A Framework for Predicting the Geometrical Errors of Thin-Floor Components in End Milling Using a Flexible Fixturing Setup

by

Amir AMJADIAN

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLEMENT FOR A MASTER'S DEGREE
WITH THESIS IN MECHANICAL ENGINEERING
M.A.Sc.

MONTREAL, AUGUST 13th, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS THESIS M.SC.A.

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Jean-François Chatelain, Thesis Supervisor
Department of Mechanical Engineering, École de technologie supérieure

Mr. Patrick Terriault, President of the Board of Examiners
Department of Mechanical Engineering, École de technologie supérieure

Mr. Lucas Hof., Member of the jury
Department of Mechanical Engineering, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

AUGUST 4TH, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENT

Words cannot fully express my true appreciation for the help and wisdom I received from others in the past 3 years, and this is only an attempt in doing so.

I would like to thank my brother Ehsan Amjadian and my mother Dr. Khadijeh Eshaghi for their selfless support and sacrifice that enabled such an invaluable undertaking. I could travel to seek knowledge and was able to contribute to the scientific repertoire in mechanical engineering.

Immeasurable gratitude goes to my supervisor: professor: Jean-François Chatelain. He took me under his wings and patiently guided me through many research obstacles that would seem baffling at first. He provided invaluable feedback on my analysis results. Professor Chatelain noticed my passion for machining and nurtured them with his depth and breadth of knowledge in the field, setting me on an eventful path of mastery in CAD/CAE and in-domain Python scripting applied to machining and end milling.

I would like to thank Dr. Davood Shahriari for many intellectual discussions and conversations regarding simulation problems in Abaqus. He offered great insights and shed light on countless features and capabilities of Abaqus integral to the development of my thesis.

Many thanks to professor Serafettin Engin who guided me through my MATLAB script with respect to several challenging problems. I learned a lot from him, both during classes and beyond. The main assumption of this thesis is inspired by the knowledge he shared with me during the course.

I am grateful to the lab technicians specially Joel Grignon who went above and beyond in conducting various critical measurements. His enthusiasm and generosity with his time as the technological application technician are greatly appreciated.

Special thanks to Mohammad Ebrahim Bajgholi for his generous support both as a friend and officemate.

APPLICATION POUR LA PRÉDICTION DES ERREURS GÉOMÉTRIQUES ISSUES DE L'USINAGE DE PIÈCES MINCES EN MONTAGE FLEXIBLE

Amir AMJADIAN

RESUME

L'utilisation de composantes usinées avec parois minces est très répandue dans l'industrie aéronautique dans le but d'alléger les aéronefs. L'usinage de telles pièces, généralement de grandes dimensions et ayant des formes avec surfaces complexes, nécessite l'utilisation de gabarit de montage dédiés à chaque famille de composantes. La fabrication et l'entreposage de tels gabarits est très coûteux pour cette industrie, ce qui a mené au développement de systèmes de montage flexibles basés sur de nombreux appuis ponctuels ajustables en hauteur. Ces montages flexibles sont actuellement utilisés pour des pièces rigides et des pièces minces devant subir des opérations ne générant aucune déflexion sous effet des forces appliquées. Cette recherche vise à développer un modèle numérique simplifié permettant de prédire les déformations de plaques supportées par appuis ponctuels lors d'opérations d'usinage de type « rainurage ». Un modèle de force de coupe est utilisé pour évaluer les forces moyennes en différentes positions de la trajectoire de coupe pour prédire les déformations à l'aide d'un algorithme itératif programmé avec le langage Python utilisé par un API « Application Procedural Interface » du logiciel d'éléments finis Abaqus. La comparaison des résultats du modèle proposé avec ceux obtenus lors d'un essai d'usinage effectué à l'aide d'un montage flexible équipé d'un capteur de déplacement montre une différence dans les déplacements obtenus, tel que prévu. En effet, la simplification de ce premier modèle proposé, ne tient pas encore en considération les changements de géométrie en cours d'usinage. L'application proposée a cependant été développée dans le but de tenir compte de ces changements de géométrie dans le cadre de développements futurs.

Mots-clés: Composantes aux parois minces, éléments finis, Abaqus, Python, Interface de programmation pour applications (API), prédiction d’erreurs géométriques, montage flexible.

A FRAMEWORK FOR PREDICTING THE GEOMETRICAL ERRORS OF THIN-FLOOR COMPONENTS IN END MILLING USING A FLEXIBLE FIXTURING SETUP

Amir AMJADIAN

ABSTRACT

Thin-floor components are commonly used in aerospace industry such as spars, and bulkheads. Their superlative ratio of strength to weight makes them highly demanded by manufacturers. One of the inevitable problems in machining such products is the deformation which is induced by cutting forces during manufacturing and causes surface form errors. This issue becomes more important when a flexible fixturing system is involved to support the workpiece. In order to assure the machining accuracy in milling with this situation, a prediction model for surface dimensional errors is required to avoid costly compensation operations and reach high productivity. In this dissertation, a structured simulation procedure in milling suitable for part geometrical errors induced by axial cutting forces is proposed while the workpiece is fixed to (on) a flexible fixturing setup. The process is designed in an FE model as an implicit/Static analysis of material removal and deformation under the influence of applied axial cutting forces. A cutting force model is used to measure the average cutting forces in different positions of the cutter. An Abaqus Python API is applied to conduct numerous iterative procedures during creating the model in Abaqus including parametric study, creating repetitive geometry and managing multiple steps and forces. The theoretical model is based on the assumption that the workpiece is an elastic body. Because there is not much support on the back face of the plate, it will deform vertically and causes cutting deflections.

The advantages of the proposed model over previous works are:

1. A framework is proposed to predict the workpiece non-linear behavior during machining due to its constant changing geometry.
2. It greatly speeds up the experimental loop.

3. It can help to develop an off-line error compensation model in Abaqus by manipulating and adjusting the depth of cut value through the trajectory in an iterative process.

