition is achieved, heat generated is almost the same.

Equivalent plastic strain along the joint-line is shown in Figure 6.4. Plastic strains represent an integration of the material deformation as result of the stiring action during the process. The high level of plastic strain (highly deformed region) along the weld line is an indication of the the size of the weld nuget and the adjacent thermo-mechancal affected zone (TMAZ).

Blank undergoes high deformation during the plunge step due to high forces experienced by plunging of tool into the blank and also due to material displacement in this location. High stress can be seen in Von-Mises equivalent stress distribution shown in Figure 6.5. The residual stress at end of weld process is given in Figure 6.6. It can be seen that high residual stress still exists at the plunge area even after the welding process.

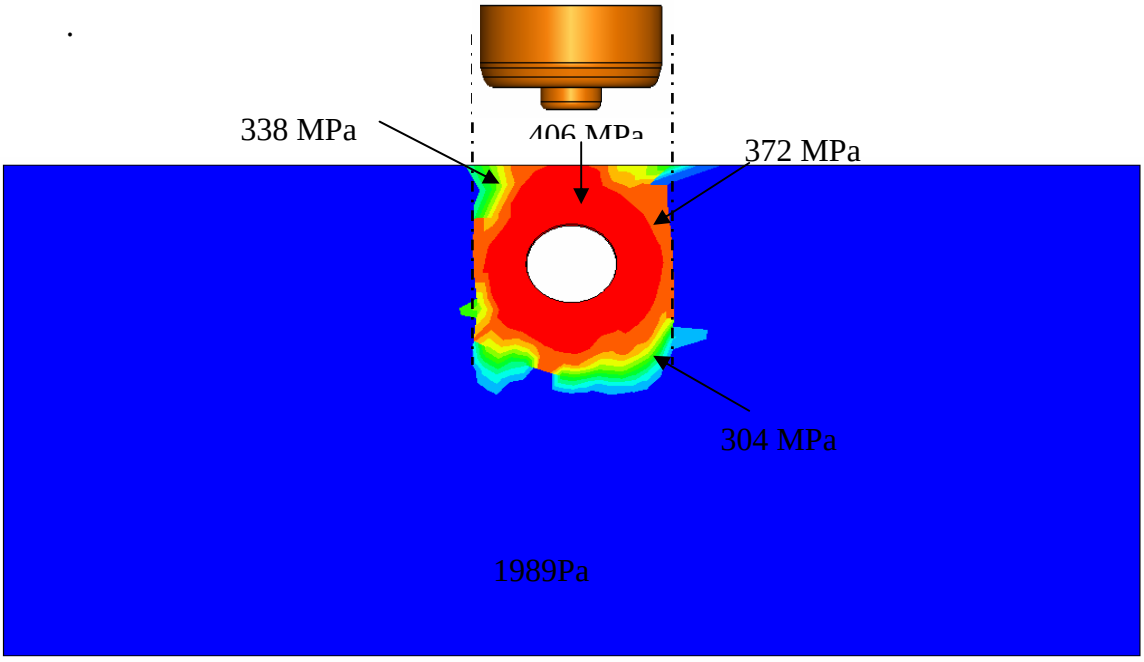


Figure 6.5 Von-Mises Stress at End of Plunge Step

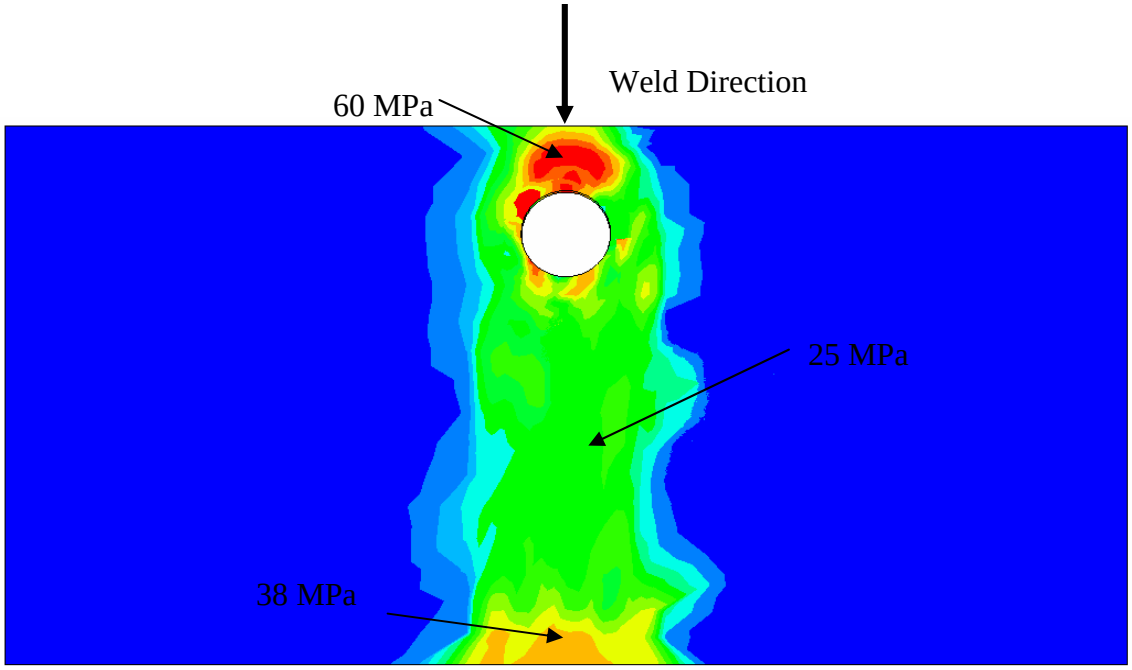


Figure 6.6 Residual Stress distribution at end of Weld

One can conclude that simulation results presented in Figures 6.1 to 6.6 can be used to evaluate alternative process designs in order to explore the process conditions that will lead to sound joint

In the second case study, conditions not favorable for a successful weld are investigated. All process parameters are kept the same except for reduced rotational velocity of tool and plunge depth. Since the plunge depth is not sufficient, shoulder doesn't come in full contact with the material and hence no sufficient heating is achieved. Reduced rotational velocity also means less heat is generated. As a result, sticking friction between the tool and the material is not established and material does not often enough and the whole joining process is stopped. Under these conditions no flow of weld material into the gap along the weld line and a groove is generated as a result of removing instead of depositing material behind the FSW tool. This is evident in figure 6.7. Figure 6.8 displays the temperature fields for this case. Maximum temperature never

exceeds 110°C which is not enough to establish softening by dynamic recrystalization in the material.

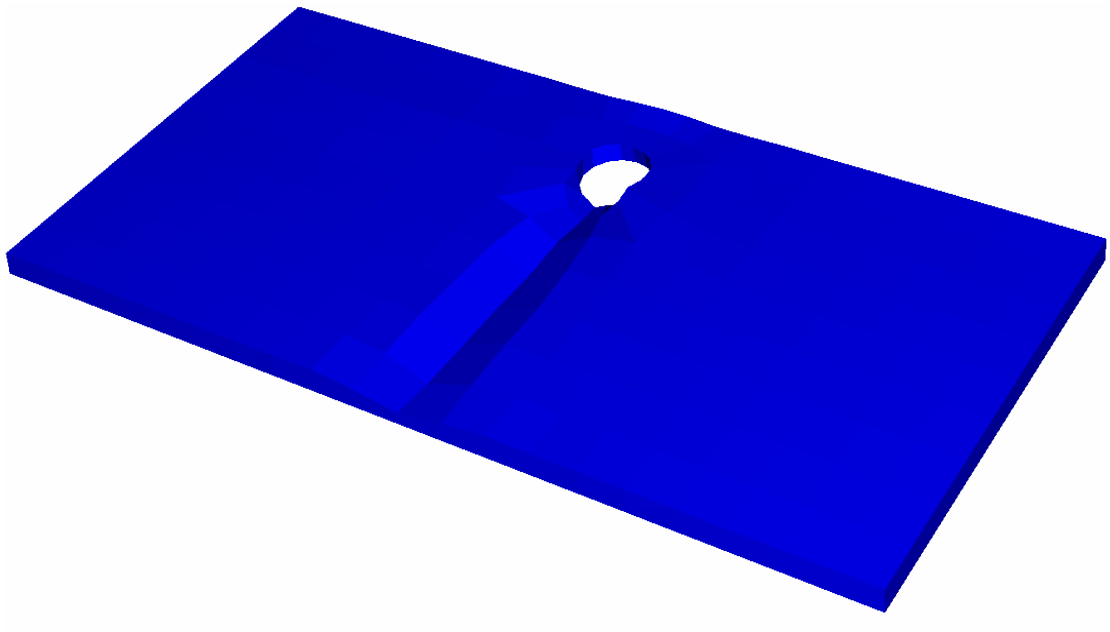


Figure 6.7 Deformed plot showing unsuccessful Weld with no metal deposition

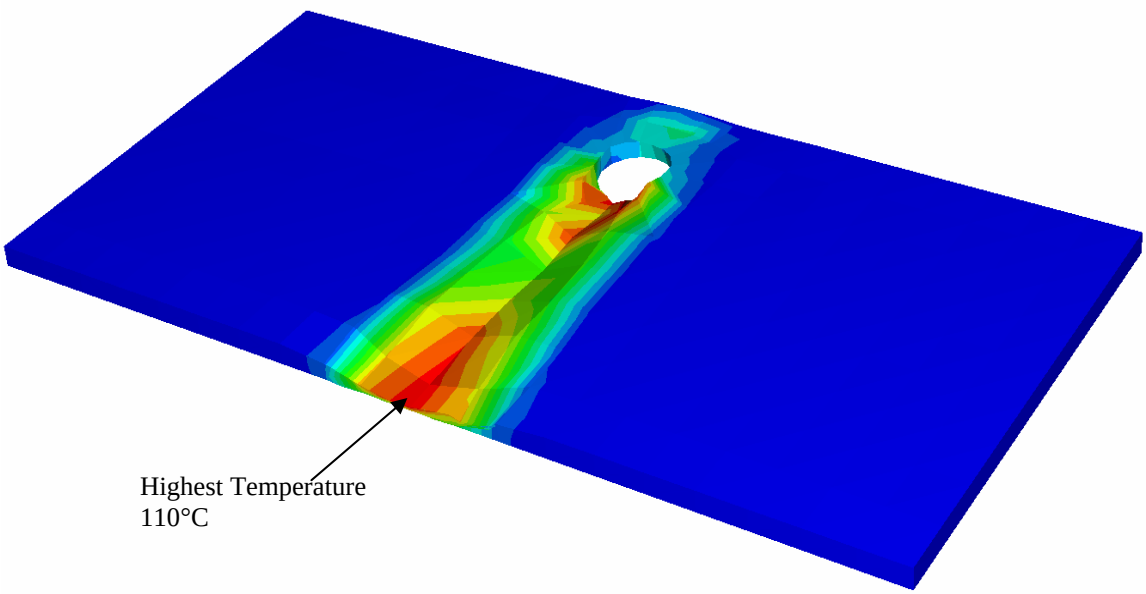


Figure 6.8 Insufficient Temperature Distribution

6.1.2. Experimental Evaluation

Frictional stir welding is performed experimentally using two blanks of Al 2024 T3. Translational and rotational velocity of tool is 1045 rpm and 2.4mm/sec respectively. Depth of plunge is set to thickness of the sheet and dwell period of 15 seconds is used for the welding process. Figure 6.9 shows the friction stir welding process is progress. Successful welds were created with the given process parameters. Figure 6.101 shows the joined blank at the end of friction stir welding process (case study 1). No weld (Figure 6.12) was created when plunge depth (1.9mm) was set to less than thickness of blank (2mm). Temperature indictating paints were used to verify the temperature distribution during the welding process. Each paint has its melting point and when the paint has melted, it shows the corresponding temperature has been achieved. Through this technique, maximum temperature of 529C has been recored which is very close to numerical prediction of 538C.

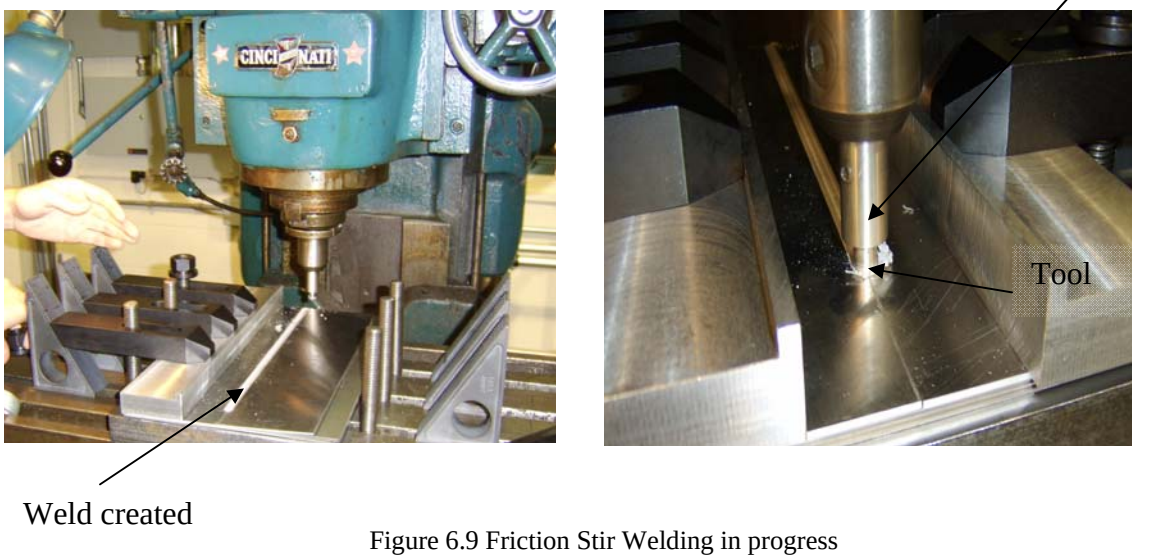


Figure 6.9 Friction Stir Welding in progress

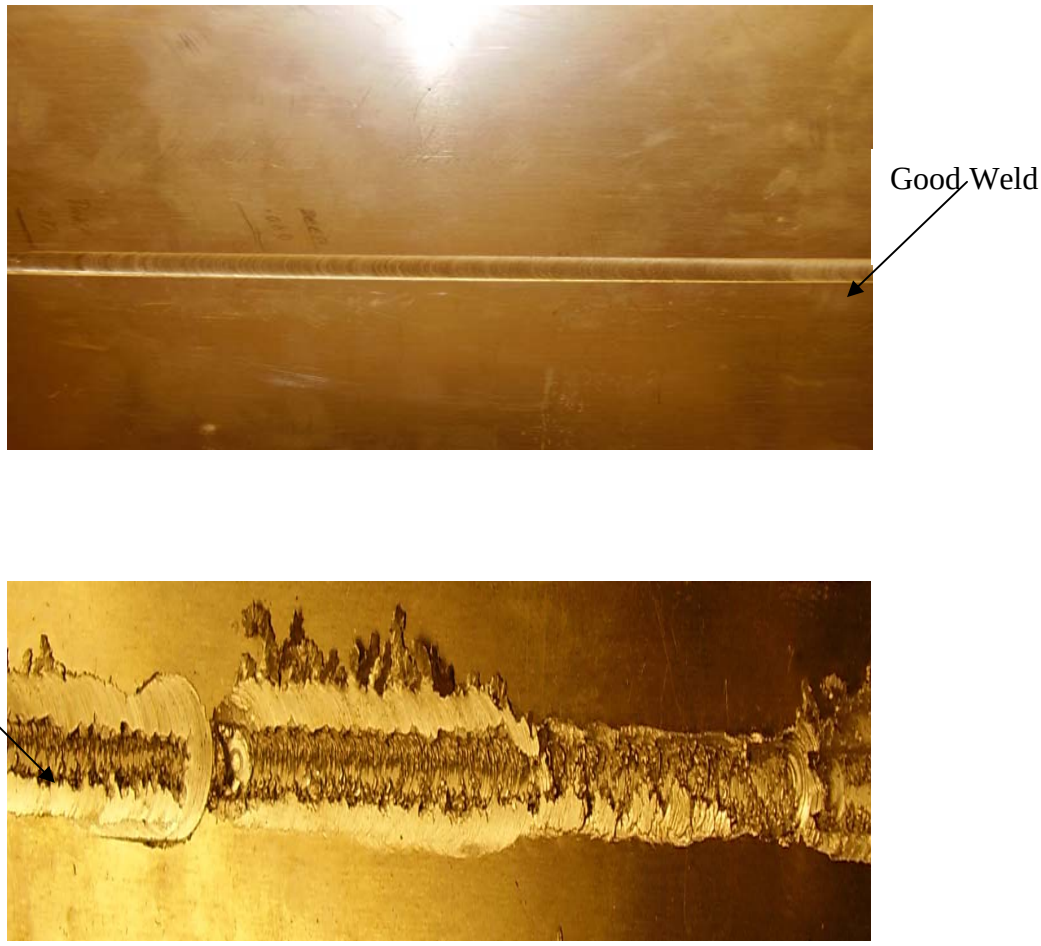


Figure 6.10 Successful and Unsuccessful Weld with no metal deposition

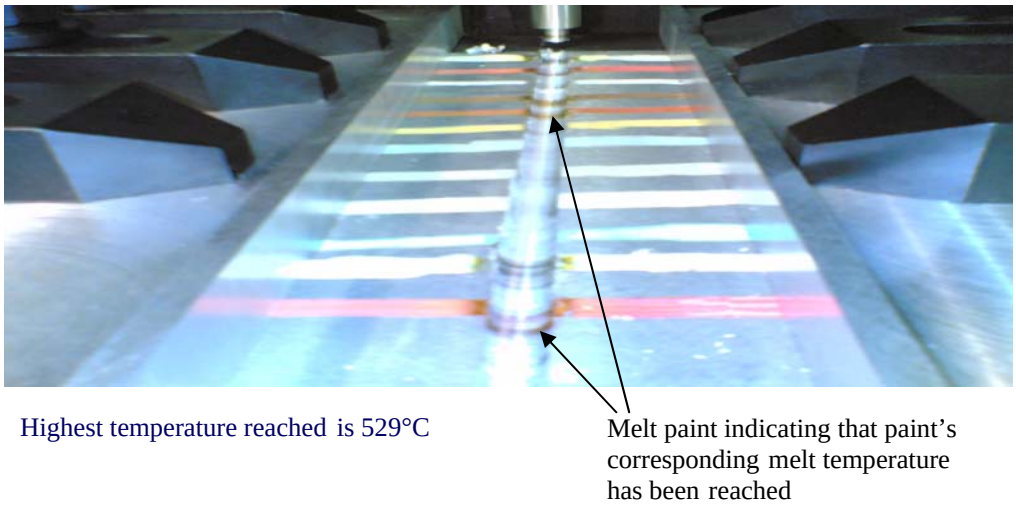


Figure 6.11 Welded blanks with temperature indicating paints

6.2 Effect of Process Parameters on Process-Induced Damages

6.2.1 Friction Stir Welding

Results from Experimental FSW

Friction stir welding was conducted to weld Al 2024-T3 and Al 7075-T6 with different process parameters as shown in Table 3.7 Two blanks of Al 2024-T3 and Al 7075-T3 were welded respectively at their optimal process conditions from the Table 3.5 & 3.6 to compare the other cases of Bi-Alloy welds.

Mechanical Testing of Welds

Universal Testing Machine with Extensometer was used to determine the mechanical properties of the weld to determine the effect of the process parameters. Tensile test was conducted with the specimens cut out off welded blank from every case study using the MTS machine and data acquisition system operated using LabVIEW. Mechanical properties like Yield Strength and Ultimate Tensile Strength (Table 6.4) were calculated from load-displacement data collected during the tensile tests.

Table 6.4 Mechanical Properties of Al 2024-T3 Welds

#	Rot. Vel.	Tran. Vel	% Elong	% Red. In Area	YS (MPa)	UTS (MPa)
2024-1	1300	2.125	12.37	14.58	211.52	324.28
2024-2	1300	3.625	16	12.13	319.36	437.66
2024-3	1300	5.75	10.67	5.95	313.17	407.01
2024-4	1300	7.625	6.49	9.23	301.38	429.48
2024-5	**1045**	**5.75**	**18.18**	**17.08**	**309.75**	**406.86**
2024-6	1045	3.625	11.4	7.09	330.66	445.23
2024-7	1045	7.625	8.49	8.85	314.57	424.62
2024-8	840	4.625	12.73	11.17	303.69	418.08
2024-9	840	5.75	11.18	10.03	310.56	424.73

Table 6.5 Mechanical Properties of Al 7075-T6 Welds

Case #	% Elongation	% of Red in Area	Yield Strength (MPa)	Ultimate Tensile Strength (MPa)
7075-1	3.5	4.2	330	345
7075-2	5.9	6.3	360	365
7075-3	**7.5**	**9.7**	**360**	**440**

Faster rotational velocity and lower weld speed leads to higher velocity and might lead to melting of the material rather than softening of the material. This is very critical to the quality of the weld since re-solidification might introduce voids and cracks and eventual weakening of the weld. Similarly, slower rotational velocity of the tool and faster weld speed reduces the friction and time of contact of between the tool shoulder and material. As a result, material does not soften enough for the stirring process and a weld is created with poor quality. From the results summarised in Table 6.4, process conditions in case 2024-5 provide the ideal design in the heat generated needed for softening and stirring of the material. Sample 2024-5 creates the weld with the most desirable properties of highest percent elongation, percent reduction in area, and the highest ultimate stress. This means that it has both very high strength and very high ductility which is the main objective of this research.

Al 7075-T6 has different material properties. It has lower liquidus and solidus temperature which reduces the melting point of the material. Hence, lesser heat than the amount of heat required for Al 2024 T3 is required for a successful weld. Thus, case 7075-3 (Table 6.5) creates the weld with high strength and ductility.

Table 6.6 Strength of Al 7075-T6 & Al 2024-T3 Welds

Case #	Yield Strength (MPa)	Ultimate Tensile Strength (MPa)
Bi-A 1	320	395
Bi-A 2	325	416
Bi-A 3	335	355
Bi-A 4	315	355
Bi-A 5	335	354
Bi-A 6	335	360
Bi-A 7	317	351
Bi-A 8	335	364
Bi-A 9	345	355

Mechanical properties listed in Table 6.6 shows the effect of process parameters in the quality of the final weld. Lowest strength is achieved for case Bi-A 3 & Bi-A 7 in both yield and ultimate strength. It can be seen that Bi-A 2 creates the weld with the highest strength compared to the other cases. Bi-A 2 produces weld with 21% higher strength compared to the weakest weld. Insufficient heat is generated with the combination of higher weld feed and lower rotational speed (Bi-A 3); this does not provide the ideal condition for the FSW tool to soften and stir the material. This leads to creation of voids and ultimately to low joint strength. However, lower weld feed and higher rotational speed (Bi-A 7) leads to high friction and excessive heat generated between the tool and blanks. This results in the melting and solidification process, thereby reducing strength of the weld. Process parameters used in Bi-A 2 provides the optimal condition to generate heat in order to soften and perfectly stir the material leading to a complete solid weld. The stress-strain curve shown in Figure 6.12 further re-iterates

the conclusion that case Bi-A 2 generates the high strength weld compared to other cases. Also, it can be seen (Figure 6.13) that Bi-A 2 has a higher strength than one of its parent material (Al 2024-T3) and higher ductility than Al 7075-T6 which proves the success of Tailored Preform Processing.