The first results show a gap between the theoretical and experimental models due to the simplification of a dynamic movement into a static behavior. Therefore, in the next step, the numerical model is developed in an attempt to reduce this gap.

Index Terms: Thin-floor components, FEA, Abaqus, Python, application programming Interface (API), error prediction, end milling, flexible fixturing setup.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

Micro-electro-mechanical systems	MEMS
Depth of cut	DOC
Radial depths of cut	RDOC
Finite element analysis	FEA
Finite element method	FEM
Cutter-workpiece engagement	CWE
Computer-aided engineering	CAE
Computer-aided design	CAD
Computer-aided manufacturing	CAM
Hyper text markup language	HTML
Numerical control	NC
Genetic algorithm	GA
Ant colony algorithm	ACA
Graphical User Interface	GUI
Rigid iterative algorithm	RIAL
Flexible iterative algorithm	FIAL
Double iterative algorithm	DIAL
Maximum Surface Form Error	MSFE
Application programming Interface	API

International system of units	SI
Model database	Mdb
Output database	Odb
Coordinate measurement machine	CMM

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

mm	millimeter
rev	revolution
N	newton
s	second
mJ	millijoule
Rad	radian
°	Degree
∠	angle

CHAPTER 1

INTRODUCTION

Thin-walled components in industries such as aeronautics and aviation, automotive and in micro-electro-mechanical systems (MEMS) are required to have a structural homogeneity and a highly desirable ratio of strength to weight, to ensure the stiffness and low weight quality of the parts. For instance, parts such as ribs, stringers, webs and skins in aircraft structures are considered thin-walled components. Fixturing and machining such plates specially with complex geometry have been a challenge in the field to overcome . Fixtures are support devices to locate, restrain and hold up workpieces in manufacturing processes. Different types of manufacturing processes require different kinds of fixtures namely assembly fixture, inspection fixture and machining fixture. Clamping and positioning thin components during machining is considered in this research. Poor design in fixturing such parts can cause major elastic or plastic deformation and eventually reduce the accuracy of machining. Hence, a compatible fixturing system for these kinds of parts is needed. An adjustable fixturing setup was proposed to address this issue by scientists (Asada & By, 1985; Cai, Hu, & Yuan, 1996; Feng & Menq, 1996; Fields, Youcef-Toumi, & Asada, 1989; Sela, Gaudry, Dombre, & Benhabib, 1997). These systems are highly demanded by manufacturers in the recent years. Complex workpieces are commonly difficult and time-consuming to set up, but the proposed setup has critical advantages including its flexibility and fixturing options that can be repeated rapidly. This results in less cycle time, minimizing processing cost and high productivity. New methods for milling thin and resilient components which are clamped to this fixturing system have been investigated for years (Aoyama & Kakinuma, 2005; Bao, Wang, He, Kang, & Guo, 2021; Diez et al., 2015; Y. G. Kang, Yang, Huang, & Zhu, 2014; Prabhaharan, Padmanaban, & Krishnakumar, 2006; Zhang et al., 2021). The tooling system is able to clamp the workparts with numbers of adjustable location pins. The main function of these pins is to position and hold the workpiece during the machining operation. Knowing the geometry of the part, these

pins can adapt to more complicated shapes (Figure 1-1). Given the workpiece flexibility and the lack of support, this task requires an off-line prediction model which is carried out using an FEA (Finite Element Analysis) software program to assure the machining accuracy of the part through the trajectory, in order to avoid costly processes and increase the productivity (Adetoro & Wen, 2008; Bolar & Joshi, 2017; Escamilla, Zapata, Gonzalez, Gámez, & Guerrero, 2010; Ji, Qin, & Ye, 2011; Rai & Xirouchakis, 2008; S. Ratchev, Liu, Huang, & Becker, 2006; Schmitz, Bayly, Soons, & Dutterer, 2001; Tsai & Liao, 1999). Using experience-based methods of cutting parameters and back plating support systems do not help to reach the optimum conditions. This is the usual approach in milling operations. For example, plunge milling strategy is a great approach for roughing and slotting operations, but when a canned cycle is considered, we are losing the productivity with this method. In general, reasons such as time-consuming to set up the operation, sacrificing resources and productivity, and necessary large floor space for storage, could convince us the need of a new approach.

Therefore, the questions are:

1. In what way one can facilitate the experimentation on modeling machining processes?
2. Can the proposed solution be applied to an FEA for modeling the error prediction or compensation in terms of flexible milling?

Proposed answers:

1. Creating a systematic semi-automated framework which is friendly to any type of machining simulation.
2. Yes, a framework was able to automate preprocessing and postprocessing for modeling a flexible milling operation of thin components by using Python scripting in Abaqus.

Figure 1-1 Flexible fixturing setup
(MTorres, n.d.)

The approach proposed in this research offers a framework to predict the deformation of a thin-floor aluminum plaque, due to cutting forces during a slotting operation, using an adaptable fixturing setup. Achieving this objective would help researchers to develop a software program capable of optimizing the tool path through an iterative process, prior to machining the parts.

There are various aeronautical components which are categorized as thin-wall parts including skin panels, shells, turbine blades, hubs, spars, frames, ribs, stringers, and blisks. They are designed to reduce the mechanical assembly features and to maintain a stable behavior of the part. Advanced alloys such as aluminum and titanium which broadly used in aeronautical section are applied to manufacture thin-wall structures. They have excellent strength properties, corrosion resistance and weight-resistance ratio. Hence, a large amount of material is needed to be removed from the original block by a machining process (buy to fly ratio of 30:1). The thickness of a thin-wall part is six times lower than the other two directions, consequently, they are resilient and can be simply bent. Because of the aforementioned reason,

static and dynamic problems will appear during machining these parts (Del Sol et al., 2019). Figure 1-2 illustrates an example of thin-walled components in aerospace industry.

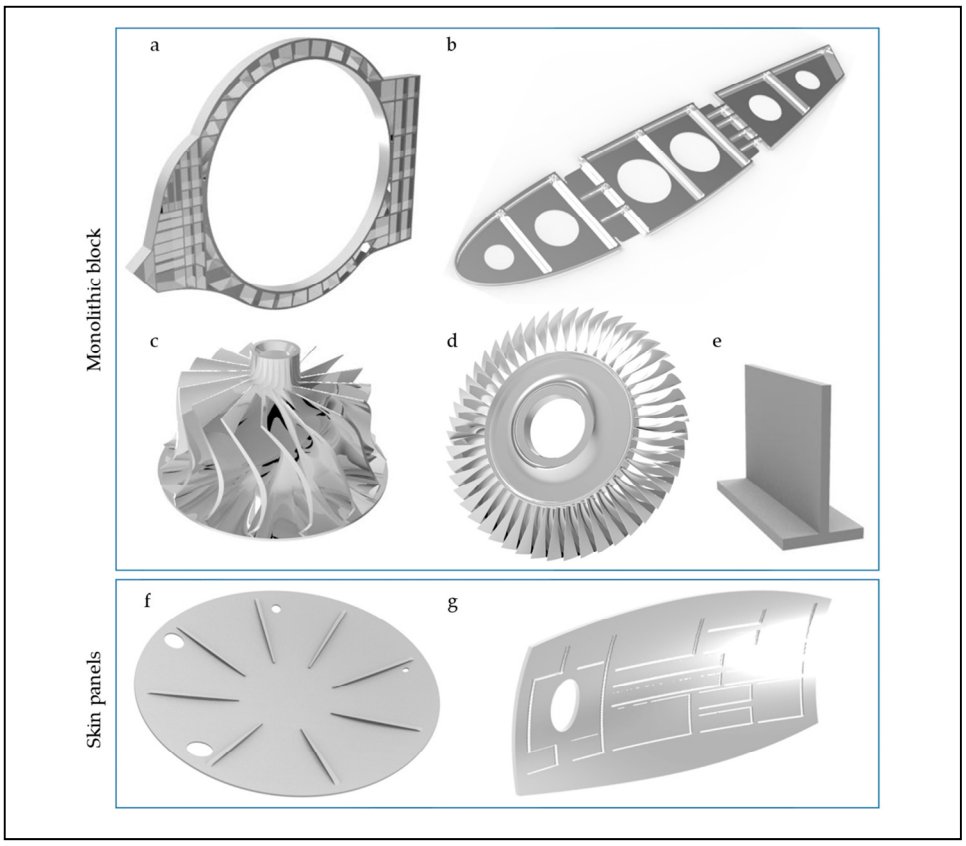


Figure 1-2 Examples of thin-walled components: (**a**) frame; (**b**) rib;(**c**) impeller;(**d**) blisk; (**e**) sample parts; (**f**) bulkhead; (**g**) fuselage skin.

(Del Sol et al., 2019)

The proposed FEA model is based on a static solution considering applied cutting forces on the cutting region in different positions of the tool, through the trajectory. In this research, it is assumed that the plate is an elastic body. Due to the lack of support on the bottom face of the plate, the generated cutting forces -which are the result of the material removal- can deform the plate in the X-Z plane. During machining of thin components, that is the prime reason of deviation in the cutting process (S. Ratchev, Liu, & Becker, 2005; S. Ratchev et al., 2006). The deformation zone is mostly concentrated in the middle of the metal part that is surrounded among the locator pins (see Figure 1-3). In section 3.5, a simplified cutting force model based

on a few assumptions has been considered in the FEA model (Abaqus). These forces are calculated in a MATLAB script from a cutter-workpiece engagement (CWE) model (Altintas, 2012), and their average magnitudes are imported as input in Abaqus software program. To facilitate the modeling process in Abaqus, various scripts in Python are developed to conduct numerous iterative procedures. Lastly, the simulation results will be compared to the experimental data and will be discussed in conclusion section.

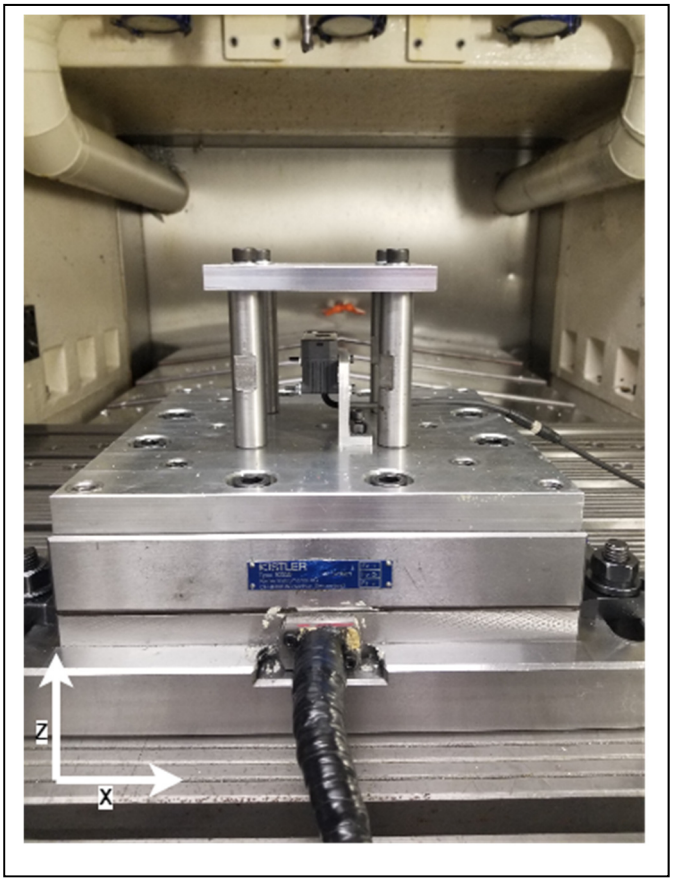


Figure 1-3 The proposed testbed layout

The following chapters in the present dissertation are organized as follow. CHAPTER 2 reviews the relevant literature on the recent developments in machining thin and deformable plates, clamped to the adjustable fixturing system. CHAPTER 3 provides methodological details of the present dissertation. It presents a framework in FEA (Abaqus software program)

to predict the deviation of thin-floor plates by applying an adaptable configuration setup. The proposed framework tries to anticipate the dimensional errors of the machined surface in the vertical direction. An experimental study with cutting conditions identical to those brought in the present chapter is offered in CHAPTER 4. In CHAPTER 5, the presented prediction model in Abaqus is improved considering two effective approaches namely modifying the cutting force model in Abaqus and the dynamic behavior of the part. Finally, the conclusions and recommendations for future studies are discussed in the conclusion section.