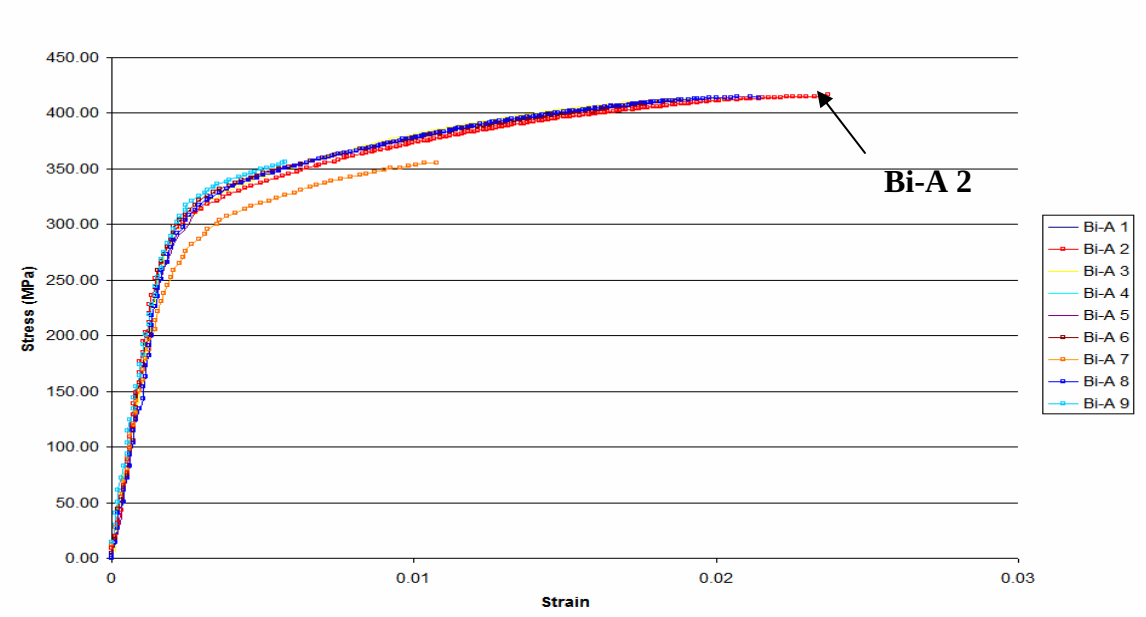


Figure 6.12 Comparison of Stress-Strain Curve of different cases

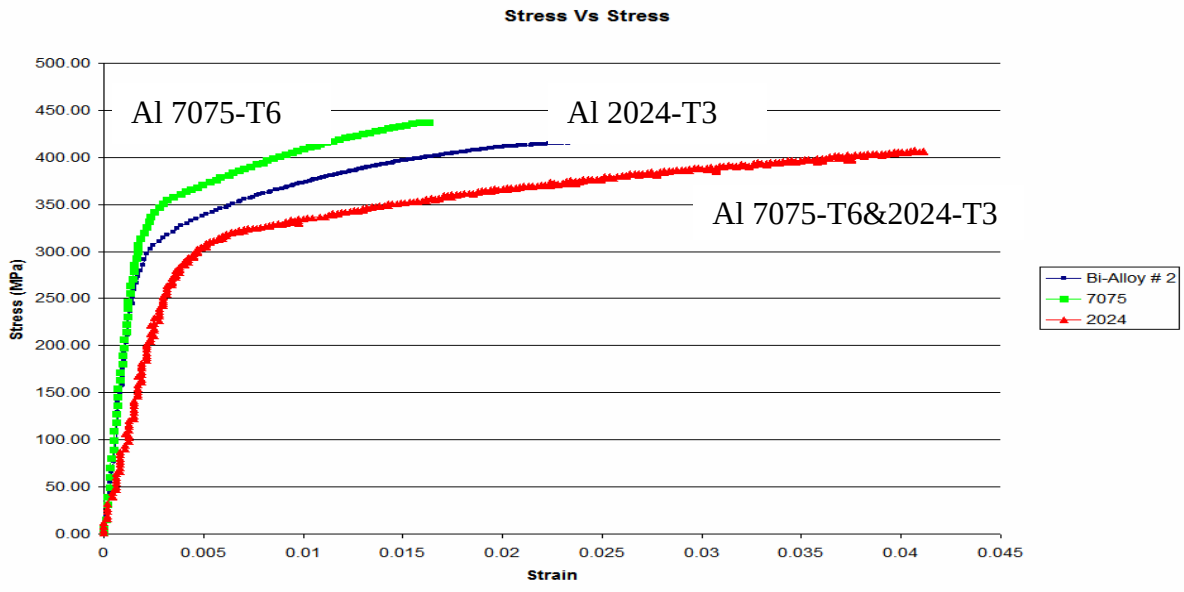


Figure 6.13 Comparison of Stress-Strain Curves of 2024, 7075 and Bi-A 2

All the results were also fitted to three dimensional surfaces relating the effect of the independent variables (rotational speed and weld feed) on one of the quality characteristics of the process. A typical presentation of response surface for ductility is displayed in Figure 6.14. In this plot, ductility of the joint as measured by the reduction of area% was selected as the major response. The process contour map for joint ductility extracted from the surface plot is shown in Figure 6.15. Both figures indicate that rotational speed in the range of 800 to 1100 rpm and welding speed of 4.5-6.5 inch/min (114-165 mm/min) would yield optimum joint ductility. These process maps can be generated for any cases and material properties and will be very useful in selecting the process conditions for any desired properties.

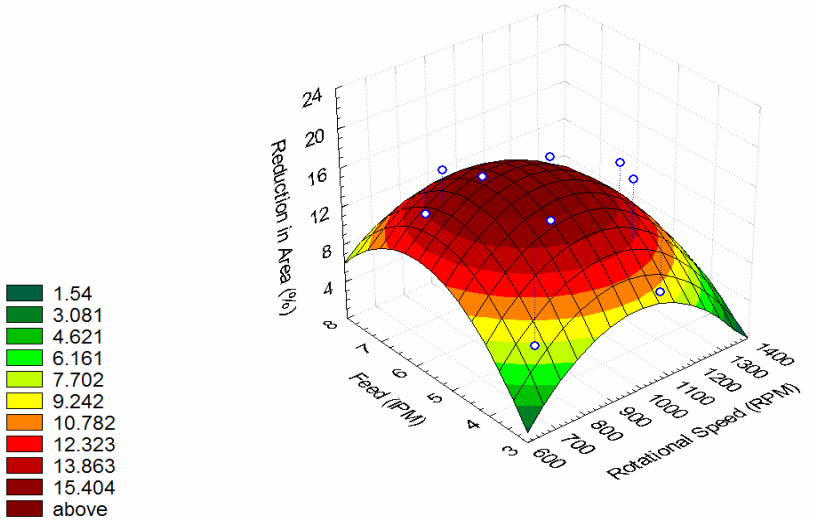


Figure 6.14. Surface plot of the effect of process parameters on ductility of FSW joints

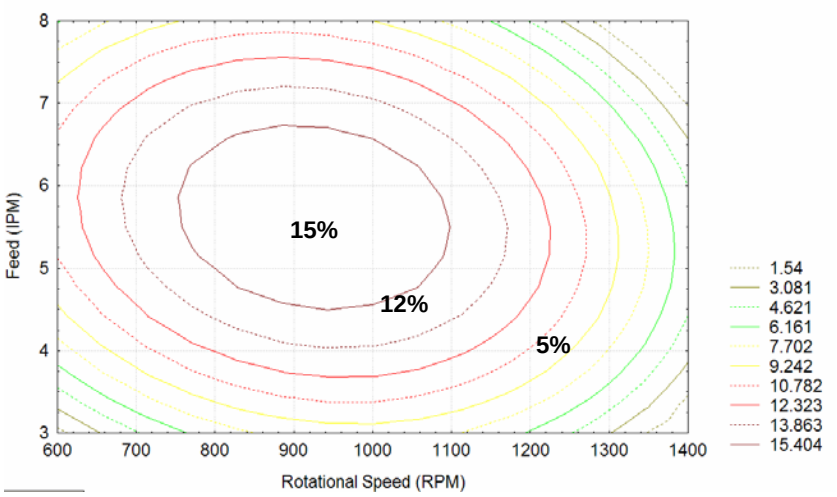


Figure 6.15. Process contour map for ductility of FSW joints.

Table 6.7 Percentage of Reduction in Area and Elongation

Case #	% Red in Area	Elongation %
Bi-A 1	6.8	4.7
Bi-A 2	12.3	9.4
Bi-A 3	9.9	8.7
Bi-A 4	7.2	5.5
Bi-A 5	5.9	4.7
Bi-A 6	12.5	10.2
Bi-A 7	7.9	6.3
Bi-A 8	8.9	7.5
Bi-A 9	7.4	5.5

In addition to the strength of the weld, ductility is another major desired property of the weld. Area under the curve for Bi-A 2 is the maximum which indicates higher ductility of material. Figure 6.13 illustrates the strength and ductility of Bi-A 2 relative to

its parent materials (2024-T3 and 7075-T6). Weld created gets the desired qualities of ductility from Al 2024-T3 and high strength from Al 7075-T6. Percent of reduction in area and elongation for different cases are tabulated in Table 6.7. Bi-A 6 has the highest elongation at 10.2% among the bi-alloy welds followed by Bi-A 2 at 9.4% while both cases have comparable reduction in area.

Thus from the mechanical testing of friction stir welds, the combination of process parameters that creates weld with high strength and ductility is selected as the optimum condition to proceed to the forming operations. Bi-A 2 has the highest strength and second highest ductility next to Bi-A 6. However, Bi-A 6 creates the weld with low strength. Hence, process parameters of 675 RPM and 5.75 IPM in Bi-A 2 was chosen as the optimal conditions to generate the weld with the desired properties.

Results from Numerical Modeling of Friction Stir Welding

Finite Element Analysis of Friction Stir Welding of Bi-Alloys, Al 2024-T3 and Al 7075-T6, is conducted numerically using ABAQUS. Arbitrary Lagrangian-Eulerian (ALE) approach is used to simulate the welding process. The tool is held in a fixed position while rotating about its axis with the prescribed rotational velocity while the materials in both the blanks are given the weld velocity. This approach is successful in modeling friction stir welding of different alloys to eliminate excessive element distortion and premature end of the analysis.

Material properties of Al 2024-T3, Al 7075-T6 and backing plate are given in Table 4.1 and Table 4.2. The process parameters used in this analysis are the same as in case of Bi-A 2.

Figure 6.16 shows the temperature distribution of friction stir welding of Al 7075-T6 and Al 2024-T3. Thermal conductivity of Al 705-T6 is higher than Al 2024-T3 and this is evident in the temperature distribution. As a result, Al 7075-T6 blank heats up more than Al 2024-T3 under the same conditions of process parameters. There is considerable difference in the range of melting point for Al 7075-T6 and Al 2024-T3.

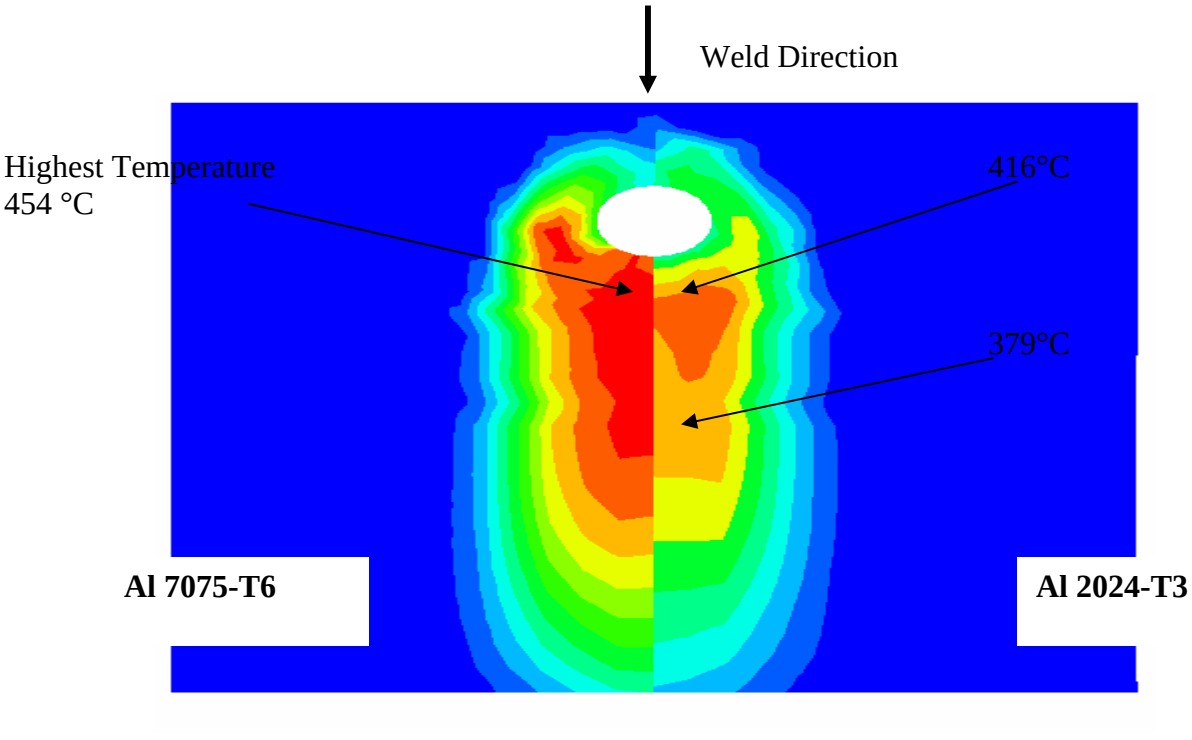


Figure 6.16 Temperature distribution of Al 7075-T6 and Al 2024-T3

This coupled with different thermal conductivity adds to the complexity of welding different alloys since the temperature of the blank is very critical to the quality of weld. Al 7075-T6 has a lower melting point than Al 2024-T3, thereby the process parameters have to be designed so the generated heat is less than the melting point of Al 7075-T6. The process parameters in Bi-A 2 provides the optimal design by generating a temperature of 454°C which is less than the melting point of Al 7075-T6 thereby softening the material. Heat generated in Al 2024-T3 is also sufficient for the softening of the material to stir thereby resulting in the weld with the best desired properties compared to the other cases.

6.2.2 Stretch Forming

Experimental Results from Stretch Forming

Stretch forming is experimentally conducted using the MTS machine (Figure 3.9-3.10). Several stretch-forming experiments are carried to determine the optimal conditions among the different cases in Table 3.7. Stretch-forming experiments were conducted with the final objective of improved ductility and formability and reduced process-induced damages.

All stretch forming processes were conducted with a constant punch velocity of 10mm/min and the blanks are rigidly clamped using the blank holder. The ‘V’ shape notch in the blank holder helps to lock the blank into the groove in the die, thus preventing the blank from slipping as the punch moves in to deform the blank. The stretch forming process is continued until failure of the blank. The welded blank after the failure is shown in Figure 6.17. The final stretch formed blank is shown In Figure 6.18.

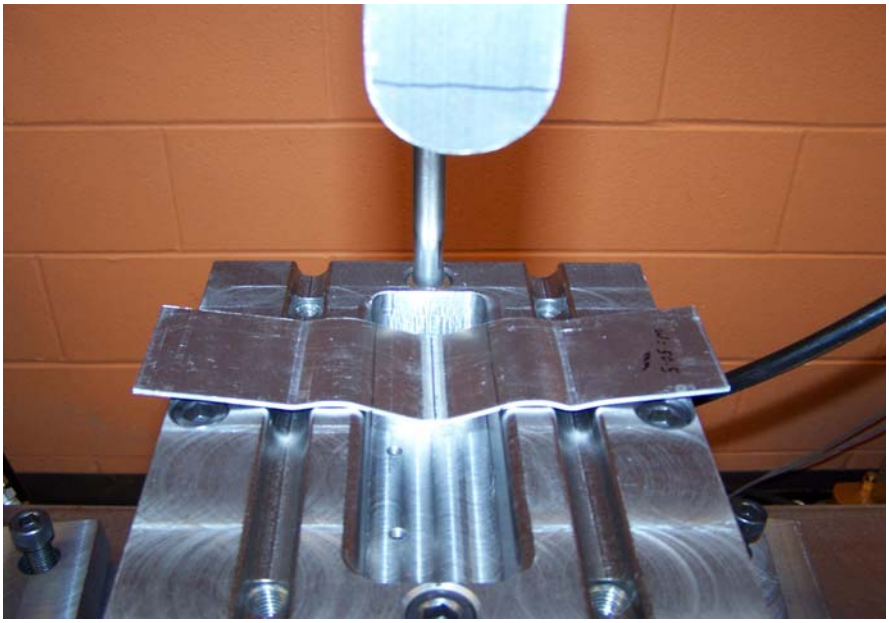


Figure 6.17 Welded blank at the end of stretch forming process

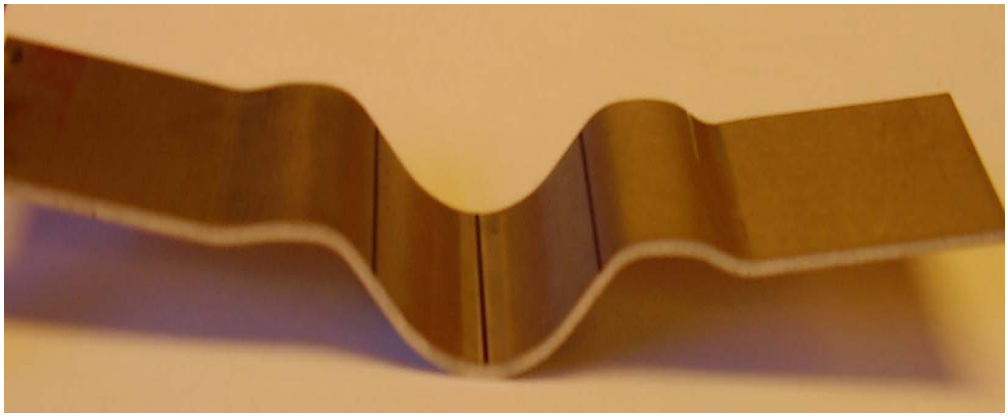


Figure 6.18 Welded blank after stretch forming process

After the stretch forming process, the blanks are removed from the die to conduct post-forming evaluation. Failure of the blank in every case occurs in the clamping area due to the stress concentration rather than in the weld. This clearly shows that the quality of the weld is high and doesn't undergo fracture but rather in the clamping area.

Formability of the welded blanks is the most important property in this research. To evaluate the effect of process parameters on the formability, total depth of plunge at failure for different cases (Table 6.8) are measured. Bi-A 2 shows the highest formability than any other cases. This illustrates that low rotational velocity used during the welding processes increases the ductility of the material.

Table 6.8 Maximum Depth at Failure for Stretch Forming Process

Case #	Depth of Plunge at Failure (mm)
Bi-A 1	12.09
Bi-A 2	16.37
Bi-A 3	14.99
Bi-A 4	9.14
Bi-A 5	11.77
Bi-A 6	15.45
Bi-A 7	7.62
Bi-A 8	9.79
Bi-A 9	9.31

Results from Numerical Simulation

In the stretch forming model, the tailor welded blanks in case Bi-A 2 has been used. Tailor welded blank is modeled with a total 10302 nodes and 10000 elements. The tailor welded blank is made of Al 7075-T6 on the left and Al 2024-T3 on the right and the weld.S4R shell elements are used to model the tailor welded blanks. Material properties of the blank are given in Table 4.1.

Stretch forming process is completed in two steps; during the first step, blank holder is brought down with a displacement to come in contact with the blank. Blank holder force and punch velocity is applied during the second step thereby stretching the blank with the movement of blank. A constant velocity of 10mm/min is applied to the punch until its reaches a depth of 16.5mm while the blank is rigidly clamped at the ends to prevent any slipping and be subjected to pure stretching.

Figure 6.19 shows the finite element model of stretch forming process with different parts like blank holder, punch, die and tailor welded blank. The final deformed blank at the end of stretch forming process is shown in front view (Figure 6.20).

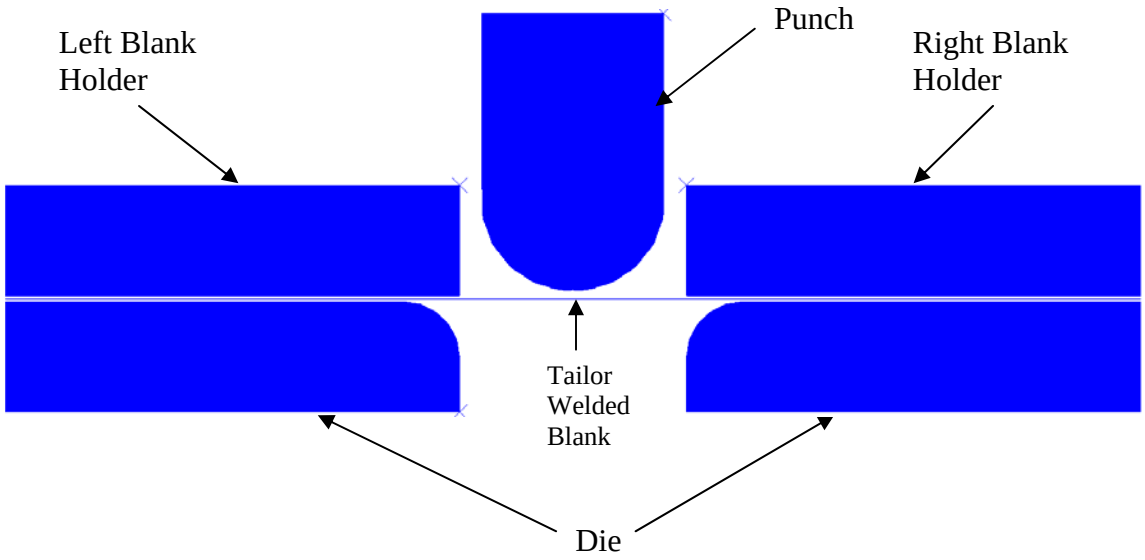


Figure 6.19 Initial tool setup for Stretch Forming process

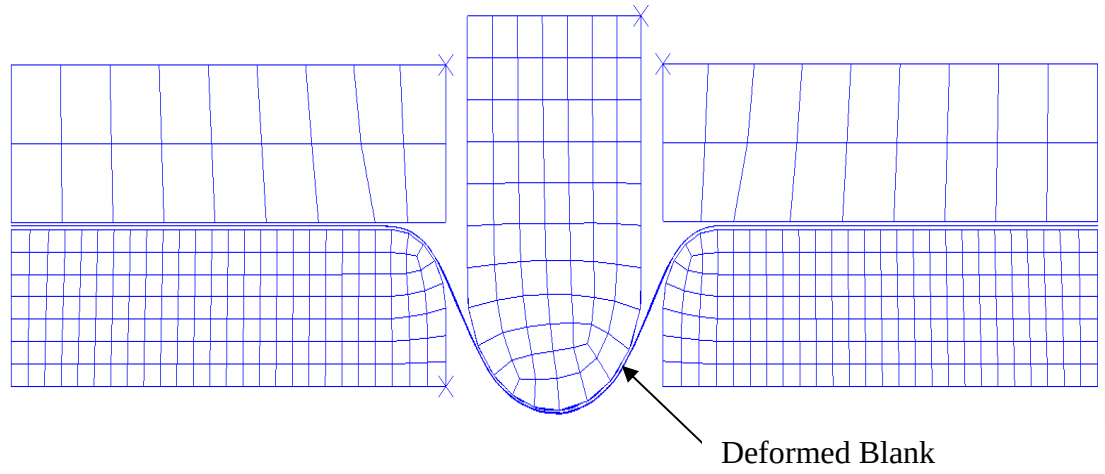


Figure 6.20 Stretch Forming model at the end of stretch forming process

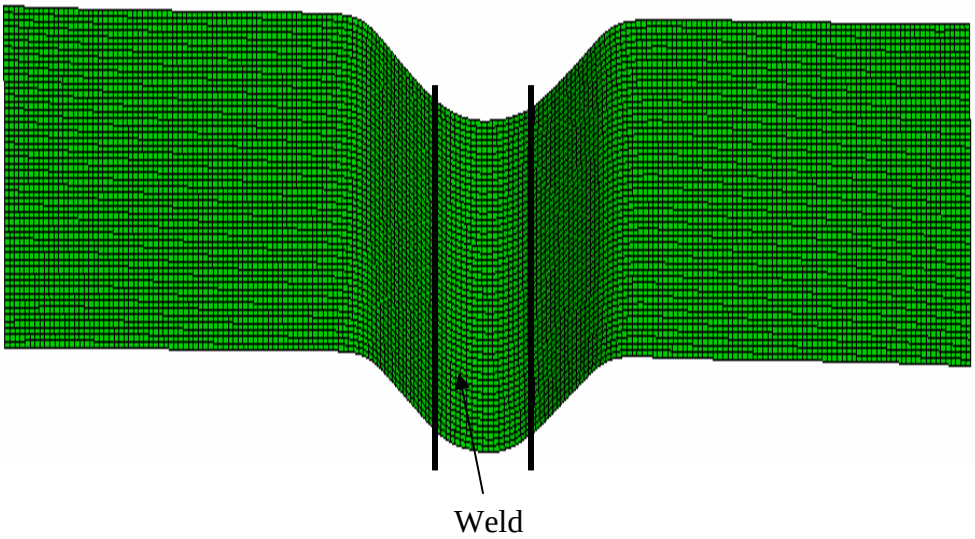


Figure 6.21 Deformed Tailor Welded Blank after stretch forming process

From the Figure 6.21, it can be seen that the weld zone does not undergo any considerable sheet thinning that might lead to the fracture of the blank. This further

provides concrete evidence that the weld zone is solid and doesn't undergo fracture before the clamping area does under stress concentration.

6.2.3 Deep Drawing

Results from Experimental Deep Drawing

A square pan model was used in the current research. Square pan drawing process is a very good representation of several manufacturing process since it involves stretching, bending and drawing processes in more than one axis. The purpose of this experimental study is to drawing ability and formability of the friction stir welded blanks.

MTS servo-hydraulic universal testing machine (Figure 3.13-Figure 3.14) was testing machine was used to conduct the deep drawing experiments. The spring plate plays an important role in exerting and adjusting the blank holding pressure. By tightening the bolts against the die, the springs between the plate and holder are compressed, which generates reaction forces on the holder thus becoming the blank holding force. The actual blank holding force can be controlled by adjusting the height of the springs. The programmable Microprofiler controls the hydraulic force and movement of the actuator on which the die sits on.

In this study, formability of the blanks under different cases is investigated. All the deep drawing processes were conducted for all 9 cases with a constant velocity of 10mm/min and blank holder force of 3.75KN. Gage blocks were used to ensure a constant height thereby a uniform blank holding force on the welded blanks. Deep drawing processes were carried out until the fracture of the blanks. As the punch deforms

the blanks, the load-displacement values were recorded by Data Acquisition system. At the end of the drawing process, the blank (Figure 6.22) is removed from the die to

Figure 6.22 Final product of the square pan

further post-deformation analysis. A strip is cut out at 45° (Figure 6.23) to measure the sheet thickness after the springback analysis.

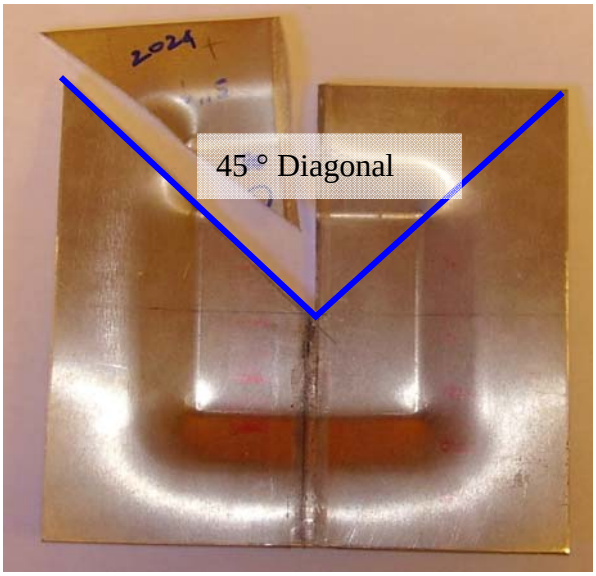


Figure 6.23 Final product of the square pan with cut-off strip

One of the major factors to determine the formability is to calculate the depth of the deformation at fracture. The maximum depth of plunge at failure for the different cases is listed in Table 6.9. The highest depth of plunge occurs for case Bi-A 2 while the least depth occurs for Bi-A7 as a result high rotational speed and slow weld velocity leading to high temperature.