CHAPTER 2

LITRATURE REVIEW

2.1 Flexible fixturing system for machining low-rigidity components

Deformation in sheet metal is a serious problem in the aircraft and automotive industry. Most of the fixture designs are meant for rigid bodies such as "3-2-1" principle (Lowell, 1982). The numbers are the number of locators in each datum (primary, secondary, and tertiary). In order to minimize the deflection of the workpiece (normal to the primary plane or Z direction) during machining, an "N-2-1" (N ≥3) locating principle has been proposed (Cai et al., 1996). Finding the optimal "N" for the principle is the purpose of the study. "N" is the number of locators required in fixturing the sheet metal on the primary datum (restraining the workpiece in Z-direction).

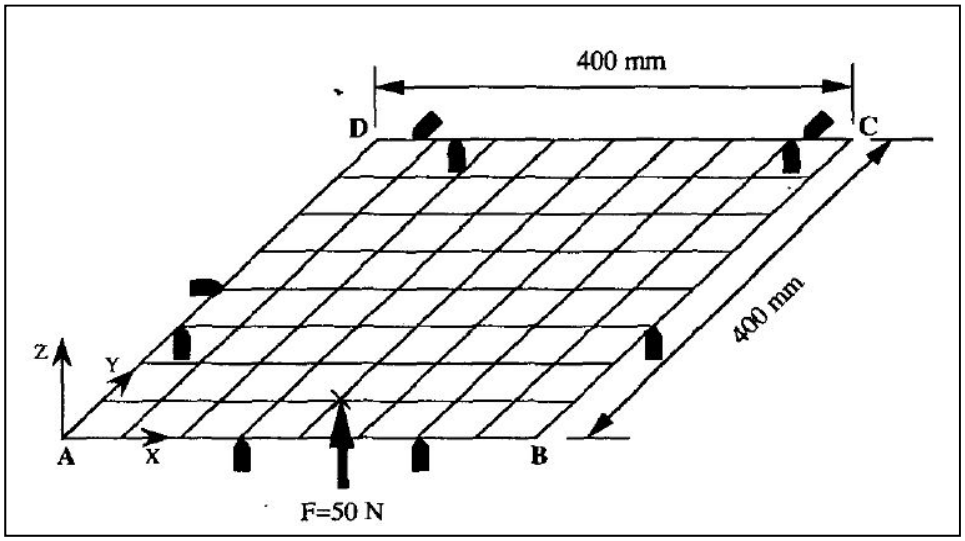


Figure 2-1 "N-2-1" Principle for locating the sheet metal, N=6
(Cai et al., 1996)

Moreover, the model also contains two locators on secondary and one on tertiary dimension which are sufficient to restrain motion of the workpiece in X-Y plane. The positions of locators in primary dimension are very important and not the number of it. The FEA and nonlinear programming technique are applied to analyze deformation and find the optimal fixture layout, respectively. The author was able to verify the model for sheet metal fixturing, theoretically and numerically. Optimal fixture layout design algorithms have been used to find the best fixture layout (positions of the locators) for the "N" locators. The results show a satisfactory reduction in deformation. However, his work lacks a few facts such as: knowing the proper distances between locator pins and constraining them to the workpiece.

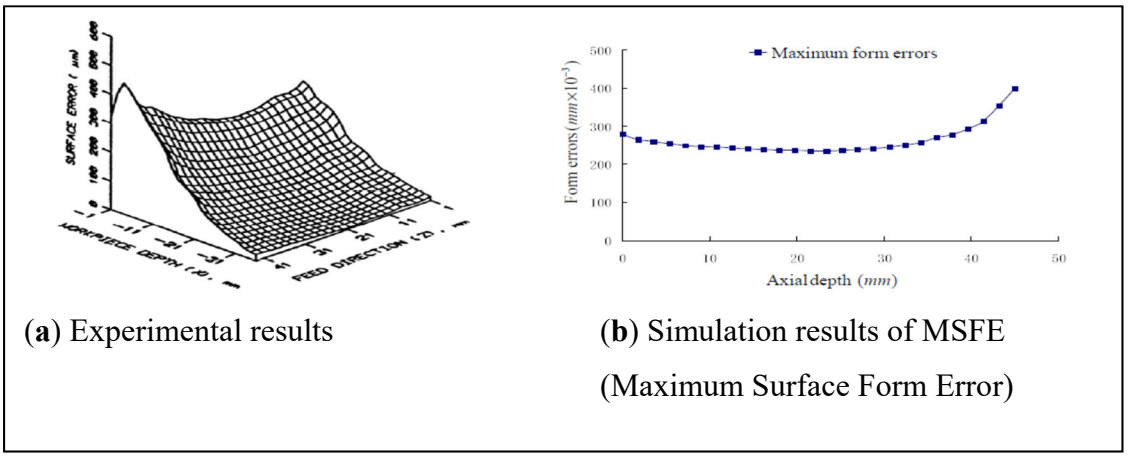
Prabhaharan et al. (2006) proposed a fixture layout optimization model which uses genetic algorithm (GA) and ant colony algorithm (ACA) separately and analyzes their performances on the workpiece deformation using FEM (Finite Element Method) to minimize the dimensional and form errors. Traditional fixture systems were designed with errors and in numerous cases, various factors were disregarded. For instance, an elastic body model for the fixture layout was investigated because a rigid body model does not predict the workpiece positional errors. Another example is the non-flexibility of the fixture which means the location of the fixturing elements are fixed and unchangeable. According to the authors, the position of fixturing elements (clamps, locator pins and support pads) in the setup is called fixture layout and optimal fixture layout is the layout that minimizes the deformation of the workpiece. To achieve this optimal layout, the authors used the GA and ACA algorithms to determine which one is more accurate. Therefore, their assumptions in finite element formulation are made as follows: "Workpiece is an elastic body whereas the fixturing elements are the rigid body; two-dimensional workpiece is discretized into triangular elements; and the response of the workpiece under the external load conditions is considered along the plane only". For each layout in each iteration with different machining loads, the maximum deformation is found. Then by using the GA and ACA, the minimum deformation among the maximum deformations is calculated. This procedure is repeated for all iterations. Two different methods were implemented to test and compare the performance of the GA and ACA (also known as method1 and method2). The first one has three locators and two clamps while the second one is defined