Table 6.9 Maximum Depth at Failure for Drawing Process

Case #	Depth of Plunge at Failure (mm)
Bi-A 1	10.3
Bi-A 2	12.7
Bi-A 3	7.2
Bi-A 4	8.5
Bi-A 5	5.9
Bi-A 6	10.4
Bi-A 7	5.3
Bi-A 8	6.2
Bi-A 9	7.3

Results of Shape Deviation

The square pan formed at the end of drawing process undergoes elastic recovery and deviation from desired shape after the removal of forming tools. The amount of springback at the end of drawing process for different cases is listed in Table 6.10. These results are only for z-component of the shape deviation since shape deviation occurred mainly in the z-direction in our case studies. Generally the shape deviations in x and y

directions are less than 10% of that in z-direction. The maximum springback occurs in flange area is in the negative z direction and minimum springback occurs in cup area in positive z direction. From the results of springback in different cases, the maximum reduction in springback is achieved in case Bi-A 2, where the lowest rotational velocity and intermediate weld velocity was used while maximum shape deviation was observed for case Bi-8 and Bi-9 where high rotational velocity has been used.

Table 6.10 Results of shape deviation in Z direction

Cases	Cup (mm)	Flange (mm)
Bi-A 1	6.8230E-04	-2.9210E-03
Bi-A 2	6.4168E-04	-2.8080E-03
Bi-A 3	6.8700E-04	-2.9880E-03
Bi-A 4	1.0200E-03	-3.2760E-03
Bi-A 5	9.1200E-04	-2.8800E-03
Bi-A 6	1.0259E-03	-3.0930E-03
Bi-A 7	7.8340E-04	-2.8100E-06
Bi-A 8	1.0589E-03	-3.1240E-03
Bi-A 9	1.0392E-03	-2.9950E-03

Al 7075-T6 has higher strength than Al 2024-T6 and springback is proportional to the strength of the material [77]. This results in the maximum springback in the flange area of Al 7075-T6 in all cases.

Results on Sheet Thinning

Thickness distribution is one of the major quality characteristics in sheet metal products. Uniform thickness distribution is always desired for the best formability. However, during deep drawing process, thinning and thickening often appears at different

locations on the workpiece due to non-uniform material flow, especially for complicated non-symmetrical shapes with sharp corners and different material types. Maximum thinning occurs at the cup corner whereas the maximum thickening is in the flange area. The thickening and thinning of the sheet for seven different cases after forming is given in Table 6.11. The original thickness of the sheet is 2 mm.

Table 6.11 Results of sheet thickness

Case #	Minimum Thickness (mm)	Maximum Thickness (mm)
Bi-A 1	1.727	2.170
Bi-A 2	1.796	2.164
Bi-A 3	1.742	2.175
Bi-A 4	1.739	2.189
Bi-A 5	1.721	2.165
Bi-A 6	1.790	2.168
Bi-A 7	1.717	2.172
Bi-A 8	1.778	2.169
Bi-A 9	1.754	2.165

From Table 6.10, the thinning is more significant in case Bi-A 8 and Bi-A 9, where the blank is subjected to excessive heat during welding due to high rotational velocity. The minimum thinning is observed in the case of Bi-A 2 which further re-iterates that Bi-A 2 provides the best design for formability. Also, the weld never undergoes fracture but always in the cup corner area where the thinning in the problem. Fracture also always occurred in Al 7075-T6 and never in Al 2024-T3 since Al 2024-T3 is more ductile.

Results of Weld Line Movement

Deep drawing process of a welded blank made of two dissimilar material leads to non-uniform drawing. Low strength material deforms more than high strength material thereby pushing the weld line into the high strength area. In this research, the welded blank is made of high strength Al 7075-T6 and low strength Al 2024-T3. During this process, low strength Al 2024-T3 deforms more than Al 7075-T6 thus pushing the weld line into 7075 in the cup area (Figure 6.24). However, in the flange area, high strength Al 7075-T6 pushes into Al 2024-T3 [78-80].

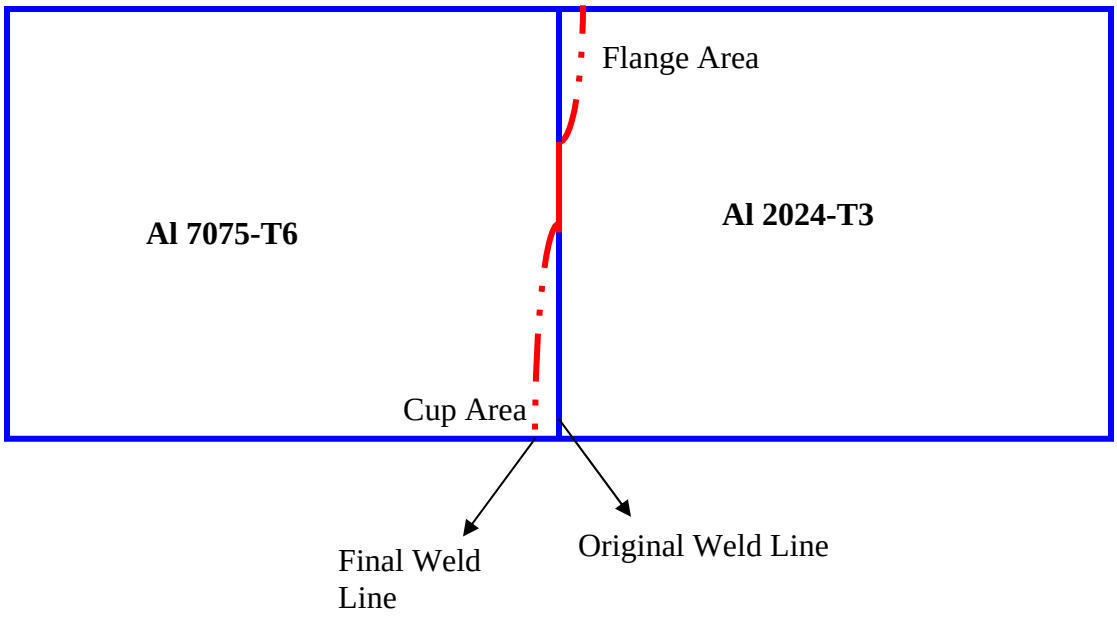


Figure 6.24 Weld Line Movement in Al 7075-T6 and Al 2024-T3 blanks

The weld line movement was measured after the deep drawing process for different cases are tabulated in Table 6.12. The flange area experiences higher flange rate since it undergoes more stretching and bending than the cup area. Bi-A 2 and Bi-A3 are affected the least by the weld line movement in flange area while in cup area, it's almost neglible.

Table 6.12 Results of weld line movement

Case #	Flange (m)	Cup (m)
Bi-A 1	1.72E-03	2.16E-04
Bi-A 2	1.49E-03	2.052E-04
Bi-A 3	1.38E-03	1.94E-04
Bi-A 4	1.69E-03	2.79E-04
Bi-A 5	1.71E-03	2.19E-04
Bi-A 6	1.82E-03	2.45E-04
Bi-A 7	1.65E-03	2.57E-04
Bi-A 8	1.83E-03	3.01E-04
Bi-A 9	1.84E-03	3.24E-04

Results from Numerical Simulation of Deep Drawing

A typical initial and final deformed mesh of the welded blank is shown in Figure 6.25

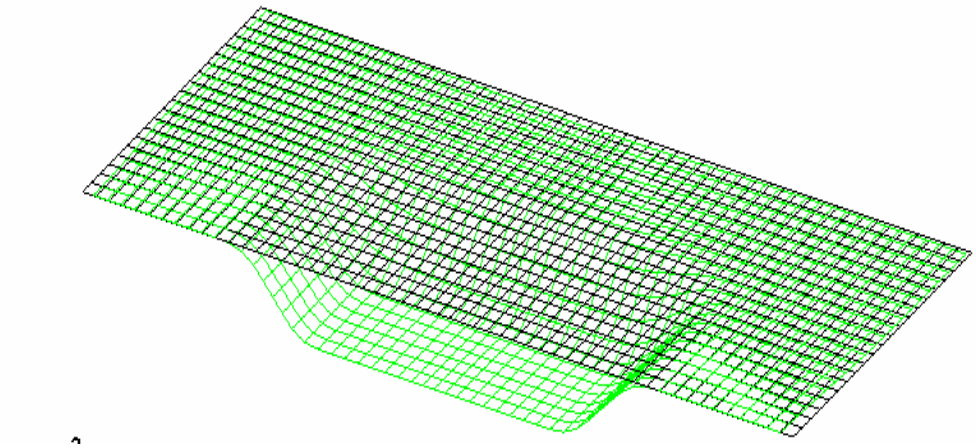


Figure 6.25 Final Deformed Mesh at the end of drawing process

Results of shape deviation

The contour in Figures 6.26 shows the distribution of springback in z direction for case Bi-A 2. With the bottom center point fixed, the pattern of shape deviation cause by the springback phenomenon was characterized by a raise-up of the bottom surface and decreased inclination of the sidewalls, resulting in a downward movement on the flange area and distortion of the flanges.

In Figure 6.26, the springback results in the vertical face of the cup rising by 5.263E-04m, whereas the flange section drops by 2.808E-3m. Thus it can be seen that springback in cup area is only 10 % of the flange area. Hence, the shape deviation in flange area is of more importance. FEA results also show a higher springback in Al 7075-T6 than 2024-T3 which conforms to the experimental results. Figure 6.27 shows the comparison between the results of experimental and numerical results with the optimal desired values. Flange area drops in both experimental and numerical prediction while the cup area rises from the desired position. From these values, it can be seen that the values predicted by the FEA model agrees well with the experimental values.

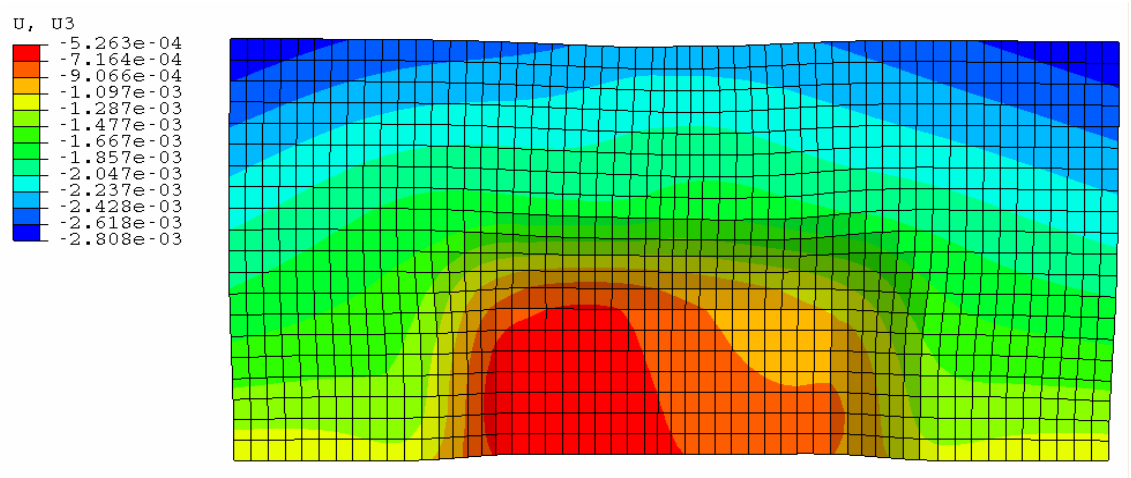


Figure 6.26 Distribution of shape deviation in Z-direction

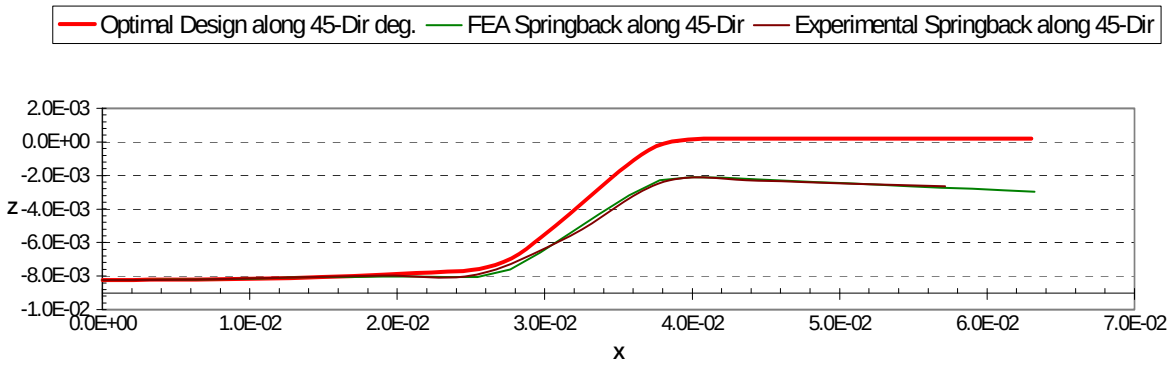


Figure 6.27 Comparison of shape distortion with experimental, FEA and Optimal Results

Result of sheet thinning

Sheet thickness distribution of the square pan for case Bi-A 2 is displayed in Figures 6.28. In forming process of the square pan, the most severe thinning and thickening are expected at bottom and flange corners of the square pan respectively. During the drawing process, the material that flows through the rounded die corners was resisted more than in the straight walls due to the converging geometry. The low material flow results in large tensile stress and thinning on both the side wall and bottom corner in the 45 degree direction (diagonal direction), while the upper corner was thickened due to the hoop compressive stress caused by the converging material flow. The material on the straight sides was easily drawn into the die cavity where only the bending and stretching dominated. This explains the small thickness change along the 0 degree direction (longitudinal direction) on the bottom and sidewall. More material flow also caused larger circumferential compressive stress on the outer flange area and thus was considerably thickened.

The sheet thickness is not uniform throughout its section as shown in Figure 6.25. The sheet thickness has been reduced from its initial thickness of 2mm to 1.796mm at corner of drawn cup, while flange area experiences the thickening to 2.164mm. The thinning of the sheet near the corners is the area of the concern since this might eventually lead to fracture.

In Figure 6.29, the sheet thickness of the blank from experimental studies is plotted with the numerical results for the elements in line from the point of maximum thinning (cup corner area) to point of maximum thickening (flange area). From the plot, it can be seen that there numerical results predict accurate sheet thinning while over estimating the sheet thickening in the flange area by less than 10%..

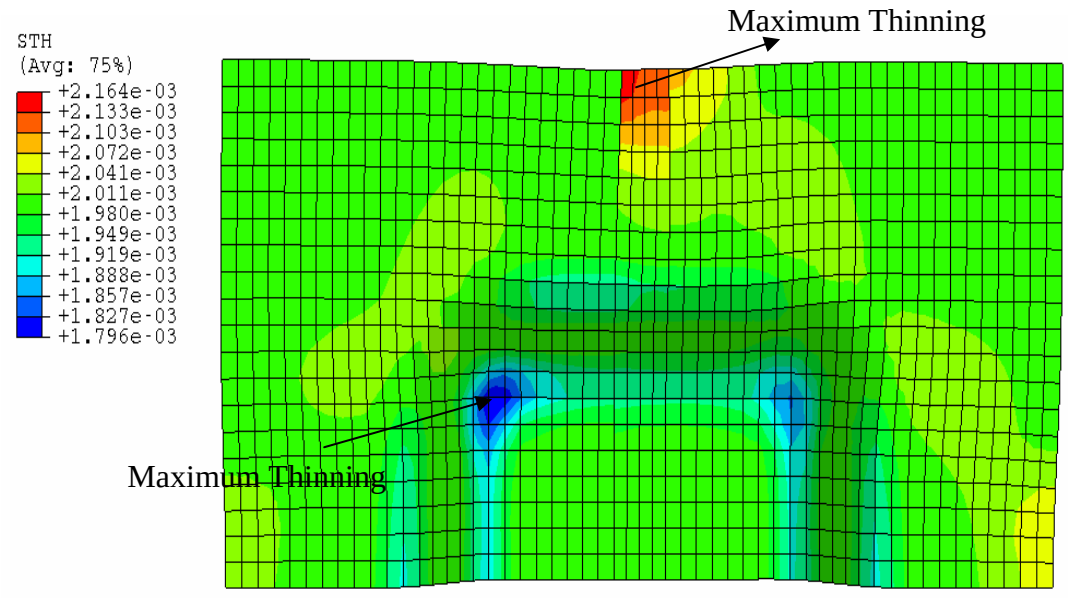


Figure 6.28 Distribution of sheet thickness

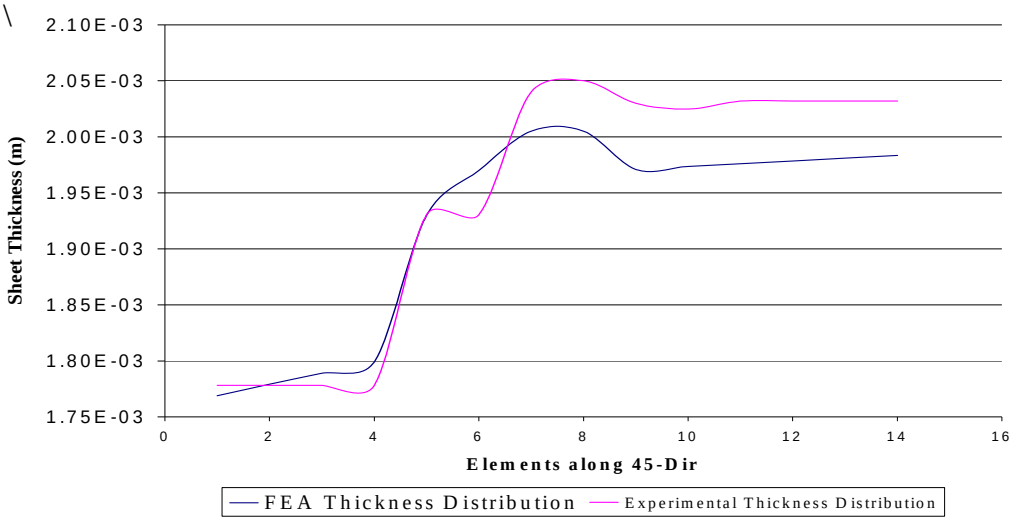


Figure 6.29 Comparison FEA and Experimental Thickness Distribution

6.3 Optimization of Forming Process

One of the main objectives of this research was to integrate the FEA simulation with an optimization to reduce the process-induced damages. In this research, the goal was to reduce the optimal process design with minimum shape deviation, sheet thinning and weld line movement.

Genetic Algorithm has been used as the optimization tool in this research with the finite element model of deep drawing as the objective function. The welded blank from case Bi-A 2 has been used in the drawing process. A constant punch velocity of 10mm/min and sheet thickness of 2mm was used for the optimization process. Weld line movement is one of the process-induced damages that are set to minimize. Al 7075-T6 moves into Al 2024-T3 in the flange area while Al 2024-T3 moves into Al 7075-T6 in the cup area. Weld line movement in the flange area is almost 10% more than the cup area, higher BHF range is set for L_BHF. Design parameters selected were the blank

holding force for Al 7075-T6 (L_BHF) and Al 2024-T3 (R_BHF) and co-efficient of friction. The ranges for these parameters are:

$$7.5KN < L\_BHF\ (Al\ 7075\text{-}T6) < 12.5KN$$

$$3KN < R\_BHF\ (Al\ 2024\text{-}T3 < 6KN$$

$$0.05 < Co\text{-}efficient\ of\ Friction < 0.20$$

Multi-objective function for this optimization study is described by equations (4.5) to (4.11) thus enabling to incorporate three major process-induced damages of springback, sheet thinning and weld line movement. Equal weight factors were assigned to each quantity for the evaluation. Therefore, the objective function is:

$$Obj = 0.334*d_1 + 0.333*d_2+0.333*d_3 \quad (6.1)$$

Where d_1, d_2, and d_3 are the shape deviation, sheet thinning and weld line movement respectively. In the Genetic Algorithms, five individuals were created in every generation. To ensure the survival of the best individual from each generation, it was always passed into (or reproduced in) the next generation based on the “elitist” strategy. In our research, the stopping criterion was specified as a predetermined number of generations at 20^{th} generations.

In Genetic Algorithm, the objective function value was set as “fitness” which describes the ability for each individual to survive in competition with others from each generation. In the current optimization problem, the goal was to minimize the objective function value, so that fitness was actually defined as the "Degree of Process-Induced Damages". The lower the degree of Process-Induced Damages, the higher the fitness. The process parameters generated in the 1^{st} generation normally were distributed uniformly in the searching space of each parameter, which can be observed in Tables 6.12. Such a

wide scattered distribution often introduced individuals with a high degree of springback values, resulted in the high average degree of springback value for the generation, as shown in Figures 6.28. As the process proceeded, the individual with the best value was passed down to the new generation based on the "elitist" strategy, while the other four new individuals were created from those individuals with better (lower) objective values randomly selected. This was based on the principle of “natural selection” in GAs. The result of this process can be shown by the dropping generation average value in Figure 6.28, indicating that not only the best individual was improved but also the whole generation group was improved as the iteration proceeded. This shows the additional advantage of applying GAs in design problems, which offers not only the best design in the final results but also multiple choices from the same generation with similar fitness values but with different configurations of design parameters. It is especially useful in practical design problems in which multiple choices are often demanded for different considerations. While the new individuals were created and evaluated, there were chances that a better result could evolve from one of the new individuals, and became the new “elite” passed to the next generation. This process was first observed in the 3rd generation. As the generation average gradually dropped and approached the best value of the generation - even if the best value had only little or even no improvement - it indicates that the searching space was getting smaller and smaller surrounding the best individual. The individuals in one generation was getting more and more similar to each other, which means the process is approaching a “local optimum”, as shown in convergence graph. Normal optimization schemes, especially those “gradient based” algorithms, stop at this point due to their searching scheme. But based on the random

search scheme of the GAs, the search space "broke out" from this local optimum in the next generation, and explored new areas in the design space; while still keeping the best design from the current generation (as shown in Figure 6.28 at the 7^{th} generation). Due to the high objective values obtained from some of the widely scattered individuals in the generation, the average value of the generation increased dramatically. But as the process proceeds, the search area quickly converged as the generation average dropped, and another individual with a considerable improvement emerged among the new generation, as seen in the 8^{th} generation. This process illustrated another major advantage of the GAs, which not only avoids being trapped in the local optimum, but also gives opportunities to discover a much better solution with a very efficient technique. This advantage, which maybe difficult or even impossible with traditional optimization methods, is especially effective in dealing with non-linear design problems (such as the sheet metal forming process in current research) which involve three or more design parameters with broad design spaces. As the process continued, another searching cycle similar to that from the 1^{st} to 7^{th} generation was implemented and the second local optimum converged at the 11^{th} generation. As the optimization continues, not only the best individual improves but all the other individuals in that generation improves. This is evident in the converging graph. After the 11^{th} generation, another searching cycle began. Theoretically, as such a process continuing, the results should keep improving. But as observed in this research, there is very minimal improvement in the best case design as seen from generation 12. This could be attributed to that small design parameters and their relatively small range. This also can result in the situation that sometimes two or more identical designs were found in the same generation. With more design parameters and wide design space, this would be less

likely to happen. Another possible explanation is the objective function used to evaluate the design. As the three different categories in the equation respond differently to each parameter in each design, it was possible that when the value of one category is decreased, the other two may increased, and resulted in a higher objective value for this design, and dropped out in the next generation. This can be modified by observing the response from current process, adjusting the weight factor for each category depends on the importance or sensitivity, and starting a new optimization process. For our studies, the "global optimum" was claimed as obtained in the 20^{th} generation. If multiple options of the final design are expected, similar designs with slightly higher values can also be selected from generations 10, 13 and 17. By comparing the results of the best individual in the 20^{th} generation (Table 6.13) to both the best and worst individuals in the 1^{st} generation (Table 6.12), it is shown that as the best result improved in all three areas of concern in this research. Comparison of shape deviation, sheet thinning and weld line movement with best and worst case are shown in the side by side contours of Figure 6.29-Figure 6.31. Shape deviation in the flange area is more critical to this research than in the cup area. Also, flange area springbacks about 10 times more than the cup area. The best case in the 20^{th} generation has the shape deviation reduced by 9.12% compared to the worst case in the 1^{st} generation. Sheet thinning has been reduced by 5.61% and most of the improvement in the critical cup corner area. The movement of the weld zone is shown in Figure 6.31. Upper region of the weld (flange area) moves to the right into the Al 2024-T3 and this is of concern in this reason since it's about 10% more than cup area. This weld line movement has been reduced by 3.9%.

From these research results, it can be interpreted that low friction reduces the weld line movement in the flange area. High friction also promotes the situation for excessive sheet thinning. Higher the blank holder force on Al 7075-T6, lower the weld line movement in to Al 2024-T3. However, very high blank holder forces also cause severe thinning along the cup line leading severe fracture. Low blank holder force in Al 2024-T3 reduces the weld line movement in to their material, but very low blank holder force leads to wrinkling of the material. Detailed results for each generation are tabulated in Appendix A.

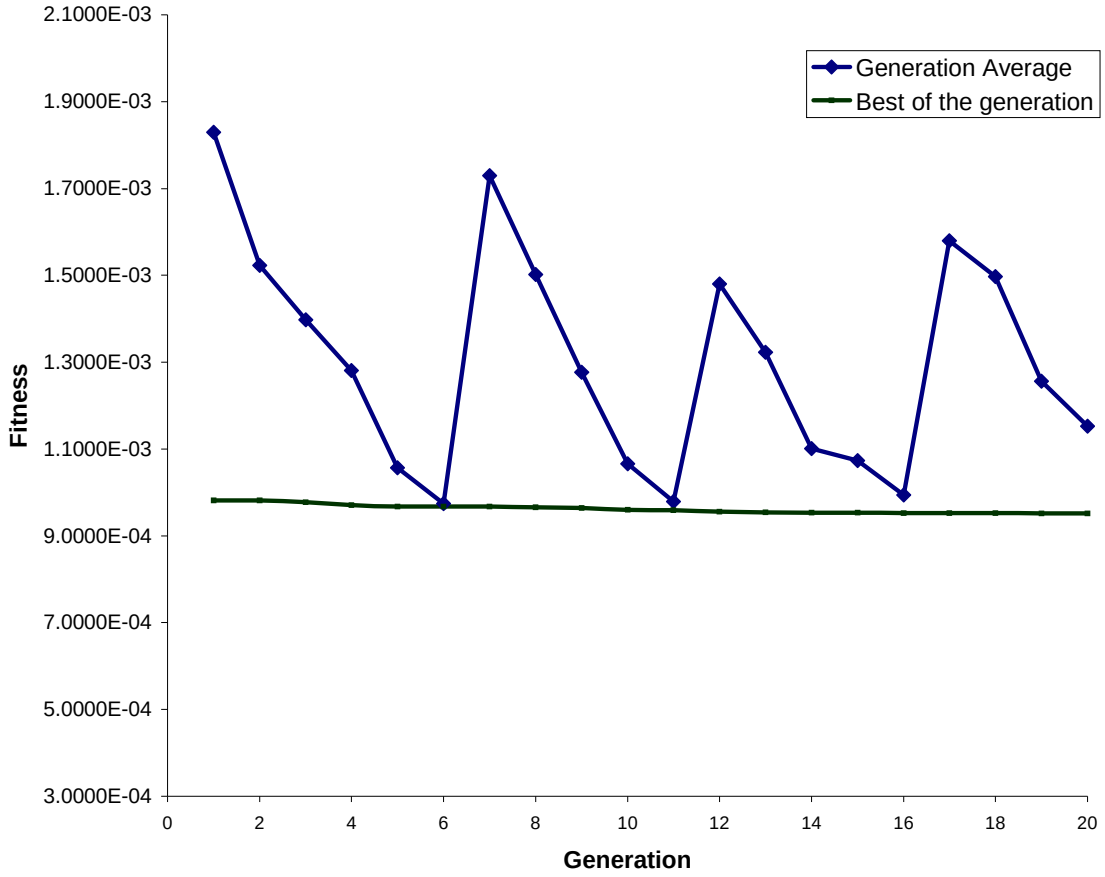


Figure 6.30 The convergence history of the optimization process

Table 6.13 Results from the 1st generation

Gen 1	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	u	Max. (m)	Min. (m)	Max. (m)	Min. (m)	Min. (m)	
1	7.721	3.763	0.124	6.7689E-04	4.3588E-04	1.3140E-03	9.8348E-04	1.7740E-03	1.2532E-03
2	8.677	4.986	0.07	1.2395E-03	6.0609E-04	1.4770E-03	9.8585E-04	1.7681E-03	2.3470E-03
3	10.348	4.282	0.093	8.2534E-04	5.7752E-04	2.8860E-03	7.5020E-04	1.7612E-03	9.8330E-04
4	11.131	5.995	0.118	1.6010E-03	6.4213E-04	1.2933E-03	7.6833E-03	1.7681E-03	2.6818E-03
5	9.817	5.145	0.185	7.8348E-04	5.5085E-04	6.2554E-03	6.2554E-03	1.7225E-03	1.4786E-03

Table 6.14 Results from the 20th generation

Gen 20	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	u	Max. (m)	Min. (m)	Max. (m)	Min. (m)	Min. (m)	
1	7.783	3.673	0.057	6.7689E-04	4.3588E-04	1.3140E-03	9.8348E-04	1.8210E-03	9.6910E-04
2	9.178	5.057	0.122	8.3947E-04	6.0609E-04	1.4770E-03	9.8585E-04	1.7458E-03	1.0310E-04
3	10.216	4.279	0.182	8.0266E-04	5.7752E-04	8.4225E-03	8.7428E-03	1.7405E-03	1.9032E-03
4	8.032	4.648	0.154	8.8073E-04	6.4213E-04	1.2933E-03	7.6833E-03	1.8320E-03	9.7900E-04
5	11.034	5.963	0.121	7.8348E-04	5.5085E-04	6.2554E-03	6.2554E-03	1.7800E-03	1.2194E-03

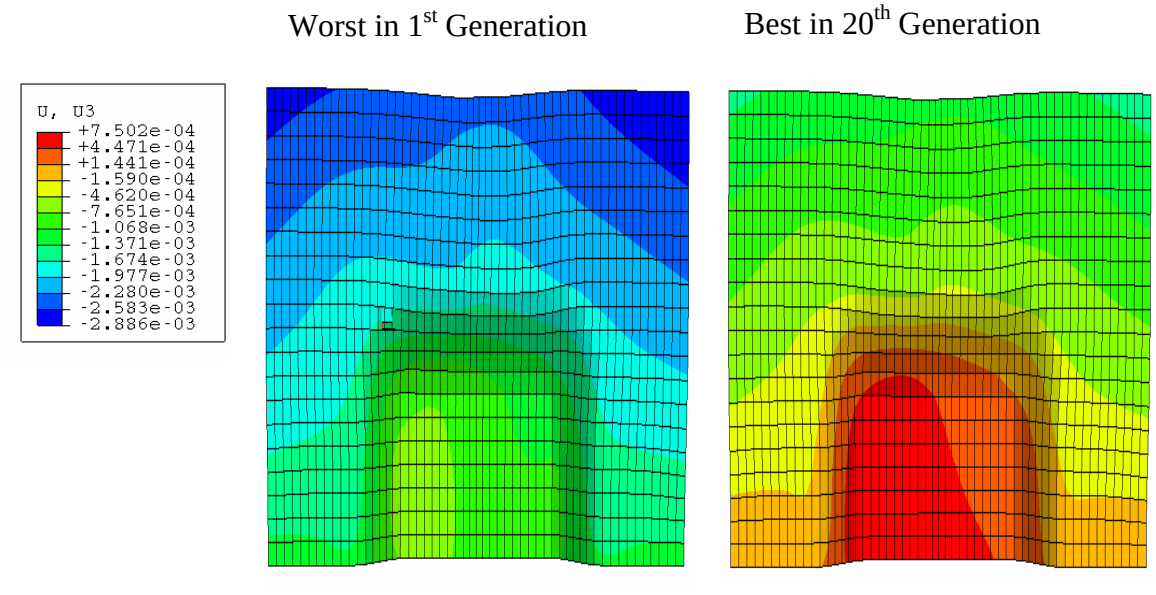


Figure 6.31 Improvement of Shape Deviation by Optimization

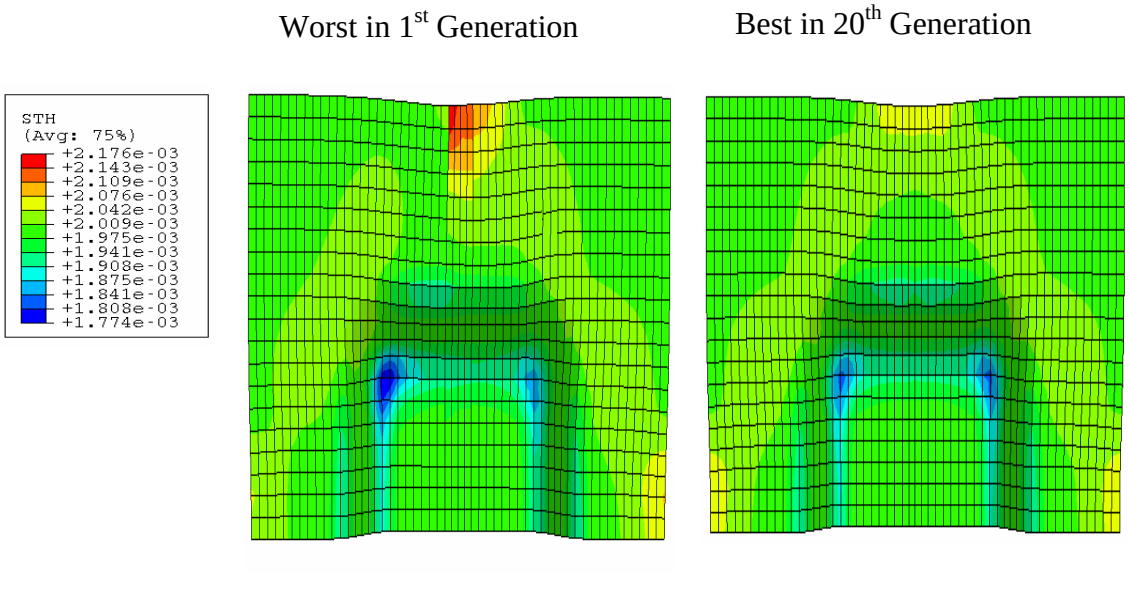


Figure 6.32 Improvement of Sheet Thinning by Optimization

Worst in 1st Generation Best in 20th Generation

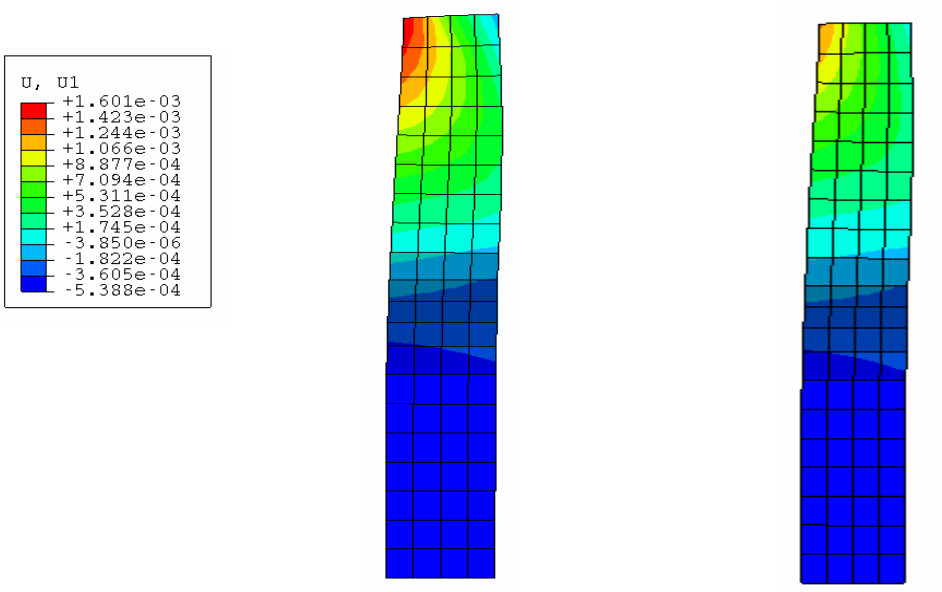


Figure 6.33 Improvement of Weld Line Movement by Optimization

CHAPTER SEVEN

CONCLUSIONS AND RECOMMENDATIONS

The present research aims at exploration of a new processing approach, 'Tailored Preform Processing (TPP)" technology for net-shape manufacturing of monolithic structures. The introduced technology combines friction stir welding (FSW) and forming in order to tailor the properties of the preform to meet design requirements and provide net-shape structural parts. TPP can provide several advantages over current technology for manufacturing of frame structures for the transportation industry. They include significant energy savings, cost and waste reduction, weight reduction, part count reduction, improved damage tolerance, improved material utilization, and improved productivity. The work presented in this Ph.D. dissertation involves understanding of the process behavior and the effects of various process parameters on the properties and integrity of the produced structure. The stated objectives of the proposed work that include providing capability for robust process design, prediction and characterization of process-induced damage and properties of the finished structure are met. This was accomplished through innovative solution to the tailored perform processing problems using the presented experimental and virtual models. The followings are specific conclusions that can be drawn from the presented research:

1. A numerical model using a non-linear finite element method is developed to characterize thermal and deformation behavior along the weld line during friction stir welding process. This model uses coupled temperature-displacement analysis in the

FEM simulation in order to allow for simultaneous determination of temperature, displacement, and mechanical responses.

2. Experimental verification of the proposed numerical model for friction stir welding was conducted using two case studies. The experimental observations confirm the predictions of the numerical model.
3. The results obtained on FSW of different aluminum alloys using the design of experiments offer insights into the effects of the major process parameters in establishing successful FSW joints with optimum formability.
4. Some of the results generated from the experimental investigation were used for constructing process maps for FSW process. These maps are effective tools that can be used by industry as road maps in selecting process designs that satisfy both quality requirements and productivity constraints.
5. An integrated virtual model based on finite element simulation and genetic algorithm search is developed for characterization and management of process-induced distortion in sheet forming of FSW tailored blanks.
6. The new model has the capability of quantifying the effects of process parameters on the evolution of process induced distortion without the need for experimentation using expensive hard tooling.
7. The genetic algorithm module of the integrated model provides a mean for determining the global optimum conditions for minimizing or eliminating distortion (spring-back, sheet thinning and weld line movement).
8. Evaluation of the developed integrated model was conducted using FSW blanks made of two different aluminum alloys (7075-T6 and 2024-T3) in a pan drawing case. The

investigation proved the effectiveness of the new virtual model in characterization of process-induced distortion and in conducting parametric design analysis for minimization of distortion during forming of tailored FSW blanks.

Recommendations for future work:

1. TPP approach has been applied to Aerospace material like 2XXX and 7XXXX in this research. This approach could be expanded to automotive alloys like Al 5XXX and Al 6XXX series.
2 Friction Stir Welding of other different alloys of Aluminum or welding of dissimilar metals like Steel, Titanium, and Magnesium could be explored.
3 The integrated approach could be applied for Lap Welding.
4 Effects of corrosion resistance and fatigue could be investigated for the Tailor Welded Preforms.

REFERENCES

1. B. Irving, "Welding of Tailored Blanks is Hot Issue for Automakers", *Welding Journal*, Aug. 1995, 49-55.

2. J. M. Trogolo, "Evaluation of Tailored Welded Blanks through Technical Cost Modeling", *SAE* #980446, 1998.

3. M. Staisik, and R. H. Wagoner, "Forming of Tailored Welded Aluminum Blanks", *Proc. TMS Conf.*, J.D. Bryant, ed., 1995, 69-74.

4. T. Meinders, A. van den Berg, J. Huetink, "Deep Drawing Simulations of Tailored Blanks and Experimental Verification", *Journal of Materials Processing Technology*, 103 2000, 65-73.

5. A. Sherif El-Gizawy, "Speed Effects on Drawing Al-Zn Alloy Sheets", *Global Symposium on Innovation in Materials Processing and Manufacturing*, TMS Annual Meeting, New Orleans, Louisiana, February, 2001, 361-373.

6. Surendra Chitti Babu, A. Sherif El-Gizawy, "An Integrated Virtual Model for Characterization and Management of Process-induced Damage in Sheet Forming Processes", *Proceedings,4th CIRP Int. Seminar On Intelligent Computation in Mfg. Eng.*, Naples, Italy, June, 2004, 321-326.

7. A. G. Hsu, C. Ulsoy, M. Y. Demeri, "Development of Process Control in Sheet Metal Forming", *Journal of Material Processing Technology*, 127, 2003, 361-368.

8. L. Rui, W. Ya-Jun, L. Zhe, Z. Wei, "Expert System Based on Fuzzy Neural Network for the Optimal Control of the Blank Holder Force", *Shanghay Jiaotong Daxue Xuebao*, 35, n 3, Mar. 2001, 411-415.

9. G. T. Kridli, A. Sherif El-Gizawy, R. Lederich, "Development of Process Maps for Superplastic forming of Welalite 49", *Materials Science and Engineering*, February, A244, 1998, 224-232.

10. K. Lange, *Handbook of Metal Forming*, New York: McGraw-Hill, 1985, 20, 54-55.

11. F. Strasser, *Practical Design of Sheet Metal Stampings*, 1st edition, 1959, Philadelphia: Chilton Co., 43-45.

12. J. A. Schey, *Introduction to Manufacturing Processes*, 3rd edition, 2000, New York: McGraw-Hill, 397-402.

13. H. Takuda, K. Mori, N. Hatta, "The Application of Some Criteria for Ductile Fracture to the Prediction of Fracture Limits", *Journal of Materials Processing Technology*, 1999, 116-121.

14. C. Y. Tang, W. Shen, T. C. Lee, "A Damage Based Criterion for Fracture Prediction in Metal Forming Process", *Journal of Materials Processing Technology*, 1999, 9-83.

15. J. P. D. Correia, G. Ferron, "Wrinkling of Anisotropic Sheet Metals Under Deep Drawing", *Journal De Physique IV*, 105, Mar. 2003, 89-96.

16. Z. Q. Sheng, S. Jirathearanat, T. Altan, "Adaptive FEM Simulation for Prediction of Variable Blank Holder Force in Conical Cup Drawing", *International Journal of Machine Tools and Manufacturing*, 44, 2004, 487-494.

17. N. Krishnan, J. Cao, "Estimation of Optimal Blank Holder Force Trajectories in Segmented Binders Using an ARMA Model", *ASME Journal of Manufacturing Science Engineering*, 125, 2003, 763-770.

18. W. R. Wang, G. L. Chen, Z. Q. Lin, S. H. Li, "Determination of Optimal Blank Holder Force Trajectories for Segmented Binders of Step Rectangle Box Using PID Closed-Loop FEM Simulation", *International Journal of Advanced Technology*, 32, 2007, 1074-1082.

19. T. Yagami, K. Manabe, Y. Yamauchi, "Effect of Alternating Blank Holder Motion of Drawing and Wrinkle Elimination on Deep-Drawability", *Journal of Materials Processing Technology*, Jun 12 2007. 187-191.

20. C. Hsu, A. G. Ulsoy, M. Y. Demeri, "Development of Process Control in Sheet Metal Forming", *Journal of Material Processing Technology*, 127, n 3, Oct. 2003, 361-368.

21. W. M. Thomas, E. D. Nicholas, J. C. Needham, M.G. Murch, P. Templesmith, C. J. Dawes, "Friction Stir Welding", *International Patent Application No. PCT/GB92102203 and Great Britain Patent Application No. 9125978.8*, 1991.

22. Rajiv S Mishra, "Friction Stir Processing Technologies", *Advanced Materials & Processes*, Oct 2003, 43-46.

23. P. Heurtier, C. Desrayaud, F. Montheillet, "A Thermomechanical Analysis of the Friction Stir Welding Process", *Materials Science Forum*, vol. 396-40, Trans Tech Publications, Switzerland, 2002, 1537-1542.

24. F. Palm, U. Hennebohle, V. Erofeev, E. Earpuchin, O. Zaitzev, “Improved Verification of FSW-Process Modeling Relating to the Origin of Material Plasticity”, *Proceedings of Fifth International Symposium of Friction Stir Welding*, TWI Ltd, Metz, France, 2004.

25. P. Ulysee, “Three dimensional modeling of the friction stir welding process”, *Int. J. Mach. Tools Manufacture*, 42 (2002) 1549-1557.

26. Surendra Chitti Babu and A. Sherif El-Gizawy, “Characterization of Friction Stir Welding Process Behavior using Numerical and Physical Modeling Techniques”, *International Conference on Mechanical Design and Production*, January 2008, Cairo, Egypt.

27. Surendra Chitti Babu and A. Sherif El-Gizawy, “Development of Virtual Model for Process Design of Friction Stir Welding Operations,” *Proceedings of Virtual Concept*, November 2006, Playa Del Carmen, Mexico.

28. Y. J. Chao, X. Qi, “Heat Transfer and Thermo-Mechanical Analysis of Friction Stir Joining of AA 6061-T6 Plates”, J. Mater. Process. Manuf. Sci 7, 1998, 215-233.

29. F. Frigaard, F. Grong, O. T. Midling, “A Process Model for Friction Stir Welding of Age Hardening Aluminum Alloys”, *Metallurgical Material Transactions*, May 2001, 32 A.

30. M. Zahedul, H. Khandkar, J. Khan, “Thermal Modeling of Overlap Friction Stir Welding for Al Alloys”, *Journal of Material Processing and Manufacturing Science*, October 2001, 10(2), 91-106.

31. A. Askari, S. Silling, B. London, M. Mahoney, “Modeling and Analysis of Friction Stir Welding Processing”, *Friction Stir Welding and Processing*, TMS, Warrendale, PA, 2001, 43-54.

32. S. Xu, X. Deng, “Two and Three Dimensional Finite Element Models for the Friction Stir Welding Process”, 4th *Int. Symp. on Friction Stir Welding*, Park City, UT, USA. 2003.

33. Y. J. Chao, Y. Wang, K. W. Miller, “Effect of Friction Stir Welding on Dynamic Properties of AA2024-T3 and AA7075-T7351”, *Welding Research Supplement*, PP 196-200, 2001.

34. P. Cavaliere, R. Nobile, F. W. Panella, A. Squillace, “Mechanical and Microstructural Behavior of 204-7075 Aluminum Alloy Sheets Joined by Friction Stir Welding”, *International Journal of Machine Tools & Manufacture*, 46, 588-594, 2006.

35. P. Cavaliere, E. Cerri, A. Squillace, “Mechanical Response of 2024-7075 Aluminum Alloys Joined by Friction Stir Welding”, *Journal of Materials Science,* 40, 2005, 3669-3676.

36. K. Kimapong, T. Wanabe, “Friction Stir Welding of Aluminum Alloy to Steel”, *Welding Journal*, 83 (10), 277-282, 2004.

37. X. Zhu, Y. J. Chao, “Numerical Simulation of Transient Temperature and Residual Stresses in Friction Stir Welding of 304L Stainless Steel”, *Journal of Materials Processing Technology,* 146, 263-272, 2004.

38. H. Uzun, C. D. Dalle, A. Argagnotto, T. Ghidini, C. Gambar, “Friction Stir Welding of Dissimilar Al 6013-T4 to X5CrNi18019 Stainless Steel”, *Materials and Design*, 26, 41046, 2005.

39. J. A. Baumann, R. J. Lederich, D. R. Bolser, R. Talwar, “Property Characterization of 2024Al/7075Al Bi-Alloy Friction Stir Welded Joints”, *Friction Stir Welding and Processing III,* 199-207, 2003.

40. J. H. Ouyang, R. Kovacevic, "Material Flow and Microstructure in the Friction Stir Butt Welds of the Same and Dissimilar Aluminum Alloys", *Journal of Materials Engineering & Performance*, 11, 51-63, 2002.

41. H. Larsoon, L. Karlsson, S. Stoltz, and E. L. Bergqvist, *Second International Symposium on Friction Stir Welding*, Gotherburg, Sweden, 2000.

42. E. Doege, H. Dohrmann, R. Kosters, “Simulation and Optimization of the Forming Process of Tailored Blanks” *Numisheet ’96*, 1996, 199-204.

43. T. Meinders, A. van den Berg, J. Huetink, “Deep Drawing Simulations of Tailored Blanks and Experimental Verification”, *Journal of Material Processing Technology* ,103, 2000, 65-73.

44. Y. Lee, M. J. Worswick, M. Finn, W. Christy, M. Jain, “Simulated and Experimental Deep Drawing of Aluminum Alloy Tailor Welded Blanks”, *21st Biennial Congress*, Ann Arbor, Michigan, USA, June 2000.

45. A. Mustafa Ahmedtoglu, D. Brouswers, L. Shulkin, L. Taupin, G. L. Kinzel, T. Altan, “Deep Drawing of Round Cups from Tailor-Welded Blanks”, *Journal of Material Processing Technology,* 53(1995) 684-694.

46. M. F. Shi, K. M. Pickett, K. K. Bhatt, “Formability Issues in the Application of Tailor Welded Blank Sheets”, *SAE paper no. 930278,* International congress and Exposition, Detroit, Michigan, 1993.

47. L. C. Chan, S. M. Chan, C. H. Cheng, T. C. Lee, "Formability and Weld Zone Analysis of Tailored-Welded Blanks for Various Thickness Ratios", *J. Eng. Mater. And Techno.*, ASME, 2005, 127, 179-185.

48. F. J. Saunders, R. H. Wagoner, "Forming of Tailored Welded Blanks", *Metall. Mater.* Trans. A, 27(A/9), 2605-2616.

49. B. Kinsey, N. Krishnan, J. Cao, "A Methodology to Reduce and Quantify Wrinkling in Tailor Welded Blank Forming", *Int. J. Materials and Product Tech*, Vol 21, Nos. 1/2/3, 2004,154 – 168.

50. S. He, X. Wu, S. J. Hu, "Formability Enhancement for Tailor-Welded Blanks Using Blank Holding Force Control", *Journal of Manufacturing Science and Engineering*, Volume 125, Issue 3, August 2003, 461-467.

51. J. Kusiak, "A Technique of Tool-Shape Optimization in Large Scale Problems of Metal Forming". *Journal of Materials Processing Technique*, Vol. 57, 1996, 79-84.

52. J. S. Shu, C. Hung, "Finite Element Analysis and Optimization of Springback Reduction: the 'Double-Bend' Technique", *International Journal of Machine Tools & Manufacture*, 36, No. 4, 1996, 423-434.

53. I. Chou, C. Hung, "Finite Element Analysis and Optimization on Springback Reduction", *International Journal of Machine Tools & Manufacture*, 39, 1999, 517-536.

54. C. Liu, "Springback Reduction in U Channel-'Double Bend' Technique", *J. Appl. Metalworking*, 3, 1984, 148-156.

55. S. Oral, H. Darendeliler, "Optimum Die Profile for the Cylindrical Bending of Plates", *Journal of Materials Processing Technology*, 70, No. 1-3, Oct 1997, 151-155.

56. L. M. Kutt, J. A. Nardiello, P. L. Ogilvie, A. B. Pifko, J, M, Papazian, "Non-linear Finite Element Analysis of Springback", *Communications in Numerical Methods in Engineering*, 15, 1999, 33-42.