with three locators and three clamps to sustain the part in the fixture setup during the machining process. In their research, three node systems were applied. The optimal fixture layout for all three different node systems was found with applying the GA and ACA models, separately. The results reported that both GA and ACA can be applied in the fixture layout optimization problems. However, the objective function value and convergence of the ACA are closer to the optimal solution (the most minimum value among all possible layouts) than GA in both method1 and method2. The consistency in performance of ACA shows better results comparing to GA. Moreover, method2 proved that using 2-clamps on the part's edge is sufficient to reduce the component deformation. The whole point of designing the best fixture layout model is to have the workpiece under control and firm while it is attached to the fixture setup and being machined. Thus, any kind of vibration and flexibility during machining which cause instability of the process can interfere the purpose of the study.

2.2 Geometrical error prediction in machining low-rigidity components

Y. G. Kang et al. (2014) used a systematic simulation model suitable for surface errors prediction in peripheral milling of thin-walled components. These parts can be deformed easily during machining. Therefore, in order to meet high machining productivity and quality, the deformation must be under control. In the proposed procedure the deflection of the tool and the workpiece were predicted by using the cantilever beam theory and applying FEM, respectively. The total static deflection at a specific node in the FEM was calculated. By using Abaqus and defining all the cutting conditions in cutting forces modeling for a flexible tool-workpiece system, it is possible to calculate the instantaneous cutting forces acting on the tool in X and Y directions. In order to meet the static balance condition, an iteration algorithm is used to determine the chip load, cutting forces and deformation of the surface. A correction factor is defined in order to eliminate those portions of the flutes which are not in contact with the workpiece. Failing to do so results in calculating extra unreal forces and their influences on the tool, workpiece and chips which do not exist. This could eventually endanger internal validity of the proposed model. Based on the FIAL (flexible iterative algorithm) and DIAL (double iterative algorithm) algorithms, the maximum deformation can be calculated. By

controlling them, it is possible to have an acceptable surface finish. An experimental test is done to validate and verify the proposed method. According to Figure 2-2, the maximum surface errors increase when more material is removed. It also shows that the simulation time can be optimized by 50% with the new method. Nevertheless, how to compensate the predicted errors is still a problem which cannot be neglected. Tool path factor is also important to reduce the surface errors which was not considered in this study.



(**a**) Experimental results

(**b**) Simulation results of MSFE (Maximum Surface Form Error)

Figure 2-2 Results comparison
(Y. G. Kang et al., 2014)

Conventionally, for finding milling forces and important variables such as feed, depth of cut and cutting speed for each cutter, it was needed to run some test on the sample tool which was time-consuming and costly (mechanistic approach). It was even necessary to run more tests for complex tool geometries such as helical ball end mills. Budak (1996) considered a unified mechanics approach which determines a model in which it predicts the milling force coefficients and cutting forces for Titanium alloy, using orthogonal and oblique analyses in three directions for a tool. This can be also used for tools with more complex geometry. In the approach, for each particular tool-workpiece material combination, the author has modeled the elemental force components on the engaged tooth by considering it as orthogonal cutting condition in order to find cutting coefficients for two fundamental phenomena: edge force components and cutting components (Kte, Kre, Kae and Ktc, Krc, Kac respectively). By

finding these variables, scientists were able to develop a mathematical model to predict the force components for each tooth of a tool with any geometrical design. Comparing and verifying these values with the measured force coefficients from milling tests (experimental results) is the purpose of this paper. The results from both models show an acceptable variation. It means that this approach allows us to model an orthogonal cutting condition and using it to predict cutting forces coefficients in different oblique cutting operations rather than prepare a set of cutting tests for each tool geometry. According to the author's work, the contribution of edge force in Z-direction becomes more remarkable when the size of chip thickness is small. This achievement is useful in many aspects such as, helping operation planners to optimize the tool path without chatter and tool breakage and also investigating the performance of various designs for the geometry of the tools before the machining tests. However, there are still a few limitations in the geometrical design of the tools (rake angle, helical angle, cutter radius, etc.) and cutting conditions (feed rate, width of cut, axial depth of cut, etc.). For example, the model does not cover the full range of feed rate and high-speed cutting process has not been investigated in the calculations either.

During metal machining of a thin-walled workpiece, the geometrical shape can be influenced by mechanical, chemical and thermal effects. For the purpose of compensation and in order to predict the deviations caused by thermo-mechanical loads, a simulation model has been developed by Denkena, Schmidt, et Krüger (2010). According to the study, a structural component has been selected and machined from all sides. An experimental setup is designed to measure the part temperature and deformation. The experiment is done using two machining strategies to investigate both influences more accurately. Both strategies included roughing and finishing in peripheral end milling. In the first strategy, both operations take place subsequently and in the second one, a sufficient break time is considered between roughing and finishing which causes decreasing the temperature of the workpiece to the initial condition. By using a Finite element analysis simulation, the author modeled the same conditions as in the experimental model. A method is then developed for reconstruction of the process-induced shape deformations to predict the errors. The geometrical errors from two aforementioned

strategies were captured by a confocal microscope. There are 3 main deflections (see Figure 2-3):

1. Axial symmetric convex shape error which occurs on the top edge of workpiece.
2. A wavy profile generated by the machining path.
3. A profile along with the axial tool direction (because of the peripheral end milling operation with a large depth of cut).