57. D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

58. J. H. Holland, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", Cambridge, Mass.: MIT Press, 1992.

59. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Berlin; New York: Springer-Verlag, 1992.

60. M. Gen, R. Cheng, "Genetic Algorithms & Engineering Optimization", New York: Wiley, 2000.

61. Surendra Chitti Babu, A. Sherif El-Gizawy, "An Integrated Virtual Model for Characterization and Management of Process-Induced Damage in Sheet Forming Processes", *Proceedings,4th CIRP Int. Seminar On Intelligent Computation in Mfg. Eng.*, Naples, Italy, June, 2004, 321-326.

62. *ABAQUS Theory Manual*, Version 6.6, Hibbitt, Karlsson & Sorensen, Inc., 2006.

63. *ABAQUS/Explicit User's Manual*, Version 6.6, Hibbitt, Karlsson & Sorensen, Inc., 2006.

64. *ABAQUS/Standard User's Manual*, Version 6.6, Hibbitt, Karlsson & Sorensen, Inc., 2006.

65. *Metals Handbook*, Vol. 2, 10th ed., 1990, p114-116.

66. The Aluminum Association, *Aluminum Standards and Data*, 2nd ed., 1970, p6-13.

67. ASTM, *Annual Book of ASTM Standards*, Vol. 02.02, 1999, p394-404.

68. A. Makinouchi, E. Nakamachi, E. Onate, R. H. Wagoner, *Proceedings of the 2nd International Conference, NUMISHEET'93, Numerical Simulation of 3-D Sheet Metal Forming Process*, Isehara, Japan, 31 August – 2 September, 1993.

69. C. H. Toh, S. Kobayashi, "Deformation Analysis and Blank Design in Square Cup Drawing", *Int. J. Mach. Tool Des. Res*., Vol. 25, No. 1, 1985, 15-32.

70. S. D. El-Wakil, M. N. E. M. Kamal, A. H. Darwish, "Mechanics of the Square Box Drawing Operation of Aluminum Blanks", *Sheet Metal Industries*, Aug. 1980, p679-759.

71. E. Onate, P. Cendyoa, J. Rojek, J. Miquel, "A Simple Thin Shell Triangle with Translational Degrees of Freedom for Sheet Stamping Analysis", *Proceedings of the 3rd International Conference, NUMISHEET '96, Numerical Simulation of 3-D Sheet Metal Forming Process*, Dearborn, Michigan, 29 September – 3 October, 1996, 102-111.

72. D. Y. Yang, D. W. Lee, S. W. Lee, J. W. Yoon, "Effective Blankholding Gap Control Algorithm for Implicit and Explicit Codes in Sheet Metal Forming Processes", *Proceedings of the 3rd International Conference, NUMISHEET '96,*

Numerical Simulation of 3-D Sheet Metal Forming Process, Dearborn, Michigan, 29 September – 3 October, 1996, p32-39.

73. A. C. van den Berg, T. Meinders, B. Stokman, "Deep Drawing Simulation of Tailored Blanks", *Automotive Tailored Blanks, Netherlands*, 2000, 133-144.

74. K. M. Zhao, B. K. Chun, J. K. Lee, "Finite Element Analysis of Tailor-Welded Blanks", *Finite Elements in Analysis and Design*, 37, 2001, 117-130.

75. Surendra Chitti Babu, A. Sherif El-Gizawy and Tai-Kun Yeh, "An Approach Combining Finite Element and Global Optimization Methods for Prediction and Control of Process-Induced Damage (Distortion) in Sheet Formed Products", *International Journal of Machine Tools and Manufacture*, Under Review.

76. Surendra Chitti Babu and A. Sherif El-Gizawy, "An Integrated Virtual Model for Characterization and Management of Process-induced Damage in Sheet Forming Processes", *AmeriPAM Conference*, November 2004, Troy, Michigan.

77. K. Mori, K. Akita ad Y. Abe, "Springback Behavior in Bending of Ultra-High-Strength Steel Sheets Using CNC Servo Press", *International Journal of Machine Tools and Manufacture*, 47, 2007, 321-325.

78. Y. Choi, Y. Heo, H. Y. Kim, D. Seo, "Investigations of Weld-line Movements for the Deep Drawing Process of Tailor Welding Blanks", *Journal of Material Processing Technology*, 108, 2000, 1-7.

79. H. Y. Kim, D. Seo, Y. Heo, Y. Choi, "Characteristics of Weld Line Movements for the Deep Drawing with Draw Beads of Tailor-Welded Blanks", *Journal of Materials Processing Technology*, 111, 2001, 164-169.

80. S. K. Panda, D. R. Kumar, "Numerical and Experimental Studies on Strain Distribution and Weld Live Movement in Stretch Forming of Tailor Welded Blanks", *NUMIFORM '07, Materials Processing and Design: Modeling, Simulation and Application.*

Appendix A

(ABAQUS input file)

A sample input file for ABAQUS.

```
*Heading
 Used for Springback
** Job name: Ddraw_For_Sback_Enh Model name: Model-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Blank_Shell
*End Part
**
*Part, name=Die
*End Part
**
*Part, name=Holder-L
*End Part
**
*Part, name=Punch
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Die-1, part=Die
*Node
      1,  0.129999995,           0.,           0.
      2,  0.129999995,           0., 0.0500000007
      3, 0.0480000004,           0., 0.0500000007
      4, 0.0379999988,           0.,           0.
      5, 0.0379999988,           0., 0.0399999991
      6, -0.129999995,           0., 0.0500000007
      7, -0.0480000004,           0., 0.0500000007
      8, -0.129999995,           0.,           0.
      9, -0.0379999988,           0.,           0.
     10, -0.0379999988,           0., 0.0399999991
     11, -0.0379999988, 0.0280000009,           0.
     12, -0.0379999988, 0.0280000009, 0.0399999991
     13,           0.,  0.129999995,           0.
     14, -0.0280000009, 0.0379999988,           0.
     15, 0.0280000009, 0.0379999988,           0.
     16, 0.0379999988, 0.0280000009,           0.
     17,           0.,  0.129999995, 0.0500000007
     18, 0.0379999988, 0.0280000009, 0.0399999991
     19, -0.0280000009, 0.0480000004, 0.0500000007
     20, -0.0280000009, 0.0379999988, 0.0399999991
     21, 0.0280000009, 0.0379999988, 0.0399999991
     22, 0.0280000009, 0.0480000004, 0.0500000007
     23, -0.0480000004, 0.0280000009, 0.0500000007
     24, 0.0480000004, 0.0280000009, 0.0500000007
*Element, type=R3D4
   1,  115,  116,  993,  990
   2,  116,  117,  994,  993
```

```
   3,  117,  118,  995,  994
   4,  118,  119,  996,  995
   5,  119,  120,  997,  996
   6,  120,  121,  998,  997
   7,  121,  122,  999,  998
   8,  122,  123, 1000,  999
   9,  123,  124, 1001, 1000
  10,  124,  125, 1002, 1001
  11,  125,  126, 1003, 1002
  12,  126,  127, 1004, 1003
  13,  127,  128, 1005, 1004
  14,  128,  129, 1006, 1005
  15,  129,  130, 1007, 1006
  16,  130,  131, 1008, 1007
  17,  131,  132, 1009, 1008
  18,  132,  133, 1010, 1009
  19,  133,  134, 1011, 1010
  20,  134,  135, 1012, 1011
  21,  135,  136, 1013, 1012
  22,  136,  137, 1014, 1013
  23,  137,  138, 1015, 1014
  24,  138,  139, 1016, 1015
  25,  139,  140, 1017, 1016
*Node
  17743,  7.95994166e-18,  0.129999995, 0.0500000007
*Nset, nset=Die-1-RefPt_, internal
17743,
*Elset, elset=Die-1, generate
     1,  17740,      1
*End Instance
**
*Instance, name=Holder-R-1, part=Holder-L
          0.,           0.,        0.052
*Node
      1,           0.,  0.129999995, 0.0199999996
      2,           0.,  0.129999995,           0.
      3,           0., 0.0450000018, 0.0199999996
      4,           0., 0.0399999991, 0.0149999997
      5,           0., 0.0399999991,           0.
      6,  0.129999995,           0.,           0.
      7,  0.129999995,           0., 0.0199999996
      8, 0.0450000018,           0., 0.0199999996
      9, 0.0399999991,           0., 0.0149999997
     10, 0.0399999991,           0.,           0.
     11, 0.0299999993, 0.0399999991,           0.
     12, 0.0299999993, 0.0399999991, 0.0149999997
     13, 0.0299999993, 0.0450000018, 0.0199999996
     14, 0.0399999991, 0.0299999993,           0.
     15, 0.0399999991, 0.0299999993, 0.0149999997
     16, 0.0450000018, 0.0299999993, 0.0199999996
     17,           0.,  0.120233029,           0.
     18,           0.,  0.110853069,           0.
     19,           0.,  0.101697981,           0.
     20,           0., 0.0929246172,           0.
     21,           0., 0.0841150135,           0.
     22,           0., 0.0758666545,           0.
     23,           0.,  0.068053633,           0.
```

```
     24,           0., 0.0614104383,           0.
     25,           0., 0.0553878546,           0.
    492,   0.12938948, 0.0125842206, 0.0133333327
*Element, type=R3D4
 1,  20, 131, 135,  19
 2, 131, 132, 136, 135
 3, 132,  34,  35, 136
 4,  19, 135, 137,  18
 5, 135, 136, 138, 137
 6, 136,  35,  36, 138
 7,  18, 137, 139,  17
 8, 137, 138, 140, 139
 9, 138,  36,  37, 140
10,  17, 139,  39,   2
11, 139, 140,  38,  39
12, 140,  37,   1,  38
13,  22, 133, 141,  21
14, 133, 134, 142, 141
15, 134,  32,  33, 142
16,  21, 141, 131,  20
17, 141, 142, 132, 131
18, 142,  33,  34, 132
19,  32, 134, 144,  31
20,  31, 144, 145,  30
21,  30, 145, 146,   3
22,   3, 146, 147,  29
23,  29, 147, 143,   4
24, 134, 133, 149, 144
25, 133,  22,  23, 149
490, 492, 117,   7,  52
*Node
    493,  0.091923885,  0.091923885, 0.0199999996
*Nset, nset=Holder-R-1-RefPt_, internal
493,
*Nset, nset=_PickedSet5, internal
 493,
*Elset, elset=Holder-R-1, generate
   1,  490,    1
*Element, type=MASS, elset=_PickedSet5_Holder_MASS_
491, 493
*Mass, elset=_PickedSet5_Holder_MASS_
1.,
*Element, type=ROTARYI, elset=_PickedSet5_Holder_ROTI_
492, 493
*RotaryI, elset=_PickedSet5_Holder_ROTI_
1., 1., 1., 0., 0., 0.
*End Instance
**
*Instance, name=Punch-1, part=Punch
       0.035,       -0.035,        0.152
       0.035,       -0.035,        0.152, 0.0349999999999999,
0.965,        0.152,         180.
*Node
      1,           0., 0.0350000001,           0.
      2, 0.0700000003, 0.0350000001,           0.
      3,  0.061999999, 0.0350000001,  0.100000001
      4, 0.0700000003, 0.0350000001, 0.0920000002
```

```
      5, 0.00800000038, 0.0350000001,  0.100000001
      6,           0., 0.0350000001, 0.0920000002
      7,           0., 0.0599999987, 0.0920000002
      8,           0., 0.0599999987,           0.
      9, 0.00800000038, 0.0599999987,  0.100000001
     10, 0.0599999987,  0.061999999,  0.100000001
     11,  0.061999999, 0.0599999987,  0.100000001
     12, 0.00999999978,  0.061999999,  0.100000001
     13, 0.00999999978, 0.0700000003, 0.0920000002
     14, 0.0599999987, 0.0700000003, 0.0920000002
     15, 0.0700000003, 0.0599999987, 0.0920000002
     16, 0.0700000003, 0.0599999987,           0.
     17, 0.0599999987, 0.0700000003,           0.
     18, 0.00999999978, 0.0700000003,           0.
     19,           0., 0.0350000001, 0.0898633301
     20,           0., 0.0350000001, 0.0878497362
     21,           0., 0.0350000001, 0.0858768746
     22,           0., 0.0350000001, 0.0838951319
     23,           0., 0.0350000001, 0.0819213092
     24,           0., 0.0350000001, 0.0799532533
     25,           0., 0.0350000001, 0.0779879838
   6252, 0.000230549529, 0.0621349104, 0.00200176635
*Element, type=R3D4
   1,  115,  554,  619,  114
   2,  554,  555,  620,  619
   3,  555,  556,  621,  620
   4,  556,  557,  622,  621
   5,  557,  558,  623,  622
   6,  558,  559,  624,  623
   7,  559,  560,  625,  624
   8,  560,  561,  626,  625
   9,  561,  562,  627,  626
  10,  562,  563,  628,  627
  11,  563,  564,  629,  628
  12,  564,  565,  630,  629
  13,  565,  566,  631,  630
  14,  566,  553,  632,  631
  15,  553,  567,  633,  632
  16,  567,  568,  634,  633
  17,  568,  569,  635,  634
  18,  569,  570,  636,  635
  19,  570,  571,  637,  636
  20,  571,  572,  638,  637
  21,  572,  573,  639,  638
  22,  573,  574,  640,  639
  23,  574,  575,  641,  640
  24,  575,  576,  642,  641
  25,  576,  577,  643,  642
6250, 6252,  478,    8,  237
*Node
   6253, 0.0350000001, 0.0350000001,           0.
*Nset, nset=Punch-1-RefPt_, internal
6253,
*Elset, elset=Punch-1, generate
    1,  6250,     1
*End Instance
**
```

```
*Instance, name=Blank_Shell-1, part=Blank_Shell
 5.20417042793042e-17,           0.,        0.051
*Node
      1, 0.00499999989, 0.0799999982,           0.
      2, 0.00499999989,           0.,           0.
      3, 0.0799999982,           0.,          0.
      4, 0.0799999982, 0.0799999982,           0.
      5, -0.00499999989,           0.,           0.
      6, -0.00499999989, 0.0799999982,           0.
      7, -0.0799999982, 0.0799999982,           0.
      8, -0.0799999982,           0.,           0.
      9, 0.00499999989, 0.0759999976,           0.
     10, 0.00499999989, 0.0719999969,           0.
     11, 0.00499999989, 0.0680000037,           0.
     12, 0.00499999989,  0.064000003,           0.
     13, 0.00499999989, 0.0599999987,           0.
     14, 0.00499999989, 0.0560000017,           0.
     15, 0.00499999989, 0.0520000011,           0.
     16, 0.00499999989, 0.0480000004,           0.
     17, 0.00499999989, 0.0439999998,           0.
     18, 0.00499999989, 0.0399999991,           0.
     19, 0.00499999989, 0.0359999985,           0.
     20, 0.00499999989, 0.0320000015,           0.
     21, 0.00499999989, 0.0280000009,           0.
     22, 0.00499999989, 0.0240000002,           0.
     23, 0.00499999989, 0.0199999996,           0.
     24, 0.00499999989, 0.0160000008,           0.
     25, 0.00499999989, 0.0120000001,           0.
   1155, 0.00249999994, 0.00400000019,           0.
*Element, type=S4R
  1,   1,   9, 187,  94
  2,   9,  10, 188, 187
  3,  10,  11, 189, 188
  4,  11,  12, 190, 189
  5,  12,  13, 191, 190
  6,  13,  14, 192, 191
  7,  14,  15, 193, 192
  8,  15,  16, 194, 193
  9,  16,  17, 195, 194
 10,  17,  18, 196, 195
 11,  18,  19, 197, 196
 12,  19,  20, 198, 197
 13,  20,  21, 199, 198
 14,  21,  22, 200, 199
 15,  22,  23, 201, 200
 16,  23,  24, 202, 201
 17,  24,  25, 203, 202
 18,  25,  26, 204, 203
 19,  26,  27, 205, 204
 20,  27,   2,  28, 205
 21,  94, 187, 206,  93
 22, 187, 188, 207, 206
 23, 188, 189, 208, 207
 24, 189, 190, 209, 208
 25, 190, 191, 210, 209
1080, 1155,  183,    2,   27
*Nset, nset=_PickedSet5, internal
```

```
    1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
 *Elset, elset=_PickedSet5, internal, generate
    1,  500,    1
 *Nset, nset=_PickedSet6, internal
     5,    6,    7,    8,   95,   96,   97,   98,   99,  100,  101,
102,  103,
 *Elset, elset=_PickedSet6, internal, generate
  501, 1000,    1
 *Nset, nset=_PickedSet7, internal
     1,    2,    5,    6,    9,   10,   11,   12,   13,   14,   15,
16,   17,
 *Elset, elset=_PickedSet7, internal, generate
  1001,  1080,     1
 *Nset, nset=_PickedSet8, internal
    1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
 *Elset, elset=_PickedSet8, internal, generate
    1,  500,    1
 *Nset, nset=_PickedSet9, internal
     5,    6,    7,    8,   95,   96,   97,   98,   99,  100,  101,
102,  103,
 *Elset, elset=_PickedSet9, internal, generate
  501, 1000,    1
 *Nset, nset=_PickedSet10, internal
     1,    2,    5,    6,    9,   10,   11,   12,   13,   14,   15,
16,   17,
 *Elset, elset=_PickedSet10, internal, generate
  1001,  1080,     1
 *Nset, nset=_PickedSet11, internal
    1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
 *Elset, elset=_PickedSet11, internal, generate
    1,  500,    1
 *Nset, nset=_PickedSet12, internal
     5,    6,    7,    8,   95,   96,   97,   98,   99,  100,  101,
102,  103,
 *Elset, elset=_PickedSet12, internal, generate
  501, 1000,    1
 *Nset, nset=_PickedSet13, internal
     1,    2,    5,    6,    9,   10,   11,   12,   13,   14,   15,
16,   17,
 *Elset, elset=_PickedSet13, internal, generate
  1001,  1080,     1
 *Nset, nset=_PickedSet14, internal
    1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
 *Elset, elset=_PickedSet14, internal, generate
    1,  500,    1
 *Nset, nset=_PickedSet15, internal
     5,    6,    7,    8,   95,   96,   97,   98,   99,  100,  101,
102,  103,
 *Elset, elset=Left_Blank, generate
  501, 1000,    1
 *Nset, nset=Right_Blank
    1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
```

```
 *Elset, elset=Right_Blank, generate
    1,  500,    1
 *Nset, nset=Weld
     1,    2,    5,    6,    9,   10,   11,   12,   13,   14,   15,
16,   17,   18,   19,   20
    21,   22,   23,   24,   25,   26,   27,   95,   96,   97,   98,
99,  100,  101,  102,  103
   104,  105,  106,  107,  108,  109,  110,  111,  112,  113,  181,
182,  183,  184,  185,  186
  1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1107, 1108, 1109,
1110, 1111, 1112, 1113, 1114
  1115, 1116, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125,
1126, 1127, 1128, 1129, 1130
  1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141,
1142, 1143, 1144, 1145, 1146
  1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155
 *Elset, elset=Weld, generate
  1001,  1080,     1
 *Nset, nset=WELD_LINE, instance=Blank_Shell-1
 182, 185, 1118, 1119, 1120, 1121, 1122, 1123, 1124
 1125, 1126, 1127, 1128, 1129, 1130, 1131, 1132
 1133, 1134, 1135, 1136
 ** Region: (Al 2024_R:Right_Blank), (Controls:EC-1)
 ** Section: Al 2024_R
 *Shell Section, elset=Right_Blank, controls=EC-1, material="Al 2024
T3"
 0.002, 5
 ** Region: (Al 7075 L:Left_Blank), (Controls:EC-1)
 ** Section: Al 7075 L
 *Shell Section, elset=Left_Blank, controls=EC-1, material="Al 7075 T6"
 0.002, 5
 ** Region: (Weld:Weld), (Controls:EC-1)
 ** Section: Weld
 *Shell Section, elset=Weld, controls=EC-1, material=Weld
 0.002, 5
 *End Instance
 **
 *Instance, name=Holder-L-1, part=Holder-L
           0.,           0.,        0.052
           0.,           0.,        0.052,           0.,           0.,
1.052, 89.9999990194245
 *Node
       1,           0.,  0.129999995, 0.0199999996
       2,           0.,  0.129999995,           0.
       3,           0., 0.0450000018, 0.0199999996
       4,           0., 0.0399999991, 0.0149999997
       5,           0., 0.0399999991,           0.
       6,  0.129999995,           0.,           0.
       7,  0.129999995,           0., 0.0199999996
       8, 0.0450000018,           0., 0.0199999996
       9, 0.0399999991,           0., 0.0149999997
      10, 0.0399999991,           0.,           0.
      11, 0.0299999993, 0.0399999991,           0.
      12, 0.0299999993, 0.0399999991, 0.0149999997
      13, 0.0299999993, 0.0450000018, 0.0199999996
      14, 0.0399999991, 0.0299999993,           0.
      15, 0.0399999991, 0.0299999993, 0.0149999997
```

```
     16, 0.0450000018, 0.0299999993, 0.0199999996
     17,           0.,  0.120233029,           0.
     18,           0.,  0.110853069,           0.
     19,           0.,  0.101697981,           0.
     20,           0., 0.0929246172,           0.
     21,           0., 0.0841150135,           0.
     22,           0., 0.0758666545,           0.
     23,           0.,  0.068053633,           0.
     24,           0., 0.0614104383,           0.
     25,           0., 0.0553878546,           0.
    492,   0.12938948, 0.0125842206, 0.0133333327
*Element, type=R3D4
 1,  20, 131, 135,  19
 2, 131, 132, 136, 135
 3, 132,  34,  35, 136
 4,  19, 135, 137,  18
 5, 135, 136, 138, 137
 6, 136,  35,  36, 138
 7,  18, 137, 139,  17
 8, 137, 138, 140, 139
 9, 138,  36,  37, 140
10,  17, 139,  39,   2
11, 139, 140,  38,  39
12, 140,  37,   1,  38
13,  22, 133, 141,  21
14, 133, 134, 142, 141
15, 134,  32,  33, 142
16,  21, 141, 131,  20
17, 141, 142, 132, 131
18, 142,  33,  34, 132
19,  32, 134, 144,  31
20,  31, 144, 145,  30
21,  30, 145, 146,   3
22,   3, 146, 147,  29
23,  29, 147, 143,   4
24, 134, 133, 149, 144
25, 133,  22,  23, 149
490, 492, 117,   7,  52
*Node
    493,  0.091923885,  0.091923885, 0.0199999996
*Nset, nset=Holder-L-1-RefPt_, internal
493,
*Nset, nset=_PickedSet5, internal
 493,
*Elset, elset=Holder-L-1, generate
   1,  490,    1
*Element, type=MASS, elset=_PickedSet5_Holder_MASS_
491, 493
*Mass, elset=_PickedSet5_Holder_MASS_
1.,
*Element, type=ROTARYI, elset=_PickedSet5_Holder_ROTI_
492, 493
*RotaryI, elset=_PickedSet5_Holder_ROTI_
1., 1., 1., 0., 0., 0.
*End Instance
**
*Nset, nset=_PickedSet22, internal, instance=Holder-R-1
```

```
 493,
*Nset, nset=_PickedSet23, internal, instance=Holder-L-1
 493,
*Nset, nset=_PickedSet25, internal, instance=Holder-L-1
 493,
*Nset, nset=_PickedSet26, internal, instance=Holder-R-1
 493,
*Nset, nset=_PickedSet27, internal, instance=Die-1
 17743,
*Nset, nset=_PickedSet31, internal, instance=Holder-L-1
 493,
*Nset, nset=_PickedSet41, internal, instance=Blank_Shell-1
   2,   3,   5,   8,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,
38,  39
*Elset, elset=_PickedSet41, internal, instance=Blank_Shell-1
   20,   40,   60,   80,  100,  120,  140,  160,  180,  200,  220,
240,  260,
*Nset, nset=_PickedSet57, internal, instance=Holder-R-1
 493,
*Nset, nset=_PickedSet58, internal, instance=Holder-L-1
 493,
*Nset, nset=Left_Blank, instance=Blank_Shell-1
    5,    6,    7,    8,   95,   96,   97,   98,   99,  100,  101,
102,  103,
*Elset, elset=Left_Blank, instance=Blank_Shell-1, generate
 501, 1000,    1
*Nset, nset=Right_Blank, instance=Blank_Shell-1
   1,   2,   3,   4,   9,  10,  11,  12,  13,  14,  15,  16,  17,  18,
19,  20
*Elset, elset=Right_Blank, instance=Blank_Shell-1, generate
   1,  500,    1
*Nset, nset=Left_Sym, instance=Blank_Shell-1
   5,   8, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
169, 170
 171, 172, 173, 174, 175, 176, 177, 178, 179, 180
*Elset, elset=Left_Sym, instance=Blank_Shell-1, generate
 501,  981,   20
*Nset, nset=Right_Sym, instance=Blank_Shell-1
  2,  3, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41
 42, 43, 44, 45, 46, 47, 48, 49, 50, 51
*Elset, elset=Right_Sym, instance=Blank_Shell-1, generate
  20,  500,   20
*Nset, nset=Weld, instance=Blank_Shell-1
    1,    2,    5,    6,    9,   10,   11,   12,   13,   14,   15,
16,   17,
*Elset, elset=Weld, instance=Blank_Shell-1, generate
 1001,  1080,     1
*Nset, nset=Punch_RP, instance=Punch-1
 6253,
*Nset, nset=Holder_R_RP, instance=Holder-R-1
 493,
*Nset, nset=Holder_L_RP, instance=Holder-L-1
 493,
*Nset, nset=Die_RP, instance=Die-1
 17743,
*Nset, nset=Weld_Sym, instance=Blank_Shell-1
   2,   5, 181, 182, 183
```

```
 *Elset, elset=Weld_Sym, instance=Blank_Shell-1, generate
  1020,  1080,    20
 *Nset, nset=Blank, instance=Blank_Shell-1, generate
     1,  1155,     1
 *Elset, elset=Blank, instance=Blank_Shell-1, generate
     1,  1080,     1
 *Elset, elset=_Punch_SPOS, internal, instance=Punch-1
  1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794,
1795, 1796,
 *Surface, type=ELEMENT, name=Punch
 _Punch_SPOS, SPOS
 *Elset, elset=_Die_SPOS, internal, instance=Die-1, generate
   2375,  17740,      1
 *Surface, type=ELEMENT, name=Die
 _Die_SPOS, SPOS
 *Elset, elset=_Holder_R_SPOS, internal, instance=Holder-R-1, generate
   91,  270,    1
 *Surface, type=ELEMENT, name=Holder_R
 _Holder_R_SPOS, SPOS
 *Elset, elset=_Holder_L_SPOS, internal, instance=Holder-L-1, generate
   91,  270,    1
 *Surface, type=ELEMENT, name=Holder_L
 _Holder_L_SPOS, SPOS
 *Elset, elset=_Blank_R_Top_SPOS, internal, instance=Blank_Shell-1,
generate
    1,  500,    1
 *Surface, type=ELEMENT, name=Blank_R_Top
 _Blank_R_Top_SPOS, SPOS
 *Elset, elset=_Blank_R_Bot_SNEG, internal, instance=Blank_Shell-1,
generate
    1,  500,    1
 *Surface, type=ELEMENT, name=Blank_R_Bot
 _Blank_R_Bot_SNEG, SNEG
 *Elset, elset=_Blank_L_Bot_SNEG, internal, instance=Blank_Shell-1,
generate
  501, 1000,    1
 *Surface, type=ELEMENT, name=Blank_L_Bot
 _Blank_L_Bot_SNEG, SNEG
 *Elset, elset=_Blank_L_Top_SPOS, internal, instance=Blank_Shell-1,
generate
  501, 1000,    1
 *Surface, type=ELEMENT, name=Blank_L_Top
 _Blank_L_Top_SPOS, SPOS
 *Elset, elset=_Weld_Top_SPOS, internal, instance=Blank_Shell-1,
generate
  1001,  1080,     1
 *Surface, type=ELEMENT, name=Weld_Top
 _Weld_Top_SPOS, SPOS
 *Elset, elset=_Weld_Bot_SNEG, internal, instance=Blank_Shell-1,
generate
  1001,  1080,     1
 *Surface, type=ELEMENT, name=Weld_Bot
 _Weld_Bot_SNEG, SNEG
 *Rigid Body, ref node=Die-1.Die-1-RefPt_, elset=Die-1.Die-1
 *Rigid Body, ref node=Holder-R-1.Holder-R-1-RefPt_, elset=Holder-R-
1.Holder-R-1
 *Rigid Body, ref node=Punch-1.Punch-1-RefPt_, elset=Punch-1.Punch-1
```

```
 *Rigid Body, ref node=Holder-L-1.Holder-L-1-RefPt_, elset=Holder-L-
1.Holder-L-1
 *End Assembly
 **
 ** ELEMENT CONTROLS
 **
 *Section Controls, name=EC-1, hourglass=ENHANCED
 1., 1., 1.
 *Amplitude, name=Holder_Disp
 0., 0., 0.01, 1.
 *Amplitude, name=Holder_Force
 0., 1., 1., 1.
 *Amplitude, name=Punch
 0., 1., 1., 1.
 **
 ** MATERIALS
 **
 *Material, name="Al 2024 T3"
 *Density
 2780.,
 *Elastic
  7.31e+10, 0.33
 *Plastic
  3.45e+08,0.
  4.83e+08,1.
 *Material, name="Al 7075 T6"
 *Density
 2810.,
 *Elastic
  7.17e+10, 0.33
 *Plastic
  5.03e+08,0.
  5.72e+08,1.
 *Material, name=Weld
 *Density
 2800.,
 *Elastic
  7.22e+10, 0.33
 *Plastic
  3.25e+08,0.
  4.16e+08,1.
 **
 ** INTERACTION PROPERTIES
 **
 *Surface Interaction, name=Friction
 *Friction
 0.057,
**
** BOUNDARY CONDITIONS
**
** Name: Die Type: Displacement/Rotation
*Boundary
Die_RP, 1, 1
Die_RP, 2, 2
Die_RP, 3, 3
Die_RP, 4, 4
Die_RP, 5, 5
```

```
Die_RP, 6, 6
** Name: Holder_L_BC Type: Displacement/Rotation
*Boundary
Holder_L_RP, 1, 1
Holder_L_RP, 2, 2
Holder_L_RP, 4, 4
Holder_L_RP, 5, 5
Holder_L_RP, 6, 6
** Name: Holder_R_BC Type: Displacement/Rotation
*Boundary
Holder_R_RP, 1, 1
Holder_R_RP, 2, 2
Holder_R_RP, 4, 4
Holder_R_RP, 5, 5
Holder_R_RP, 6, 6
** Name: Left_Blank_Symm Type: Symmetry/Antisymmetry/Encastre
*Boundary
Left_Sym, YSYMM
** Name: Punch_BC Type: Displacement/Rotation
*Boundary
Punch_RP, 1, 1
Punch_RP, 2, 2
Punch_RP, 4, 4
Punch_RP, 5, 5
Punch_RP, 6, 6
** Name: Right_Blank_Symm Type: Symmetry/Antisymmetry/Encastre
*Boundary
Right_Sym, YSYMM
** Name: Weld_Symm Type: Symmetry/Antisymmetry/Encastre
*Boundary
Weld_Sym, YSYMM
** ----------------------------------------------------------------
**
** STEP: Punch
**
*Step, name=Punch
Apply Velocity for the Punch
*Dynamic, Explicit
, 0.012
*Bulk Viscosity
0.06, 1.2
**
** BOUNDARY CONDITIONS
**
** Name: Punch_Vel Type: Velocity/Angular velocity
*Boundary, amplitude=Punch, type=VELOCITY
Punch_RP, 3, 3, -1.
**
** LOADS
**
** Name: Holder_L_Force   Type: Concentrated force
*Cload, amplitude=Punch
Holder_L_RP, 2, 0.
Holder_L_RP, 3, - 7.783E3
** Name: Holder_R_Force   Type: Concentrated force
*Cload, amplitude=Punch
Holder_R_RP, 2, 0.
```

```
Holder_R_RP, 3, -5.673E3
**
** INTERACTIONS
**
** Interaction: Die_L
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Die_L
Die, Blank_L_Bot
** Interaction: Die_R
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Die_R
Die, Blank_R_Bot
** Interaction: Holder_L
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Holder_L
Holder_L, Blank_L_Top
** Interaction: Holder_R
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Holder_R
Holder_R, Blank_R_Top
** Interaction: Punch_L
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Punch_L
Punch, Blank_L_Top
** Interaction: Punch_R
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Punch_R
Punch, Blank_R_Top
** Interaction: Punch_Weld
*Contact Pair, interaction=Friction, mechanical constraint=KINEMATIC,
weight=1., cpset=Punch_Weld
Punch, Weld_Top
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=2, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
*Node Output
A, AR, CVOL, RF, RM, RT, U, UR
UT, V
*Element Output, directions=YES
E, ER, EVOL, IVOL, LE, MISESMAX, NE, PE, PEEQ, PEEQMAX, PEEQT, PEMAG,
PEQC, S, STH
*FILE OUTPUT, NUMBER INTERVAL=10
*NODE FILE
U,
*EL FILE
STH,
*Output, history, frequency=0
*End Step
```

Appendix B

(Software specifications)

The finite element analysis for friction stir welding, stretch forming and deep drawing process of square pan model was conducted using the finite element method software ABAQUS developed by Hibbitt, Karlsson & Sorensen, Inc. (HKS).

In this research, ABAQUS was accessed through NCSA (National Center for Supercomputing Applications). The finite element analysis was conducted using NCSA Xeon Linux Cluster. The specifications of the computer using for finite element analysis are

Processor type : Intel Xeon 3.2 GHz

Number of processors : 64

Memory : 16 GB

Scratch disk : 1290 GB

The CPU time for finite element analysis of each case is

Friction Stir Welding: 20 to 22 hours Approx

Forming process : 120 minutes

Springback analysis : 2 minutes

For the optimization using Genetic Algorithm, analysis was carried out for 20 generations in each case (5 individuals in each generation). Hence the total CPU time was about 40 hours.

Appendix C
(GA Input files)

```
C     Ga-1
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax)
      dimension nposibl(nparmax),nichflg(nparmax)
      dimension iparent(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension paramsm(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga8   / nichflg
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Input variable definitions:
c
c  icreep   = 0 for no creep mutations
c           = 1 for creep mutations; creep mutations are recommended.
c  idum     The initial random number seed for the GA run.  Must equal
c           a negative integer, e.g. idum=-1000.
c  ielite   = 0 for no elitism (best individual not necessarily
c              replicated from one generation to the next).
c           = 1 for elitism to be invoked (best individual replicated
c              into next generation); elitism is recommended.
c  iend     = 0 for normal GA run (this is standard).
c           = number of last population member to be looked at in a set
c             of individuals.  Setting iend-0 is only used for
debugging
c             purposes and is commonly used in conjunction with iskip.
c  iniche   = 0 for no niching
c           = 1 for niching; niching is recommended.
c  irestrt  = 0 for a new GA run, or for a single function evaluation
c           = 1 for a restart continuation of a GA run.
c  iskip    = 0 for normal GA run (this is standard).
c           = number in population to look at a specific individual or
c             set of individuals.  Setting iskip-0 is only used for
c             debugging purposes.
c  itourny  No longer used.  The GA is presently set up for only
c           tournament selection.
c  iunifrm  = 0 for single-point crossover
c           = 1 for uniform crossover; uniform crossover is
recommended.
c  kountmx  = the maximum value of kount before a new restart file is
```

```
c             written; presently set to write every fifth generation.
c             Increasing this value will reduce I/O time requirements
c             and reduce wear and tear on your storage device
c  maxgen   The maximum number of generations to run by the GA.
c           For a single function evaluation, set equal to 1.
c  microga  = 0 for normal conventional GA operation
c           = 1 for micro-GA operation (this will automatically reset
c             some of the other input flags).  I recommend using
c             npopsiz=5 when microga=1.
c  nchild   = 1 for one child per pair of parents (this is what I
c               typically use).
c           = 2 for two children per pair of parents (2 is more common
c               in GA work).
c  nichflg  = array of 1/0 flags for whether or not niching occurs on
c             a particular parameter.  Set to 0 for no niching on
c             a parameter, set to 1 for niching to operate on
parameter.
c             The default value is 1, but the implementation of niching
c             is still controlled by the flag iniche.
c  nowrite  = 0 to write detailed mutation and parameter adjustments
c           = 1 to not write detailed mutation and parameter
adjustments
c  nparam   Number of parameters (groups of bits) of each individual.
c           Make sure that nparam matches the number of values in the
c           parmin, parmax and nposibl input arrays.
c  npopsiz  The population size of a GA run (typically 100 works well).
c           For a single calculation, set equal to 1.
c  nposibl  = array of integer number of possibilities per parameter.
c             For optimal code efficiency set nposibl=2**n, i.e. 2, 4,
c             8, 16, 32, 64, etc.
c  parmax   = array of the maximum allowed values of the parameters
c  parmin   = array of the minimum allowed values of the parameters
c  pcreep   The creep mutation probability.  Typically set this
c           = (nchrome/nparam)/npopsiz.
c  pcross   The crossover probability.  For single-point crossover, a
c           value of 0.6 or 0.7 is recommended.  For uniform crossover,
c           a value of 0.5 is suggested.
c  pmutate  The jump mutation probability.  Typically set = 1/npopsiz.
c
c
c  For single function evaluations, set npopsiz=1, maxgen=1, &
irestrt=0.
c
c  My favorite initial choices of GA parameters are:
c     microga=1, npopsiz=5, inunifrm=1, maxgen=100
c     microga=1, npopsiz=5, inunifrm=0, maxgen=100
c  I generally get good performance with both the uniform and single-
c  point crossover micro-GA.
c
c  For those wishing to use the more conventional GA techniques,
c  my old favorite choice of GA parameters was:
c     iunifrm=1, iniche=1, ielite=1, itourny=1, nchild=1
c  For most problems I have dealt with, I get good performance using
c     npopsiz=100, pcross=0.5, pmutate=0.01, pcreep=0.02, maxgen=26
c  or
c     npopsiz= 50, pcross=0.5, pmutate=0.02, pcreep=0.04, maxgen=51
c
```

```
c  Any negative integer for idum should work.  I typically arbitrarily
c  choose idum=-10000 or -20000.
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Code variable definitions (those not defined above):
c
c  best     = the best fitness of the generation
c  child    = the floating point parameter array of the children
c  cpu      = cpu time of the calculation
c  creep    = +1 or -1, indicates which direction parameter creeps
c  delta    = del/nparam
c  diffrac  = fraction of total number of bits which are different
c             between the best and the rest of the micro-GA population.
c             Population convergence arbitrarily set as diffrac<0.05.
c  fbar     = average fitness of population
c  fitness  = array of fitnesses of the parents
c  fitsum   = sum of the fitnesses of the parents
c  g0       = lower bound values of the parameter array to be
optimized.
c             The number of parameters in the array should match the
c             dimension set in the above parameter statement.
c  g1       = the increment by which the parameter array is increased
c             from the lower bound values in the g0 array.  The minimum
c             parameter value is g0 and the maximum parameter value
c             equals g0+g1*(2**g2-1), i.e. g1 is the incremental value
c             between min and max.
c  ig2      = array of the number of bits per parameter, i.e. the
number
c             of possible values per parameter.  For example, ig2=2 is
c             equivalent to 4 (=2**2) possibilities, ig2=4 is
equivalent
c             to 16 (=2**4) possibilities.
c  ig2sum   = sum of the number of possibilities of ig2 array
c  ibest    = binary array of chromosomes of the best individual
c  ichild   = binary array of chromosomes of the children
c  icount   = counter of number of different bits between best
c             individual and other members of micro-GA population
c  icross   = the crossover point in single-point crossover
c  indmax   = maximum # of individuals allowed, i.e. max population
size
c  iparent  = binary array of chromosomes of the parents
c  istart   = the generation to be started from
c  jbest    = the member in the population with the best fitness
c  jelite   = a counter which tracks the number of bits of an
individual
c             which match those of the best individual
c  jend     = used in conjunction with iend for debugging
c  jstart   = used in conjunction with iskip for debugging
c  kount    = a counter which controls how frequently the restart
c             file is written
c  kelite   = kelite set to unity when jelite=nchrome, indicates that
c             the best parent was replicated amongst the children
c  mate1    = the number of the population member chosen as mate1
c  mate2    = the number of the population member chosen as mate2
c  nchrmax  = maximum # of chromosomes (binary bits) per individual
```

```
c  nchrome  = number of chromosomes (binary bits) of each individual
c  ncreep   = # of creep mutations which occurred during reproduction
c  nmutate  = # of jump mutations which occurred during reproduction
c  nparmax  = maximum # of parameters which the chromosomes make up
c  paramav  = the average of each parameter in the population
c  paramsm  = the sum of each parameter in the population
c  parent   = the floating point parameter array of the parents
c  pardel   = array of the difference between parmax and parmin
c  rand     = the value of the current random number
c  npossum  = sum of the number of possible values of all parameters
c  time0    = clock time at start of run
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Subroutines:
c  ____________
c
c  code     = Codes floating point value to binary string.
c  crosovr  = Performs crossover (single-point or uniform).
c  decode   = Decodes binary string to floating point value.
c  evalout  = Evaluates the fitness of each individual and outputs
c             generational information to the 'ga.out' file.
c  func     = The function which is being evaluated.
c  gamicro  = Implements the micro-GA technique.
c  input    = Inputs information from the 'ga.inp' file.
c  initial  = Program initialization and inputs information from the
c             'ga.restart' file.
c  mutate   = Performs mutation (jump and/or creep).
c  newgen   = Writes child array back into parent array for new
c             generation; also checks to see if best individual was
c             replicated (elitism).
c  niche    = Performs niching (sharing) on population.
c  possibl  = Checks to see if decoded binary string falls within
c             specified range of parmin and parmax.
c  ran3     = The random number generator.
c  restart  = Writes the 'ga.restart' file.
c  select   = A subroutine of 'selectn'.
c  selectn  = Performs selection; tournament selection is the only
c             option in this version of the code.
c  shuffle  = Shuffles the population randomly for selection.
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Call the input subroutine.
c      TIME0=SECNDS(0.0)
       call input
c
c  Perform necessary initialization and read the ga.restart file.
       call initial(istart,npossum,ig2sum)

       OPEN (UNIT=22, FILE='ga-a1.dat', STATUS='unknown')
       write (22,*) istart,nchrome,npossum,ig2sum
c
c  Evaluate the population, assign fitness, establish the best
c  individual, and write output information.
```

```
         call evalout_a(iskip,iend,ibest)
         if(npopsiz.eq.1 .or. iskip.ne.0) then
            close(24)
            stop
         endif
c
 1050 format(1x,' #      Binary Code',16x,'Param1  Param2  Fitness')
 1111 format(//'#################  Generation',i5,'
##################')
 1225 format(/'  Number of Crossovers      =',i5)
c 1400 format(2x,'CPU time for generation=',e10.4)
c
      stop
      end
c
c#############################################################################
#
      subroutine input
c
c  This subroutine inputs information from the ga.inp file.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension nposibl(nparmax),nichflg(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga8   / nichflg
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx
c
      namelist / ga   / irestrt,npopsiz,pmutate,maxgen,idum,pcross,
     +                  itourny,ielite,icreep,pcreep,iunifrm,iniche,
     +                  iskip,iend,nchild,nparam,parmin,parmax,nposibl,
     +                  nowrite,nichflg,microga,kountmx
c
        write(*,*) 'input'
      kountmx=5
      irestrt=0
      itourny=0
      ielite=0
      iunifrm=0
      iniche=0
      iskip=0
      iend=0
      nchild=1
      do 2 i=1,nparam
         nichflg(i)=1
 2    continue
      microga=0
c
      OPEN (UNIT=24, FILE='ga-a.out', STATUS='UNKNOWN')
```

```
      rewind 24
      OPEN (UNIT=23, FILE='ga.inp', STATUS='OLD')
      READ (23, NML = ga)
      CLOSE (23)
      itourny=1
c      if (itourny.eq.0) nchild=2
c
c  Check for array sizing errors.
      if (npopsiz.gt.indmax) then
         write(6,1600) npopsiz
         write(24,1600) npopsiz
         close(24)
         stop
      endif
      if (nparam.gt.nparmax) then
         write(6,1700) nparam
         write(24,1700) nparam
         close(24)
         stop
      endif
c
c  If using the microga option, reset some input variables
      if (microga.ne.0) then
         pmutate=0.0
         pcreep=0.0
         itourny=1
         ielite=1
         iniche=0
         nchild=1
         if (iunifrm.eq.0) then
            pcross=1.0
         else
            pcross=0.5
         endif
      endif
c
 1600 format(1x,'ERROR: npopsiz > indmax.  Set indmax = ',i6)
 1700 format(1x,'ERROR: nparam > nparmax.  Set nparmax = ',i6)
c
      return
      end
c
c######################################################################
#
      subroutine initial(istart,npossum,ig2sum)
c
c  This subroutine sets up the program by generating the g0, g1 and
c  ig2 arrays, and counting the number of chromosomes required for the
c  specified input.  The subroutine also initializes the random number
c  generator, parent and iparent arrays (reads the ga.restart file).
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension nposibl(nparmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
```

```
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx
c
c
        write(*,*) 'initial'
      do 3 i=1,nparam
         g0(i)=parmin(i)
         pardel(i)=parmax(i)-parmin(i)
         g1(i)=pardel(i)/dble(nposibl(i)-1)
 3    continue
      do 6 i=1,nparam
         do 7 j=1,30
            n2j=2**j
            if (n2j.ge.nposibl(i)) then
               ig2(i)=j
               goto 8
            endif
            if (j.ge.30) then
               write(6,2000)
               write(24,2000)
               close(24)
               stop
            endif
 7       continue
 8       continue
 6    continue
c
c  Count the total number of chromosomes (bits) required
      nchrome=0
      npossum=0
      ig2sum=0
      do 9 i=1,nparam
         nchrome=nchrome+ig2(i)
         npossum=npossum+nposibl(i)
         ig2sum=ig2sum+(2**ig2(i))
 9    continue
      if (nchrome.gt.nchrmax) then
         write(6,1800) nchrome
         write(24,1800) nchrome
         close(24)
         stop
      endif
c
      if (npossum.lt.ig2sum .and. microga.ne.0) then
         write(6,2100)
         write(24,2100)
      endif
c
c  Initialize random number generator
```

```
      call ran3(idum,rand)
c
      if(irestrt.eq.0) then
c  Initialize the random distribution of parameters in the individual
c  parents when irestrt=0.
         istart=1
         do 10 i=1,npopsiz
            do 15 j=1,nchrome
               call ran3(1,rand)
               iparent(i,j)=1
               if(rand.lt.0.5) iparent(i,j)=0
 15         continue
 10      continue
         if (npossum.lt.ig2sum) call possibl(parent,iparent)
      else
c  If irestrt.ne.0, read from restart file.
         OPEN (UNIT=25, FILE='ga.restart', STATUS='OLD')
         rewind 25
         read(25,*) istart,npopsiz
         do 1 j=1,npopsiz
            read(25,*) k,(iparent(j,l),l=1,nchrome)
 1       continue
         CLOSE (25)
      endif
c
      if(irestrt.ne.0) call ran3(idum-istart,rand)
c
 1800 format(1x,'ERROR: nchrome > nchrmax.  Set nchrmax = ',i6)
 2000 format(1x,'ERROR: You have a parameter with a number of '/
     +       1x,'   possibilities > 2**30!  If you really desire
this,'/
     +       1x,'   change the DO loop 7 statement and recompile.'//
     +       1x,'   You may also need to alter the code to work with'/
     +       1x,'   REAL numbers rather than INTEGER numbers; Fortran'/
     +       1x,'   does not like to compute 2**j when j>30.')
 2100 format(1x,'WARNING: for some cases, a considerable performance'/
     +       1x,'   reduction has been observed when running a non-'/
     +       1x,'   optimal number of bits with the micro-GA.'/
     +       1x,'   If possible, use values for nposibl of 2**n,'/
     +       1x,'   e.g. 2, 4, 8, 16, 32, 64, etc.  See ReadMe file.')
c
      return
      end

c
c######################################################################
#
      subroutine ran3(idum,rand)
c
c  Returns a uniform random deviate between 0.0 and 1.0.  Set idum to
c  any negative value to initialize or reinitialize the sequence.
c  This function is taken from W.H. Press', "Numerical Recipes" p. 199.
c
      implicit double precision (a-h,m,o-z)
      save
c      implicit real*4(m)
      parameter (mbig=6000000.,mseed=1618033.,mz=0.,fac=1./mbig)
```

```
c     parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1./mbig)
c
c  According to Knuth, any large mbig, and any smaller (but still
large)
c  mseed can be substituted for the above values.
      dimension ma(55)
      data iff /0/

        write(*,*) 'ran3'
      if (idum.lt.0 .or. iff.eq.0) then
         iff=1
         mj=mseed-dble(iabs(idum))
         mj=dmod(mj,mbig)
         ma(55)=mj
         mk=1

         do 11 i=1,54
            ii=mod(21*i,55)
            ma(ii)=mk
            mk=mj-mk
            if(mk.lt.mz) mk=mk+mbig
            mj=ma(ii)
 11      continue

         do 13 k=1,4
            do 12 i=1,55
               ma(i)=ma(i)-ma(1+mod(i+30,55))
               if(ma(i).lt.mz) ma(i)=ma(i)+mbig
 12         continue
 13      continue

         inext=0
         inextp=31
         idum=1
      endif

      inext=inext+1
      if(inext.eq.56) inext=1
      inextp=inextp+1
      if(inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
      rand=mj*fac
      return
      end

c
c#####################################################################
#
c
      subroutine possibl(array,iarray)
c
c  This subroutine determines whether or not all parameters are within
c  the specified range of possibility.  If not, the parameter is
c  randomly reassigned within the range.  This subroutine is only
c  necessary when the number of possibilities per parameter is not
```

```
c  optimized to be 2**n, i.e. if npossum < ig2sum.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax),nposibl(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
        write(*,*) 'possibl'
      do 10 i=1,npopsiz
         call decode(i,array,iarray)
         do 20 j=1,nparam
            n2ig2j=2**ig2(j)
            if(nposibl(j).ne.n2ig2j .and. array(i,j).gt.parmax(j)) then
               call ran3(1,rand)
               irand=dint(dble(nposibl(j))*rand)
               array(i,j)=g0(j)+dble(irand)*g1(j)
               call code(i,j,array,iarray)
               if (nowrite.eq.0) write(6,1000) i,j
               if (nowrite.eq.0) write(24,1000) i,j
            endif
 20      continue
 10   continue
c
 1000 format('*** Parameter adjustment to individual     ',i4,
     +       ', parameter  ',i3,' ***')
c
      return
      end

c
c#####################################################################
#
      subroutine decode(i,array,iarray)
c
c  This routine decodes a binary string to a real number.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
c
        write(*,*) 'decode'
      l=1
      do 10 k=1,nparam
         iparam=0
         m=l
```

```
         do 20 j=m,m+ig2(k)-1
            l=l+1
            iparam=iparam+iarray(i,j)*(2**(m+ig2(k)-1-j))
 20      continue
         array(i,k)=g0(k)+g1(k)*dble(iparam)
 10   continue
c
      return
      end
c
c#########################################################################
#
      subroutine code(j,k,array,iarray)
c
c  This routine codes a parameter into a binary string.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
        write(*,*) 'code'
c
c  First, establish the beginning location of the parameter string of
c  interest.
      istart=1

      do 10 i=1,k-1
         istart=istart+ig2(i)
 10   continue
c
c  Find the equivalent coded parameter value, and back out the binary
c  string by factors of two.
      m=ig2(k)-1
      if (g1(k).eq.0.0) return
      iparam=nint((array(j,k)-g0(k))/g1(k))

      do 20 i=istart,istart+ig2(k)-1
         iarray(j,i)=0
         if ((iparam+1).gt.(2**m)) then
            iarray(j,i)=1
            iparam=iparam-2**m
         endif
         m=m-1
 20   continue

c     write(3,*)array(j,k),iparam,(iarray(j,i),i=istart,istart+ig2(k)-
1)
c
      return
      end

c
```

```
c#########################################################################
#
      subroutine evalout_a(iskip,iend,ibest)
c
c  This subroutine evaluates the population, assigns fitness,
c  establishes the best individual, and outputs information.
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension paramsm(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
c
        write(*,*) 'evalout_a'
      do 29 n=1,nparam
         paramsm(n)=0.0
 29   continue

      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend

      do 30 j=jstart,jend
         call decode(j,parent,iparent)
         if(iskip.ne.0 .and. iend.ne.0 .and. iskip.eq.iend)
     +   write(6,1074) j,(iparent(j,k),k=1,nchrome),
     +                   (parent(j,kk),kk=1,nparam),0.0
         write(24,1075) j,(iparent(j,k),k=1,nchrome),
     +                   (parent(j,kk),kk=1,nparam)
 30   continue
c
 1074 format(i3,1x,26i1,1x,f7.5,1x,f5.3,1x,f6.3)
 1075 format(i3,1x,33i1,1x,f6.3,1x,f5.3,1x,f5.3)
      return
      end
```

```
C     Ga-2
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension fitness(indmax),nposibl(nparmax),nichflg(nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga7   / child,ichild
      common / ga8   / nichflg
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Input variable definitions:
c
c  icreep   = 0 for no creep mutations
c         = 1 for creep mutations; creep mutations are recommended.
c  idum     The initial random number seed for the GA run.  Must equal
c         a negative integer, e.g. idum=-1000.
c  ielite   = 0 for no elitism (best individual not necessarily
c           replicated from one generation to the next).
c         = 1 for elitism to be invoked (best individual replicated
c           into next generation); elitism is recommended.
c  iend     = 0 for normal GA run (this is standard).
c         = number of last population member to be looked at in a set
c           of individuals.  Setting iend-0 is only used for debugging
c           purposes and is commonly used in conjunction with iskip.
c  iniche   = 0 for no niching
c         = 1 for niching; niching is recommended.
c  irestrt  = 0 for a new GA run, or for a single function evaluation
c         = 1 for a restart continuation of a GA run.
c  iskip    = 0 for normal GA run (this is standard).
c         = number in population to look at a specific individual or
c           set of individuals.  Setting iskip-0 is only used for
c           debugging purposes.
c  itourny  No longer used.  The GA is presently set up for only
c         tournament selection.
c  iunifrm  = 0 for single-point crossover
c         = 1 for uniform crossover; uniform crossover is recommended.
c  kountmx  = the maximum value of kount before a new restart file is
c           written; presently set to write every fifth generation.
c           Increasing this value will reduce I/O time requirements
```

```
c           and reduce wear and tear on your storage device
c  maxgen   The maximum number of generations to run by the GA.
c         For a single function evaluation, set equal to 1.
c  microga  = 0 for normal conventional GA operation
c         = 1 for micro-GA operation (this will automatically reset
c           some of the other input flags).  I recommend using
c           npopsiz=5 when microga=1.
c  nchild   = 1 for one child per pair of parents (this is what I
c           typically use).
c         = 2 for two children per pair of parents (2 is more common
c           in GA work).
c  nichflg  = array of 1/0 flags for whether or not niching occurs on
c           a particular parameter.  Set to 0 for no niching on
c           a parameter, set to 1 for niching to operate on parameter.
c           The default value is 1, but the implementation of niching
c           is still controlled by the flag iniche.
c  nowrite  = 0 to write detailed mutation and parameter adjustments
c         = 1 to not write detailed mutation and parameter adjustments
c  nparam   Number of parameters (groups of bits) of each individual.
c         Make sure that nparam matches the number of values in the
c         parmin, parmax and nposibl input arrays.
c  npopsiz  The population size of a GA run (typically 100 works well).
c         For a single calculation, set equal to 1.
c  nposibl  = array of integer number of possibilities per parameter.
c           For optimal code efficiency set nposibl=2**n, i.e. 2, 4,
c           8, 16, 32, 64, etc.
c  parmax   = array of the maximum allowed values of the parameters
c  parmin   = array of the minimum allowed values of the parameters
c  pcreep   The creep mutation probability.  Typically set this
c         = (nchrome/nparam)/npopsiz.
c  pcross   The crossover probability.    For single-point crossover, a
c         value of 0.6 or 0.7 is recommended.  For uniform crossover,
c         a value of 0.5 is suggested.
c  pmutate  The jump mutation probability.  Typically set = 1/npopsiz.
c
c
c  For single function evaluations, set npopsiz=1, maxgen=1, &
irestrt=0.
c
c  My favorite initial choices of GA parameters are:
c     microga=1, npopsiz=5, inunifrm=1, maxgen=100
c     microga=1, npopsiz=5, inunifrm=0, maxgen=100
c  I generally get good performance with both the uniform and single-
c  point crossover micro-GA.
c
c  For those wishing to use the more conventional GA techniques,
c  my old favorite choice of GA parameters was:
c     iunifrm=1, iniche=1, ielite=1, itourny=1, nchild=1
c  For most problems I have dealt with, I get good performance using
c     npopsiz=100, pcross=0.5, pmutate=0.01, pcreep=0.02, maxgen=26
c  or
c     npopsiz= 50, pcross=0.5, pmutate=0.02, pcreep=0.04, maxgen=51
c
c  Any negative integer for idum should work.  I typically arbitrarily
c  choose idum=-10000 or -20000.
c
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Code variable definitions (those not defined above):
c
c  best     = the best fitness of the generation
c  child    = the floating point parameter array of the children
c  cpu          = cpu time of the calculation
c  creep    = +1 or -1, indicates which direction parameter creeps
c  delta    = del/nparam
c  diffrac  = fraction of total number of bits which are different
c           between the best and the rest of the micro-GA population.
c           Population convergence arbitrarily set as diffrac<0.05.
c  fbar     = average fitness of population
c  fitness  = array of fitnesses of the parents
c  fitsum   = sum of the fitnesses of the parents
c  g0     = lower bound values of the parameter array to be optimized.
c           The number of parameters in the array should match the
c           dimension set in the above parameter statement.
c  g1     = the increment by which the parameter array is increased