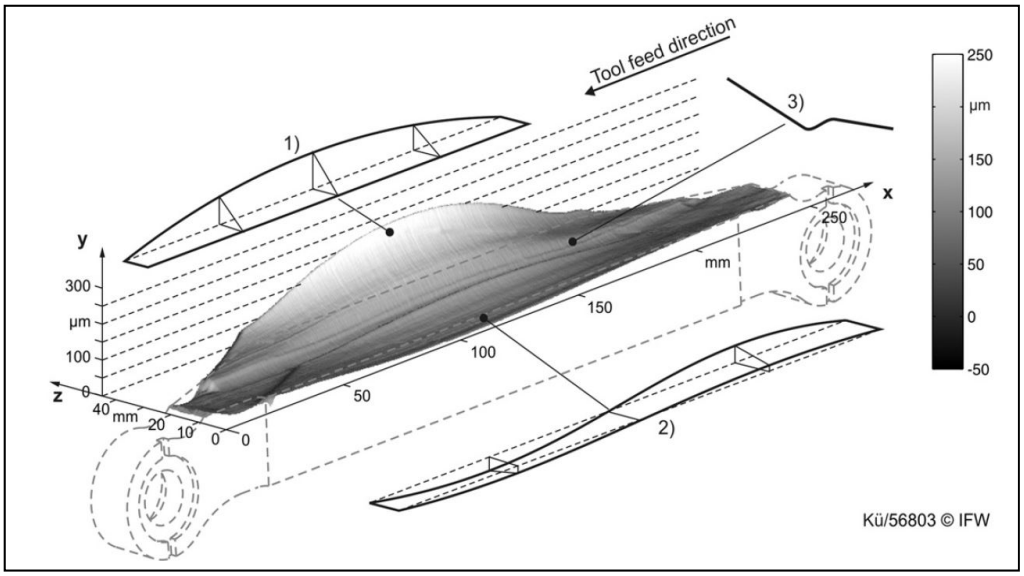


Figure 2-3 Three dominate shape errors
(Denkena et al., 2010)

A comparison of both results shows a significant reduction in error type 2 (waviness). There is no noticeable difference between the two results for the error type 1. And the heat generated during the finishing operation is three times higher than in roughing. The results show that the deformation caused by the heat acts against the mechanical force caused by cutting process. The heat deviation is positive while the force deviation is negative. A comparison of the experimental and simulation results shows that the only error that cannot be reconstructed is the deviation along the tool path, because it occurs locally. In order to identify the combined deviations (thermal and mechanical) and to investigate their influences on the final shape of the errors in milling operations, the deviation history of the milling was recorded and

reconstructed. Then, by help of the simulation program, the author could calculate the final geometry shape of the workpiece. The results from reconstructed model were satisfactorily close to the measured one. In order to reduce the heat in cutting edge, lubrication could be utilized. Based on the results, the break time can reduce the errors, then finding the optimal number of break times or using a cooler to facilitate the cooling process time, subsequently, can be investigated for the future works.

In another work, to reduce machining deformation and develop an error prediction method, an FEA model is developed by Ji et al. (2011). The paper focuses on an advanced FEA strategy for milling thin-walled parts to simulate the whole milling process as an orthogonal cutting. The entire milling test is divided into 6 steps. Step 1 is for the plunge rate (the feed rate when the tool has just started engaging) and step 6 stands for the retract rate of the tool (the feed rate when there is no engagement). The rest of the steps are for peripheral milling operation of the part. Two paths were considered while investigating such a milling process (see Figure 2-4). The maximum displacement for path-1 occurs at the middle of it and for path-2 it is located at the top (where the surface was in contact with the tool).

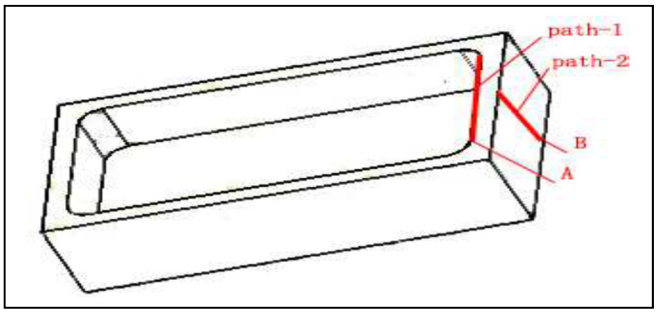


Figure 2-4 Two reference paths
(Ji et al., 2011)

Based on the force-time diagram, a slight vibration happens in the stable stage which is from step 2 to 5. In conclusion, the proposed method is for milling operation of thin-walled workpieces with a bottom. Based on the results, the higher feed rates increase the deformations. And controlling the feed rate at the corners is the best way to reduce machining errors in these

areas. The feed rates, vibrations, cutting forces and axial depth of cut are the limitations for this study. If the tool approaches too close to the bottom of the workpiece, it may damage and deform that area. Excessive vibration can result in tool damage or breakage and poor surface finish. Low feed rate may cause the operation to stop due to lack of sufficient force to cut the material. This study mentioned that the vibration is the reason of having unstable cutting forces in step 2 to step 5. And it is one of the reasons for generating shape errors. How to use FEA to reduce this negative effect on the surface finish can be focused on the future works.

In case of machining flexible parts (thin-walled components), previous studies investigated only on elastic distortion while the actual one occurs with a combination of both elastic and plastic deformations. Ma, Duan, et Tang (2010) present a theoretical elastic-plastic deformation equation model from Von Kármán, in which the boundary conditions of the cantilever cutter and the influence of bending springback have been taken into account. The springback term is defined as "the shape and size of the part changes in opposite direction of deformation because of the elastic behavior of the material in elastic deformation zone and partial recovery of material in plastic deformation zone."(Ma et al., 2010). There are 3 aspects regarding the boundary conditions of a thin-walled part for the combined deformation.