c           from the lower bound values in the g0 array.  The minimum
c           parameter value is g0 and the maximum parameter value
c           equals g0+g1*(2**g2-1), i.e. g1 is the incremental value
c           between min and max.
c  ig2          = array of the number of bits per parameter, i.e. the
number
c           of possible values per parameter.  For example, ig2=2 is
c           equivalent to 4 (=2**2) possibilities, ig2=4 is equivalent
c           to 16 (=2**4) possibilities.
c  ig2sum   = sum of the number of possibilities of ig2 array
c  ibest    = binary array of chromosomes of the best individual
c  ichild   = binary array of chromosomes of the children
c  icount   = counter of number of different bits between best
c           individual and other members of micro-GA population
c  icross   = the crossover point in single-point crossover
c  indmax   = maximum # of individuals allowed, i.e. max population
size
c  iparent  = binary array of chromosomes of the parents
c  istart   = the generation to be started from
c  jbest    = the member in the population with the best fitness
c  jelite   = a counter which tracks the number of bits of an
individual
c           which match those of the best individual
c  jend     = used in conjunction with iend for debugging
c  jstart   = used in conjunction with iskip for debugging
c  kount    = a counter which controls how frequently the restart
c           file is written
c  kelite   = kelite set to unity when jelite=nchrome, indicates that
c           the best parent was replicated amongst the children
c  mate1    = the number of the population member chosen as mate1
c  mate2    = the number of the population member chosen as mate2
c  nchrmax  = maximum # of chromosomes (binary bits) per individual
c  nchrome  = number of chromosomes (binary bits) of each individual
c  ncreep   = # of creep mutations which occurred during reproduction
c  nmutate  = # of jump mutations which occurred during reproduction
c  nparmax  = maximum # of parameters which the chromosomes make up
c  paramav  = the average of each parameter in the population
```

```
c  paramsm  = the sum of each parameter in the population
c  parent   = the floating point parameter array of the parents
c  pardel   = array of the difference between parmax and parmin
c  rand     = the value of the current random number
c  npossum  = sum of the number of possible values of all parameters
c  time0    = clock time at start of run
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Subroutines:
c  ____________
c
c  code     = Codes floating point value to binary string.
c  crosovr  = Performs crossover (single-point or uniform).
c  decode   = Decodes binary string to floating point value.
c  evalout  = Evaluates the fitness of each individual and outputs
c           generational information to the 'ga.out' file.
c  func     = The function which is being evaluated.
c  gamicro  = Implements the micro-GA technique.
c  input    = Inputs information from the 'ga.inp' file.
c  initial  = Program initialization and inputs information from the
c           'ga.restart' file.
c  mutate   = Performs mutation (jump and/or creep).
c  newgen   = Writes child array back into parent array for new
c           generation; also checks to see if best individual was
c           replicated (elitism).
c  niche    = Performs niching (sharing) on population.
c  possibl  = Checks to see if decoded binary string falls within
c           specified range of parmin and parmax.
c  ran3     = The random number generator.
c  restart  = Writes the 'ga.restart' file.
c  select   = A subroutine of 'selectn'.
c  selectn  = Performs selection; tournament selection is the only
c           option in this version of the code.
c  shuffle  = Shuffles the population randomly for selection.
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c  Call the input subroutine.
c      TIME0=SECNDS(0.0)
       call input

       OPEN (UNIT=22, FILE='ga-a1.dat', STATUS='old')
       read (22,*) istart,nchrome,npossum,ig2sum

c      OPEN (UNIT=24, FILE='ga-aa.out', STATUS='UNKNOWN')

       i = 1

c  Evaluate the population, assign fitness, establish the best
c  individual, and write output information.

        call evalout_b(iskip,iend,ibest)

        if(npopsiz.eq.1 .or. iskip.ne.0) then
```

```
         close(24)
         stop
       endif
c
c  Implement "niching".
       if (iniche.ne.0) call niche
c
c  Enter selection, crossover and mutation loop.
       ncross=0
       ipick=npopsiz

       do 45 j=1,npopsiz,nchild
c
c  Perform selection.
         call selectn(ipick,j,mate1,mate2)
c
c  Now perform crossover between the randomly selected pair.
         call crosovr(ncross,j,mate1,mate2)

 45    continue

       write(6,1225) ncross
       write(24,1225) ncross
c
c  Now perform random mutations.  If running micro-GA, skip mutation.
       if (microga.eq.0) call mutate
c
c  Write child array back into parent array for new generation.  Check
c  to see if the best parent was replicated.
       call newgen(ielite,npossum,ig2sum,ibest)
c
c  Implement micro-GA if enabled.
       if (microga.ne.0) call gamicro(i,npossum,ig2sum,ibest)
c
c  Write output for new generation
       call newgen_opt(iskip,iend,ibest)
c
c  Write to restart file.
       call restart(i,istart,kount)

c  $$$$$ End of main generational processing loop. $$$$$
c
      CLOSE (24)
c
c 1050 format(1x,' #      Binary Code',8x,'Param1  Param2  Fitness')
 1111 format(//'#################  Generation',i5,'
#################')
 1225 format(/'  Number of Crossovers      =',i5)
c 1400 format(2x,'CPU time for generation=',e10.4)
c
      stop
      end
c
c#########################################################################
#
      subroutine input
c
```

```
c  This subroutine inputs information from the ga.inp file.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension nposibl(nparmax),nichflg(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga8   / nichflg
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx
c
      namelist / ga   / irestrt,npopsiz,pmutate,maxgen,idum,pcross,
     +                  itourny,ielite,icreep,pcreep,iunifrm,iniche,
     +                  iskip,iend,nchild,nparam,parmin,parmax,nposibl,
     +                  nowrite,nichflg,microga,kountmx
c
      write(*,*) 'input'
      kountmx=5
      irestrt=0
      itourny=0
      ielite=0
      iunifrm=0
      iniche=0
      iskip=0
      iend=0
      nchild=1
      do 2 i=1,nparam
       nichflg(i)=1
 2    continue
      microga=0
c
      OPEN (UNIT=25, FILE='ga.out', STATUS='UNKNOWN')
      rewind 24
      OPEN (UNIT=23, FILE='ga.inp', STATUS='OLD')
      READ (23, NML = ga)
      CLOSE (23)
      itourny=1
c      if (itourny.eq.0) nchild=2
c
c  Check for array sizing errors.
      if (npopsiz.gt.indmax) then
       write(6,1600) npopsiz
       write(24,1600) npopsiz
       close(24)
       stop
      endif
      if (nparam.gt.nparmax) then
       write(6,1700) nparam
       write(24,1700) nparam
```

```
       close(24)
       stop
      endif
c
c  If using the microga option, reset some input variables
      if (microga.ne.0) then
       pmutate=0.0
       pcreep=0.0
       itourny=1
       ielite=1
       iniche=0
       nchild=1
       if (iunifrm.eq.0) then
          pcross=1.0
       else
          pcross=0.5
       endif
      endif

      do 3 i=1,nparam
       g0(i)=parmin(i)
       pardel(i)=parmax(i)-parmin(i)
       g1(i)=pardel(i)/dble(nposibl(i)-1)
 3    continue
      do 6 i=1,nparam
       do 7 j=1,30
          n2j=2**j
          if (n2j.ge.nposibl(i)) then
             ig2(i)=j
             goto 8
          endif
          if (j.ge.30) then
             write(6,2000)
             write(24,2000)
             close(24)
             stop
          endif
 7     continue
 8     continue
 6    continue
c
 1600 format(1x,'ERROR: npopsiz > indmax.  Set indmax = ',i6)
 1700 format(1x,'ERROR: nparam > nparmax.  Set nparmax = ',i6)
 2000 format(1x,'ERROR: You have a parameter with a number of '/
     +           1x,'   possibilities > 2**30!  If you really desire
this,'/
     +           1x,'   change the DO loop 7 statement and
recompile.'//
     +           1x,'   You may also need to alter the code to work
with'/
     +           1x,'   REAL numbers rather than INTEGER numbers;
Fortran'/
     +           1x,'   does not like to compute 2**j when j>30.')
c
      write(*,*) 'end of input'

      return
```

```
      end
c
c*********************************************************************
#
      subroutine evalout_b(iskip,iend,ibest)
c
c  This subroutine evaluates the population, assigns fitness,
c  establishes the best individual, and outputs information.
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
c
      write(*,*) 'evalout_b'
      fitsum=0.0
      best=1.0d10

      do 29 n=1,nparam
       paramsm(n)=0.0
 29   continue
c
c  Call function evaluator, write out individual and fitness, and add
c  to the summation for later averaging.

      OPEN (UNIT=20, FILE='ga-b.inp', STATUS='UNKNOWN')
      write(*,*) nchrome,nparam
      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend

      do 30 j=jstart,jend
       read(20,1075) jj,(iparent(j,k),k=1,nchrome),
     +                  (parent(j,kk),kk=1,nparam)
       write(25,1075) jj,(iparent(j,k),k=1,nchrome),
     +                  (parent(j,kk),kk=1,nparam)
       read(20,*) fitness(j)
       write(25,21) fitness(j)
21       format(f9.7)
       fitsum=fitsum+fitness(j)

       do 22 n=1,nparam
          paramsm(n)=paramsm(n)+parent(j,n)
 22    continue
c
c  Check to see if fitness of individual j is the best fitness.
       if (fitness(j).lt.best) then
          best=fitness(j)
          jbest=j
```

```
          do 24 k=1,nchrome
             ibest(k)=iparent(j,k)
 24       continue
        endif
 30   continue
c
c  Compute parameter and fitness averages.
      fbar=fitsum/dble(npopsiz)
      do 23 n=1,nparam
       paramav(n)=paramsm(n)/dble(npopsiz)
 23   continue
c
c  Write output information
      if (npopsiz.eq.1) then
       write(25,1075) 1,(iparent(1,k),k=1,nchrome),
     +                  (parent(1,k),k=1,nparam),fitness(1)
       write(25,*) ' Average Values:'
       write(25,1275) (parent(1,k),k=1,nparam),fbar
      else
       write(25,1275) (paramav(k),k=1,nparam),fbar
      endif
      write(6,1100) fbar
      write(25,1100) fbar
      write(6,1200) best
      write(25,1200) best
      write(25,1202) fbar,best
      write(6,1201) jbest
      write(25,1201) jbest
c
 1075 format(i3,1x,33i1,1x,f6.3,2(1x,f5.3),1x,f9.7)
 1100 format(1x,'Average Function Value of Generation=',f10.7)
 1200 format(1x,'Minimum Function Value              =',f10.7/)
 1201 format(1x,'Minimum Function Value Appeared on Individual',i3)
 1202 format(10x,f10.7,5x,f10.7/)
 1275 format(/' Average Values:',10x,f6.3,2(1x,f5.3),1x,f10.7/)
      return
      end
c
c######################################################################
#
      subroutine niche
c
c  Implement "niching" through Goldberg's multidimensional phenotypic
c  sharing scheme with a triangular sharing function.  To find the
c  multidimensional distance from the best individual, normalize all
c  parameter differences.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax),nposibl(nparmax),nichflg(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
```

```
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga8   / nichflg
c
c   Variable definitions:
c
c  alpha   = power law exponent for sharing function; typically = 1.0
c  del         = normalized multidimensional distance between ii and
all
c          other members of the population
c          (equals the square root of del2)
c  del2    = sum of the squares of the normalized multidimensional
c          distance between member ii and all other members of
c          the population
c  nniche  = number of niched parameters
c  sigshar = normalized distance to be compared with del; in some
sense,
c          1/sigshar can be viewed as the number of regions over which
c          the sharing function should focus, e.g. with sigshar=0.1,
c          the sharing function will try to clump in ten distinct
c          regions of the phase space.  A value of sigshar on the
c          order of 0.1 seems to work best.
c  share   = sharing function between individual ii and j
c  sumshar = sum of the sharing functions for individual ii
c
      write(*,*) 'niche'
      alpha=1.0
      sigshar=0.1
      nniche=0
      do 33 jj=1,nparam
       nniche=nniche+nichflg(jj)
 33   continue
      if (nniche.eq.0) then
       write(6,1900)
       write(24,1900)
       close(24)
       stop
      endif
      do 34 ii=1,npopsiz
       sumshar=0.0
       do 35 j=1,npopsiz
          del2=0.0
          do 36 k=1,nparam
             if (nichflg(k).ne.0) then
              del2=del2+((parent(j,k)-parent(ii,k))/pardel(k))**2.0
             endif
 36       continue
          del=(dsqrt(del2))/dble(nniche)
          if (del.lt.sigshar) then
             share=1.0-((del/sigshar)**alpha)
          else
             share=0.0
          endif
          sumshar=sumshar+share/dble(npopsiz)
 35    continue
       if (sumshar.ne.0.0) fitness(ii)=fitness(ii)/sumshar
```

```
 34   continue
c
 1900 format(1x,'ERROR: iniche=1 and all values in nichflg array = 0'/
     +           1x,'       Do you want to niche or not?')
c
      return
      end
c
c#######################################################################
#
      subroutine selectn(ipick,j,mate1,mate2)
c
c  Subroutine for selection operator.  Presently, tournament selection
c  is the only option available.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension fitness(indmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                   itourny,ielite,icreep,iunifrm,iniche,
     +                   iskip,iend,nchild,microga,kountmx

      write(*,*) 'selectn'
c
c  If tournament selection is chosen (i.e. itourny=1), then
c  implement "tournament" selection for selection of new population.
      if(itourny.eq.1) then
       call select(mate1,ipick)
       call select(mate2,ipick)
c      write(3,*) mate1,mate2,fitness(mate1),fitness(mate2)
       do 46 n=1,nchrome
          ichild(j,n)=iparent(mate1,n)
          if(nchild.eq.2) ichild(j+1,n)=iparent(mate2,n)
 46    continue
      endif
c
      write(*,*) 'end of selectn'
      return
      end
c
c#######################################################################
#
      subroutine crosovr(ncross,j,mate1,mate2)
c
c  Subroutine for crossover between the randomly selected pair.
      implicit double precision (a-h,o-z)
      save
```

```
c
      include 'params.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
c
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                   itourny,ielite,icreep,iunifrm,iniche,
     +                   iskip,iend,nchild,microga,kountmx

      write(*,*) 'crosovr'
c
      if (iunifrm.eq.0) then
c  Single-point crossover at a random chromosome point.
       call ran3(1,rand)
       if(rand.gt.pcross) goto 69
       ncross=ncross+1
       call ran3(1,rand)
       icross=2+dint(dble(nchrome-1)*rand)
       do 50 n=icross,nchrome
          ichild(j,n)=iparent(mate2,n)
          if(nchild.eq.2) ichild(j+1,n)=iparent(mate1,n)
 50    continue
      else
c  Perform uniform crossover between the randomly selected pair.
       do 60 n=1,nchrome
          call ran3(1,rand)
          if(rand.le.pcross) then
             ncross=ncross+1
             ichild(j,n)=iparent(mate2,n)
             if(nchild.eq.2) ichild(j+1,n)=iparent(mate1,n)
          endif
 60    continue
      endif
 69   continue
c
      return
      end
c
c######################################################################
#
      subroutine mutate
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension nposibl(nparmax)
      dimension child(indmax,nparmax),ichild(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
```

```
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                   itourny,ielite,icreep,iunifrm,iniche,
     +                   iskip,iend,nchild,microga,kountmx

      write(*,*) 'mutate'
c
c  This subroutine performs mutations on the children generation.
c  Perform random jump mutation if a random number is less than
pmutate.
c  Perform random creep mutation if a different random number is less
c  than pcreep.
      nmutate=0
      ncreep=0
      do 70 j=1,npopsiz
       do 75 k=1,nchrome
c  Jump mutation
          call ran3(1,rand)
          if (rand.le.pmutate) then
             nmutate=nmutate+1
             if(ichild(j,k).eq.0) then
              ichild(j,k)=1
             else
              ichild(j,k)=0
             endif
             if (nowrite.eq.0) write(6,1300) j,k
             if (nowrite.eq.0) write(24,1300) j,k
          endif
 75    continue
c  Creep mutation (one discrete position away).
       if (icreep.ne.0) then
          do 76 k=1,nparam
             call ran3(1,rand)
             if(rand.le.pcreep) then
              call decode(j,child,ichild)
              ncreep=ncreep+1
              creep=1.0
              call ran3(1,rand)
              if (rand.lt.0.5) creep=-1.0
              child(j,k)=child(j,k)+g1(k)*creep
              if (child(j,k).gt.parmax(k)) then
                 child(j,k)=parmax(k)-1.0*g1(k)
              elseif (child(j,k).lt.parmin(k)) then
                 child(j,k)=parmin(k)+1.0*g1(k)
              endif
              call code(j,k,child,ichild)
              if (nowrite.eq.0) write(6,1350) j,k
              if (nowrite.eq.0) write(24,1350) j,k
             endif
 76       continue
       endif
 70   continue
      write(6,1250) nmutate,ncreep
      write(24,1250) nmutate,ncreep
c
 1250 format(/'  Number of Jump Mutations  =',i5/
```

```
     +            '  Number of Creep Mutations =',i5)
 1300 format('*** Jump mutation performed on individual  ',i4,
     +           ', chromosome ',i3,' ***')
 1350 format('*** Creep mutation performed on individual ',i4,
     +           ', parameter  ',i3,' ***')
c
      return
      end
c
c#####################################################################
#
      subroutine newgen(ielite,npossum,ig2sum,ibest)
c
c  Write child array back into parent array for new generation.  Check
c  to see if the best parent was replicated; if not, and if ielite=1,
c  then reproduce the best parent into a random slot.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      dimension ibest(nchrmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga7   / child,ichild
c

      write(*,*) 'newgen'


      if (npossum.lt.ig2sum) call possibl(child,ichild)
      kelite=0
      do 94 j=1,npopsiz
       jelite=0
       do 95 n=1,nchrome
          iparent(j,n)=ichild(j,n)
          if (iparent(j,n).eq.ibest(n)) jelite=jelite+1
          if (jelite.eq.nchrome) kelite=1
 95    continue
 94   continue
      if (ielite.ne.0 .and. kelite.eq.0) then
       call ran3(1,rand)
       irand=1+dint(dble(npopsiz)*rand)
       do 96 n=1,nchrome
          iparent(irand,n)=ibest(n)
 96    continue
       write(24,1260) irand
      endif
c
 1260 format('  Elitist Reproduction on Individual ',i4)
c
      return
      end
```

```
c
c#########################################################################
#
      subroutine gamicro(i,npossum,ig2sum,ibest)
c
c  Micro-GA implementation subroutine
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension ibest(nchrmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent

      write(*,*) 'gamicro'
c
c  First, check for convergence of micro population.
c  If converged, start a new generation with best individual and fill
c  the remainder of the population with new randomly generated parents.
c
c  Count number of different bits from best member in micro-population
      icount=0
      do 81 j=1,npopsiz
       do 82 n=1,nchrome
          if(iparent(j,n).ne.ibest(n)) icount=icount+1
 82    continue
 81   continue
c
c  If icount less than 5% of number of bits, then consider population
c  to be converged.  Restart with best individual and random others.
      diffrac=dble(icount)/dble((npopsiz-1)*nchrome)
      if (diffrac.lt.0.05) then
      do 87 n=1,nchrome
       iparent(1,n)=ibest(n)
 87   continue
      do 88 j=2,npopsiz
       do 89 n=1,nchrome
          call ran3(1,rand)
          iparent(j,n)=1
          if(rand.lt.0.5) iparent(j,n)=0
 89    continue
 88   continue
      if (npossum.lt.ig2sum) call possibl(parent,iparent)
      write(6,1375) i
      write(24,1375) i
      endif
c
 1375 format(//'%%%%%%%  Restart micro-population at generation',
     +           i5,'  %%%%%%%')
c
      return
      end
c
```

```
c#######################################################################
#
      subroutine select(mate,ipick)
c
c  This routine selects the better of two possible parents for mating.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)

      write(*,*) 'select'
c
      if(ipick+1.gt.npopsiz) call shuffle(ipick)
      ifirst=ipick
      isecond=ipick+1
      ipick=ipick+2
      if(fitness(ifirst).lt.fitness(isecond)) then
       mate=ifirst
      else
       mate=isecond
      endif
c
write(3,*)'select',ifirst,isecond,fitness(ifirst),fitness(isecond)
c
      write(*,*) 'end of select'
      return
      end
c
c#######################################################################
#
      subroutine shuffle(ipick)
c
c  This routine shuffles the parent array and its corresponding fitness
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)

      write(*,*) 'shuffle'
c
      ipick=1
      do 10 j=1,npopsiz-1
       call ran3(1,rand)
```

```
       iother=j+1+dint(dble(npopsiz-j)*rand)
       do 20 n=1,nchrome
          itemp=iparent(iother,n)
          iparent(iother,n)=iparent(j,n)
          iparent(j,n)=itemp
 20    continue
       temp=fitness(iother)
       fitness(iother)=fitness(j)
       fitness(j)=temp
 10   continue
c
      write(*,*) 'end of shuffle'
      return
      end
c
c######################################################################
#
      subroutine decode(i,array,iarray)
c
c  This routine decodes a binary string to a real number.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)

      write(*,*) 'decode'
c
      l=1
      do 10 k=1,nparam
       iparam=0
       m=l
       do 20 j=m,m+ig2(k)-1
          l=l+1
          iparam=iparam+iarray(i,j)*(2**(m+ig2(k)-1-j))
 20    continue
       array(i,k)=g0(k)+g1(k)*dble(iparam)
 10   continue
c
      return
      end
c
c######################################################################
#
      subroutine code(j,k,array,iarray)
c
c  This routine codes a parameter into a binary string.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga2   / nparam,nchrome
```

```
      common / ga5   / g0,g1,ig2
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)

      write(*,*) 'code'
c
c  First, establish the beginning location of the parameter string of
c  interest.
      istart=1
      do 10 i=1,k-1
       istart=istart+ig2(i)
 10   continue
c
c  Find the equivalent coded parameter value, and back out the binary
c  string by factors of two.
      m=ig2(k)-1
      if (g1(k).eq.0.0) return
      iparam=nint((array(j,k)-g0(k))/g1(k))
      do 20 i=istart,istart+ig2(k)-1
       iarray(j,i)=0
       if ((iparam+1).gt.(2**m)) then
          iarray(j,i)=1
          iparam=iparam-2**m
       endif
       m=m-1
 20   continue
c     write(3,*)array(j,k),iparam,(iarray(j,i),i=istart,istart+ig2(k)-
1)
c
      return
      end

c
c#####################################################################
#
      subroutine restart(i,istart,kount)
c
c  This subroutine writes restart information to the ga.restart file.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                 itourny,ielite,icreep,iunifrm,iniche,
     +                 iskip,iend,nchild,microga,kountmx

      write(*,*) 'restart'
c
      kount=kount+1
      if(i.eq.maxgen+istart-1 .or. kount.eq.kountmx) then
       OPEN (UNIT=25, FILE='ga.restart', STATUS='OLD')
       rewind 25
```

```
       write(25,*) i+1,npopsiz
       do 80 j=1,npopsiz
          write(25,1500) j,(iparent(j,l),l=1,nchrome)
 80    continue
       CLOSE (25)
       kount=0
      endif
c
 1500 format(i5,3x,33i2)
c
      return
      end
c
c#######################################################################
#
      subroutine ran3(idum,rand)
c
c  Returns a uniform random deviate between 0.0 and 1.0.  Set idum to
c  any negative value to initialize or reinitialize the sequence.
c  This function is taken from W.H. Press', "Numerical Recipes" p. 199.
c
      implicit double precision (a-h,m,o-z)
      save
c      implicit real*4(m)
      parameter (mbig=4000000.,mseed=1618033.,mz=0.,fac=1./mbig)
c     parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1./mbig)
c
c  According to Knuth, any large mbig, and any smaller (but still
large)
c  mseed can be substituted for the above values.

      dimension ma(55)
      data iff /0/

      write(*,*) 'ran3'
      if (idum.lt.0 .or. iff.eq.0) then
       iff=1
       mj=mseed-dble(iabs(idum))
       mj=dmod(mj,mbig)
       ma(55)=mj
       mk=1
       do 11 i=1,54
          ii=mod(21*i,55)
          ma(ii)=mk
          mk=mj-mk
          if(mk.lt.mz) mk=mk+mbig
          mj=ma(ii)
 11    continue
       do 13 k=1,4
          do 12 i=1,55
             ma(i)=ma(i)-ma(1+mod(i+30,55))
             if(ma(i).lt.mz) ma(i)=ma(i)+mbig
 12       continue
 13    continue
       inext=0
       inextp=31
       idum=1
```

```
      endif
      inext=inext+1
      if(inext.eq.56) inext=1
      inextp=inextp+1
      if(inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
      rand=mj*fac

      write(*,*) 'end of ran3'
      return
      end
c
c#########################################################################
#
c
      subroutine possibl(array,iarray)
c
c  This subroutine determines whether or not all parameters are within
c  the specified range of possibility.    If not, the parameter is
c  randomly reassigned within the range.  This subroutine is only
c  necessary when the number of possibilities per parameter is not
c  optimized to be 2**n, i.e. if npossum < ig2sum.
c
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax),nposibl(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)

      write(*,*) 'possibl'
c
      do 10 i=1,npopsiz
       call decode(i,array,iarray)
       do 20 j=1,nparam
          n2ig2j=2**ig2(j)
          if(nposibl(j).ne.n2ig2j .and. array(i,j).gt.parmax(j)) then
             call ran3(1,rand)
             irand=dint(dble(nposibl(j))*rand)
             array(i,j)=g0(j)+dble(irand)*g1(j)
             call code(i,j,array,iarray)
             if (nowrite.eq.0) write(6,1000) i,j
             if (nowrite.eq.0) write(24,1000) i,j
          endif
 20    continue
 10   continue
c
 1000 format('*** Parameter adjustment to individual     ',i4,
     +           ', parameter  ',i3,' ***')
c
```

```
      return
      end

c
c#########################################################################
#
      subroutine newgen_opt(iskip,iend,ibest)
c
c  This subroutine decode and write output for the new generation.
      implicit double precision (a-h,o-z)
      save
c
      include 'params.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga5   / g0,g1,ig2
c
      write(*,*) 'newgen_opt'
      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend
      do 30 j=jstart,jend
       call decode(j,parent,iparent)
      write(24,1076) j,(iparent(j,k),k=1,nchrome),
     +                  (parent(j,kk),kk=1,nparam)
      write(*,1076) j,(iparent(j,k),k=1,nchrome),
     +                  (parent(j,kk),kk=1,nparam)
 30   continue
c
 1076 format(i3,1x,33i1,1x,f6.3,1x,f5.3,1x,f5.3)
c
      return
      end
```