1. The bending function
2. The forces and displacements of middle plane
3. Combined behavior of bending and middle plane forces

By calculating the related variables in end milling of low-rigidity parts, now it is possible to measure the deformation of the part in all cutting positions differentially and before the springback action. Next step is calculating the springback value for each cutting position. Finally, the deformation values for each cutter position can be reached by knowing the mentioned values. A finite element model (ANSYS10.0) was developed to simulate the milling process according to the numerical calculations. Figure 2-5 displays a simulation flowchart process of elastic-plastic distortion for the thin-walled workpieces in a milling operation.

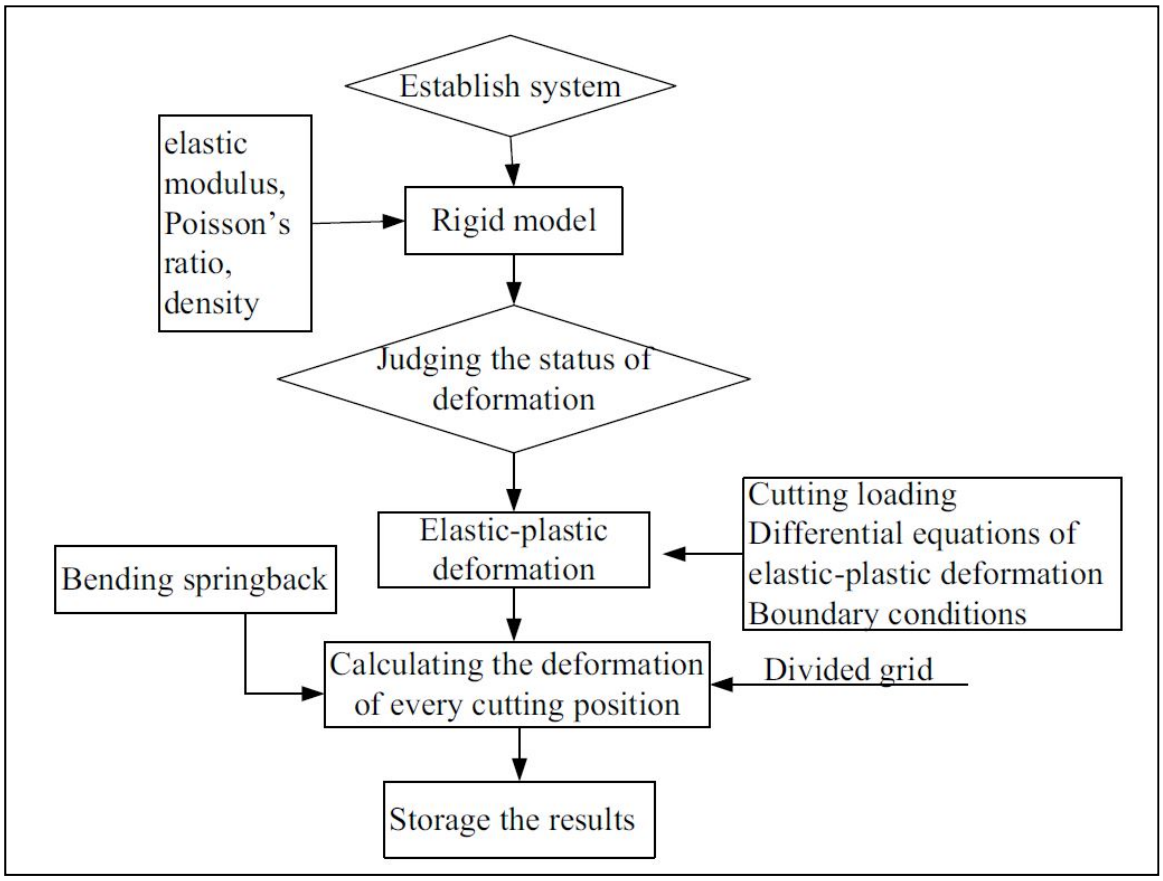


Figure 2-5 Flowchart of elastic-plastic deformation
(Ma et al., 2010)

This research provides a theoretical model to measure the elastic-plastic deformation of thin-walled parts during milling operation by considering the influence of springback on the flexible structure. The goal of the proposed model is to increase the knowledge over geometrical accuracy of milling thin-walled components, taking into account the springback circumstance. The deformation increases with increasing the distance of the cutting edge and the cantilevered part of the workpiece (the cutter). Then, in terms of depth of cut, an optimum length should be considered for end milling flexible components.

Most of the times, unitized monolithic metal structural components which are used in modern aerospace are machined with forgings to generate a thin-walled structure. The surface of these plates can be easily deformed due to poor stiffness and cutting forces and causes dimensional

surface error during the machining process. To predict and compensate this type of deformation, Izamshah R.A, Mo, et Ding (2010) used a technique in which the plate is machined several times with repetitive feeding to meet the desired depth and becomes calibrated manually. This technique requires a long time for processing, makes the productivity low and is expensive. Therefore, using a finite element analysis model is proposed to predict the deflection of the workpiece during end milling machining. Each node of the mesh is assumed the material particle during motion (The Lagrangian mesh). A thin-walled workpiece and helical fluted endmill are modeled with the 3-dimensional tetrahedron solid element. The reason of choosing this type of element is the complexity of the shape. The workpiece is modelled as a plastic body, so it is able to be deformed and cut. It must also be meshed finer than the tool because sometimes the generated chip can be very thin (see Figure 2-6).