Appendix D

(Coding and Decoding in GA)

Coding and Decoding of the design parameters in the Genetic Algorithms (GAs)

The coding and decoding is a major step of the Genetic Algorithms. Based on the algorithms, this step takes the design requirements (the number of variables, domain (upper and lower limits), and precision requirement (decimal points) of each variable) as input in creating the binary strings (coding). The created binary strings are then further translated (decoded) into set of number as design parameters. The detail is described below.

I. Coding:

For the optimization problem in current research, the design constraint for variables: blank holder force (x_1) for high strength Al 7075-T6, low strength Al 2024-T3 (x_2) and Co-efficient of Friction(x_3) are

$$7.5KN < L\_BHF \text{ (Al 7075-T6)} < 12.5KN$$

$$3KN < R\_BHF \text{ (Al 2024-T3} < 6KN$$

$$0.05 < \text{Co-efficient of Friction} < 0.20$$

The precision requirement for x_1, x_2 and x_3 are 3 decimal places. The domain of x_1 has length of 5.000 (= 12.500-7.500), with the precision requirement of 3 decimal places; the range [7.500, 12.500] should be divided into at least $5.000 * 10^3$ (5000) equal size ranges. Since

$$2^{12} \leq 5000 \leq 2^{13}$$

13 bits are required as the first part of the chromosome.

Similarly, x_2 has a length of 3.000 (=6.000 – 3.000), with a precision requirement of 3 decimal places, the range [6.000, 3.000] should be divided into at least 3.000 * 10^3 (3000) equal size ranges, then

$$2^{11} \le 3000 \le 2^{12}$$

x_3 has a length of 0.150 (=.020 – 0.050), with a precision requirement of 3 decimal places, the range [=.020-0.050] should be divided into at least 0.150 * 10^3 (150) equal size ranges, then

$$2^7 \le 150 \le 2^8$$

With a random number generator taken from W. H. Press, "Numerical Recipes", chromosomes with a length totally 13+12+8=33 chromosomes were created. A typical chromosome is showing below,

000011100111111100100000000001100
x_1 x_2 x_3

II. Decoding:

In the decoding process, the binary strings were translated back as values of the design parameters by the equation

$$x_i = a_i + decimal(1000.....110_2) \times \frac{b_i - a_i}{2^{m_i} - 1}$$

In which a_i and b_i are the lower and upper limits, m_i is the number of bits for the i^{th} variable.

$$x_1 = 7.500 + decimal(0000111001111111001000000000001100_2) \times \frac{12.500 - 7.500}{2^{13} - 1}$$

$= 7.783$

$$x_2 = 3.000 + decimal(00001110011111110010000000001100_2) \times \frac{6.000 - 3.000}{2^{12} - 1}$$

$= 5.673$

$$x_3 = 0.050 + decimal(01001110_2) \times \frac{0.200 - 0.050}{2^8 - 1}$$

$= 0.057$

So the chromosome (binary string)

00001110011111110010000000001100

corresponds to $(x_1, x_2, x_3) = (7.783, 5.673, 0.057)$

Appendix E
(Results from GA)

Table E.1 Results from the 1st generation

Gen 1	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Min. (m)	Max. (m)	Min. (m)	Min. (m)	
1	7.721	3.763	0.124	6.7689E-04	4.3588E-04	1.3140E-03	9.8348E-04	1.7740E-03	1.2532E-03
2	8.677	4.986	0.07	1.2395E-03	6.0609E-04	1.4770E-03	9.8585E-04	1.7681E-03	2.3470E-03
3	10.348	4.282	0.093	8.2534E-03	5.7752E-04	2.8860E-03	7.5020E-04	1.7612E-03	9.8330E-04
4	11.131	5.995	0.118	1.6010E-03	6.4213E-04	1.2933E-03	7.6833E-03	1.7681E-03	1.6818E-03
5	9.817	5.145	0.185	7.8348E-04	5.5085E-04	6.2554E-03	6.2554E-03	1.7225E-03	1.4786E-03

Table E.2 Results from the 2th generation

Gen 2	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	10.330	5.675	0.173	1.635E-03	5.838E-04	1.769E-03	1.877E-04	1.762E-03	9.099E-04
2	12.336	4.234	0.087	1.911E-03	5.701E-04	6.634E-03	5.370E-04	1.753E-03	1.971E-03
3	8.357	4.894	0.199	1.689E-03	4.669E-04	5.502E-03	2.477E-04	1.744E-03	1.651E-03
4	9.457	3.362	0.182	1.882E-03	5.268E-04	7.177E-03	4.160E-04	1.746E-03	2.053E-03
5	9.458	5.618	0.197	1.770E-03	6.050E-04	5.369E-03	4.641E-04	1.744E-03	1.700E-03

Table E.3 Results from the 3th generation

Gen 3	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	11.893	3.536	0.148	1.697E-03	4.830E-04	1.604E-03	3.907E-04	1.745E-03	9.030E-04
2	8.731	5.159	0.134	1.939E-03	6.458E-04	4.813E-03	2.107E-04	1.757E-03	1.573E-03
3	11.707	5.626	0.115	1.636E-03	5.231E-04	8.415E-03	5.920E-04	1.771E-03	2.268E-03
4	12.247	5.482	0.061	1.742E-03	5.951E-04	5.268E-03	4.462E-04	1.756E-03	1.661E-03
5	9.023	5.204	0.101	1.630E-03	5.017E-04	7.826E-03	1.834E-04	1.779E-03	2.095E-03

Table E.4 Results from the 4th generation

Gen 4	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	11.478	4.261	0.076	1.877E-03	6.339E-04	2.251E-03	3.527E-04	1.756E-03	1.085E-03
2	11.666	5.777	0.080	1.740E-03	5.790E-04	4.821E-03	6.065E-04	1.745E-03	1.587E-03
3	9.330	5.589	0.173	1.772E-03	5.686E-04	2.644E-03	4.012E-04	1.777E-03	1.132E-03
4	8.379	5.822	0.149	1.902E-03	6.348E-04	3.471E-03	3.127E-04	1.758E-03	1.324E-03
5	7.792	4.057	0.092	1.792E-03	5.428E-04	7.142E-03	1.398E-04	1.743E-03	1.989E-03

Table E.5 Results from the 5th generation

Gen 5	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	10.908	5.136	0.132	1.838E-03	4.558E-04	3.853E-03	1.289E-04	1.775E-03	1.323E-03
2	9.283	3.630	0.162	1.926E-03	4.494E-04	5.705E-03	2.275E-04	1.776E-03	1.734E-03
3	10.422	3.657	0.079	1.774E-03	4.440E-04	5.698E-03	1.066E-04	1.754E-03	1.680E-03
4	7.834	5.714	0.072	1.642E-03	4.847E-04	2.062E-03	1.033E-04	1.748E-03	9.453E-04
5	7.755	4.967	0.154	1.731E-03	4.401E-04	2.309E-03	4.074E-04	1.758E-03	1.050E-03

Table E.6 Results from the 6th generation

Gen 6	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	12.063	4.018	0.158	1.883E-03	6.374E-04	1.753E-03	4.272E-04	1.774E-03	9.777E-04
2	8.516	3.143	0.154	1.848E-03	5.164E-04	6.968E-03	4.420E-04	1.756E-03	9.674E-04
3	8.273	4.096	0.197	1.826E-03	6.250E-04	4.412E-03	5.879E-04	1.773E-03	9.985E-04
4	7.532	3.318	0.187	1.729E-03	5.699E-04	1.989E-03	1.595E-04	1.780E-03	1.000E-03
5	10.200	5.242	0.076	1.664E-03	5.101E-04	5.664E-03	1.311E-04	1.748E-03	9.863E-04

Table E.7 Results from the 7th generation

Gen 7	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	9.056	3.786	0.090	1.772E-03	5.264E-04	1.592E-03	6.052E-04	1.753E-03	9.493E-04
2	10.706	4.541	0.102	1.844E-03	5.523E-04	5.134E-03	3.675E-04	1.762E-03	1.626E-03
3	8.213	4.071	0.180	1.677E-03	4.839E-04	4.565E-03	3.719E-04	1.772E-03	1.477E-03
4	8.524	3.257	0.133	1.764E-03	6.447E-04	4.719E-03	5.221E-04	1.759E-03	1.578E-03
5	10.930	4.371	0.061	1.615E-03	4.856E-04	4.416E-03	2.570E-04	1.768E-03	1.417E-03

Table E.8 Results from the 8th generation

Gen 8	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.244	4.942	0.063	1.605E-03	6.239E-04	4.916E-03	2.029E-04	1.759E-03	1.538E-03
2	8.975	3.764	0.181	1.604E-03	4.835E-04	6.953E-03	3.335E-04	1.753E-03	1.941E-03
3	11.968	4.315	0.143	1.866E-03	4.415E-04	2.994E-03	3.448E-04	1.743E-03	1.199E-03
4	8.917	3.261	0.060	1.837E-03	6.160E-04	6.770E-03	3.123E-04	1.748E-03	1.959E-03
5	8.611	5.328	0.088	1.767E-03	4.747E-04	7.913E-03	5.164E-04	1.743E-03	2.181E-03

Table E.9 Results from the 9th generation

Gen 9	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	12.283	3.337	0.144	1.681E-03	4.403E-04	1.388E-03	5.450E-04	1.771E-03	8.618E-04
2	8.672	4.141	0.101	1.680E-03	5.180E-04	3.169E-03	3.247E-04	1.780E-03	1.197E-03
3	10.832	5.332	0.051	1.930E-03	5.706E-04	3.632E-03	5.999E-04	1.768E-03	1.387E-03
4	8.802	4.243	0.174	1.629E-03	5.633E-04	7.748E-03	4.684E-04	1.777E-03	2.126E-03
5	8.725	3.687	0.069	1.664E-03	6.324E-04	4.530E-03	1.344E-04	1.743E-03	1.467E-03

Table E.10 Results from the 10th generation

Gen 10	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.703	5.956	0.102	1.884E-03	4.381E-04	8.215E-03	3.571E-04	1.758E-03	2.232E-03
2	10.716	3.301	0.146	1.602E-03	5.666E-04	4.337E-03	4.022E-04	1.763E-03	1.441E-03
3	7.821	3.837	0.124	1.662E-03	5.401E-04	7.207E-03	5.410E-04	1.779E-03	2.025E-03
4	8.774	3.173	0.150	1.860E-03	4.896E-04	4.512E-03	1.564E-04	1.765E-03	1.471E-03
5	8.337	3.921	0.164	1.848E-03	4.342E-04	6.437E-03	4.459E-04	1.749E-03	1.899E-03

Table E.11 Results from the 11th generation

Gen 11	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	12.490	4.350	0.176	1.906E-03	5.746E-04	6.111E-03	2.787E-04	1.753E-03	9.684E-04
2	11.474	3.967	0.059	1.672E-03	4.359E-04	4.828E-03	1.615E-04	1.762E-03	9.877E-04
3	11.394	4.465	0.113	1.680E-03	5.464E-04	7.563E-03	3.218E-04	1.759E-03	1.003E-03
4	10.600	4.180	0.186	1.673E-03	5.202E-04	5.196E-03	6.134E-04	1.743E-03	9.589E-04
5	7.985	5.502	0.199	1.671E-03	5.914E-04	2.821E-03	1.712E-04	1.769E-03	9.765E-04

Table E.12 Results from the 12th generation

Gen 12	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	9.144	5.245	0.126	1.624E-03	4.538E-04	6.408E-03	6.067E-04	1.753E-03	1.856E-03
2	8.928	3.515	0.139	1.941E-03	4.408E-04	3.727E-03	3.033E-04	1.748E-03	1.362E-03
3	8.743	3.671	0.118	1.847E-03	5.513E-04	8.404E-03	2.799E-04	1.752E-03	2.268E-03
4	8.393	3.533	0.107	1.808E-03	5.499E-04	4.732E-03	2.753E-04	1.760E-03	1.530E-03
5	7.583	5.810	0.178	1.608E-03	4.416E-04	2.558E-03	3.703E-04	1.763E-03	1.063E-03

Table E.13 Results from the 13th generation

Gen 13	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.110	5.608	0.050	1.629E-03	5.887E-04	8.260E-03	4.248E-04	1.767E-03	2.217E-03
2	8.021	4.160	0.134	1.902E-03	4.997E-04	2.891E-03	1.698E-04	1.754E-03	1.168E-03
3	11.941	4.886	0.168	1.842E-03	5.666E-04	6.346E-03	1.253E-04	1.769E-03	1.848E-03
4	8.217	5.726	0.075	1.934E-03	4.688E-04	5.554E-03	2.324E-04	1.755E-03	1.716E-03
5	11.843	3.820	0.159	1.752E-03	4.655E-04	3.673E-03	2.866E-04	1.765E-03	1.311E-03

Table E.14 Results from the 14th generation

Gen 14	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	10.329	4.487	0.195	1.927E-03	4.780E-04	3.188E-03	3.097E-04	1.746E-03	1.254E-03
2	9.856	4.701	0.077	1.931E-03	4.592E-04	2.684E-03	3.470E-04	1.774E-03	1.144E-03
3	11.406	3.621	0.195	1.646E-03	6.179E-04	8.231E-03	2.447E-04	1.762E-03	2.200E-03
4	7.518	3.865	0.115	1.915E-03	6.155E-04	7.048E-03	3.908E-04	1.752E-03	2.035E-03
5	8.563	3.118	0.135	1.904E-03	5.890E-04	4.489E-03	3.866E-04	1.748E-03	1.522E-03

Table E.15 Results from the 15th generation

Gen 15	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	9.040	5.769	0.083	1.678E-03	5.364E-04	2.074E-03	4.844E-04	1.746E-03	1.006E-03
2	11.600	5.338	0.077	1.640E-03	4.975E-04	6.763E-03	2.669E-04	1.757E-03	1.889E-03
3	10.363	3.809	0.163	1.868E-03	5.015E-04	2.396E-03	3.983E-04	1.750E-03	1.093E-03
4	8.241	4.239	0.185	1.765E-03	6.119E-04	7.599E-03	3.930E-04	1.740E-03	2.130E-03
5	9.393	5.944	0.177	1.794E-03	5.350E-04	5.126E-03	4.623E-04	1.748E-03	1.629E-03

Table E.16 Results from the 16th generation

Gen 16	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	10.879	5.020	0.145	1.880E-03	5.591E-04	2.684E-03	4.753E-04	1.765E-03	1.166E-03
2	11.352	3.453	0.120	1.868E-03	4.838E-04	7.352E-03	3.976E-04	1.775E-03	2.067E-03
3	10.262	5.338	0.197	1.602E-03	4.640E-04	6.857E-03	3.639E-04	1.747E-03	1.924E-03
4	11.448	5.209	0.191	1.669E-03	6.124E-04	4.592E-03	5.374E-04	1.742E-03	1.526E-03
5	10.910	3.751	0.091	1.659E-03	4.568E-04	2.931E-03	3.120E-04	1.761E-03	1.136E-03

Table 6.17 Results from the 17th generation

Gen 17	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.832	3.805	0.175	1.784E-03	5.388E-04	7.829E-03	5.243E-04	1.755E-03	1.034E-03
2	8.754	3.924	0.184	1.764E-03	6.199E-04	4.619E-03	1.059E-04	1.763E-03	9.589E-05
3	11.531	5.377	0.056	1.883E-03	4.666E-04	4.736E-03	1.623E-04	1.779E-03	9.601E-04
4	12.428	5.234	0.172	1.719E-03	4.384E-04	7.107E-03	4.004E-04	1.779E-03	9.659E-04
5	10.295	5.705	0.136	1.866E-03	5.941E-04	2.816E-03	3.282E-04	1.773E-03	9.770E-04

Table 6.18 Results from the 18th generation

Gen 18	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.959	5.661	0.053	1.922E-03	4.787E-04	2.557E-03	1.649E-04	1.743E-03	1.105E-03
2	11.249	3.536	0.115	1.646E-03	5.073E-04	5.663E-03	4.418E-04	1.767E-03	1.692E-03
3	10.517	3.442	0.194	1.854E-03	6.330E-04	2.034E-03	4.845E-04	1.772E-03	1.045E-03
4	10.331	3.403	0.192	1.735E-03	5.236E-04	6.348E-03	4.490E-04	1.777E-03	1.858E-03
5	11.347	3.757	0.063	1.798E-03	5.531E-04	2.591E-03	4.523E-04	1.770E-03	1.124E-03

Table 6.19 Results from the 19th generation

Gen 19	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Avg. (m)	Max. (m)	Avg. (m)	Min. (m)	
1	8.333	4.488	0.151	1.921E-03	5.102E-04	1.484E-03	5.593E-04	1.755E-03	9.567E-04
2	11.835	5.048	0.191	1.828E-03	6.299E-04	1.365E-03	4.699E-04	1.764E-03	9.012E-04
3	8.193	5.342	0.103	1.855E-03	4.450E-04	4.599E-03	3.648E-04	1.740E-03	1.531E-03
4	9.826	3.698	0.089	1.757E-03	4.649E-04	7.502E-03	3.036E-04	1.765E-03	2.070E-03
5	9.353	5.389	0.128	1.714E-03	5.010E-04	4.576E-03	6.136E-04	1.744E-03	1.524E-03

Table 6.20 Results from the 20th generation

Gen 20	Design Parameters			Weld Movement		Shape Deviation		Max Thinning	PID
	L_BHF (KN)	R_BHF (KN)	m	Max. (m)	Min. (m)	Max. (m)	Min. (m)	Min. (m)	
1	7.783	3.673	0.057	6.7689E-04	4.3588E-04	1.3140E-03	9.8348E-04	1.8210E-03	9.6910E-04
2	9.178	5.057	0.122	8.3947E-04	6.0609E-04	1.4770E-03	9.8585E-04	1.7458E-03	1.0310E-04
3	10.216	4.279	0.182	8.0266E-04	5.7752E-04	8.4225E-03	8.7428E-03	1.7405E-03	1.9032E-03
4	8.032	4.648	0.154	8.8073E-04	6.4213E-04	1.2933E-03	7.6833E-03	1.8320E-03	9.7900E-04
5	11.034	5.963	0.121	7.8348E-04	5.5085E-04	6.2554E-03	6.2554E-03	1.7800E-03	1.2194E-03

VITA

Surendra Chitti Babu was born in Chennai, India on August 6, 1978. He received his Bachelor of Engineering (B.E.). degree in Mechanical Engineering in June 1999 from University of Madras, Tamil Nadu, India and Master of Science (M.S.) degree in Mechanical and Aerospace Engineering in December 2002 from the University of Missouri – Columbia in Columbia, Missouri. He received his Doctor of Philosophy (Ph.D.) in Mechanical and Aerospace Engineering from the University of Missouri – Columbia in Columbia, Missouri, in December 2007.