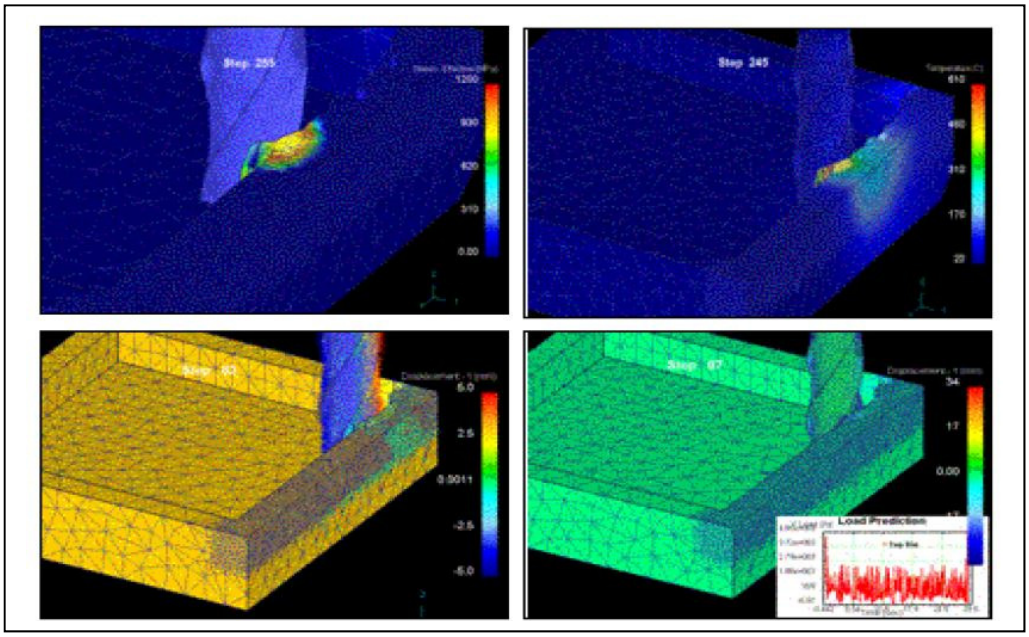


Figure 2-6 Simulation process of peripheral milling
(Izamshah R.A et al., 2010)

The bottom part of the workpiece is attached to the supports and constrained. For simplicity, in this simulation the tool is assumed as a rigid body thus, the tool deflection will not occur. The proposed model is experimentally tested in order to validate the accuracy of the model.

The results between simulation and experiment are found satisfactory. The deviation of the workpiece in both experiment and simulation results is at its lowest amount at the end of the part. And it is maximum when the tool is cutting the middle zone. The FEA model is a helpful and efficient tool for finding the root of errors during machining of thin-walled parts. It can help researchers to know about the deflection results and plan an error compensation model prior machining and without wasting any resources.

Y.-G. Kang et Wang (2013) considered two flexible iterative algorithms that are developed based on a theoretical approach of the chips being cut off the material. With the help of a simulation software program, a prediction of surface deflection in peripheral milling of thin-walled parts were analyzed. This study aims to decrease the simulation time and the algorithm errors. The work is partitioned in 4 main sections as followed.

1. A flexible Tool/Workpiece system is designed and the FEA is used to simulate the process and predict the deflection of the tool-workpiece set: It is assumed that the tool acts as a cantilevered beam.
2. The cutting forces for a flexible machining are modeled. The cutting forces and deflections caused by them must be considered to predict the surface form errors. The helical peripheral milling tool is divided to certain number of axial segments. In the cutting force model, the total forces for three X, Y and Z directions are measured which are acting on an instantaneous cutting edge. By calculating these forces, δYc and δYw -which are the tool and part deflections respectively-, can be measured.
3. FIAL (flexible iterative algorithm) is for the prediction of the actual cutting depth. To facilitate the process and precise the simulation, the author used FIAL algorithm. In contrast with RIAL (rigid iterative algorithm), in FIAL, the cutting contact length is not changed after the tool crosses the actual peak point of cut, then, no iterative step is needed until the end of the cut, because there is no cutting operation and, the tool and workpiece are not in contact (saving time by eliminating unnecessary processing). In the experiment, the deflection of TW (tool/workpiece) may cause the workpiece to be curved. Considering this deviation of the workpiece is important to find the nominal radial cutting depth in each increment.

4. DIAL (double iterative algorithm) on the other hand is for finding the maximum surface error and keeping that under control during the numerical process. There are 2 possible maximum surface errors (δYc, δYw), because the stiffness of the tool is very stronger than the workpiece, δYw is the maximum error. By help of DIAL the maximum surface error can be determined. The validation of the algorithms is demonstrated by comparing the numerical results with the experiment results.

In order to verify the validity of FIAL and DIAL, 2 tests with different cutting conditions have been implemented. The iterative number is the same for both tests. For the RIAL model, the more material is removed from the workpiece (test 1), the more prediction results (form errors and the position) go away from the measured method. But the results from FIAL model are close to the experimental results. The FIAL model simulation also shows close results to the experimental one in terms of cutting forces in contrast to the RIAL model. Therefore, the FIAL model is more accurate in determining cutting forces for flexible systems. And the DIAL model is more reliable to predict the maximum surface error for workpieces with any rigidity.

Reaching an accurate surface finish with the minimum part deflection in machining flexible and complex parts depends on the accuracy of CAD/CAM programs and how well they can define the optimal tool path and cutting strategies. But mostly, the profiles obtained from the machining and the predicted (simulated) one do not match. The reason is that the conventional force models are assumed as rigid cutting systems and do not consider factors such as tool-workpiece deflection and static-dynamic behavior of the operation together during machining. If these types of rigid forces are applied as an input for the deflection model (using FEA) while in the experiment they are actually flexible forces, it causes inaccurate results. Usually, in order to compensate the errors (the surface deviations), an NC simulation and expensive time-consuming tests are conducted. Ratchev's work presents a new method for prediction and compensation surface form errors caused by deflections in machining low-rigidity workpieces (Svetan Ratchev, Govender, Nikov, Phuah, & Tsiklos, 2003). Figure 2-7 shows the overview of the proposed model.

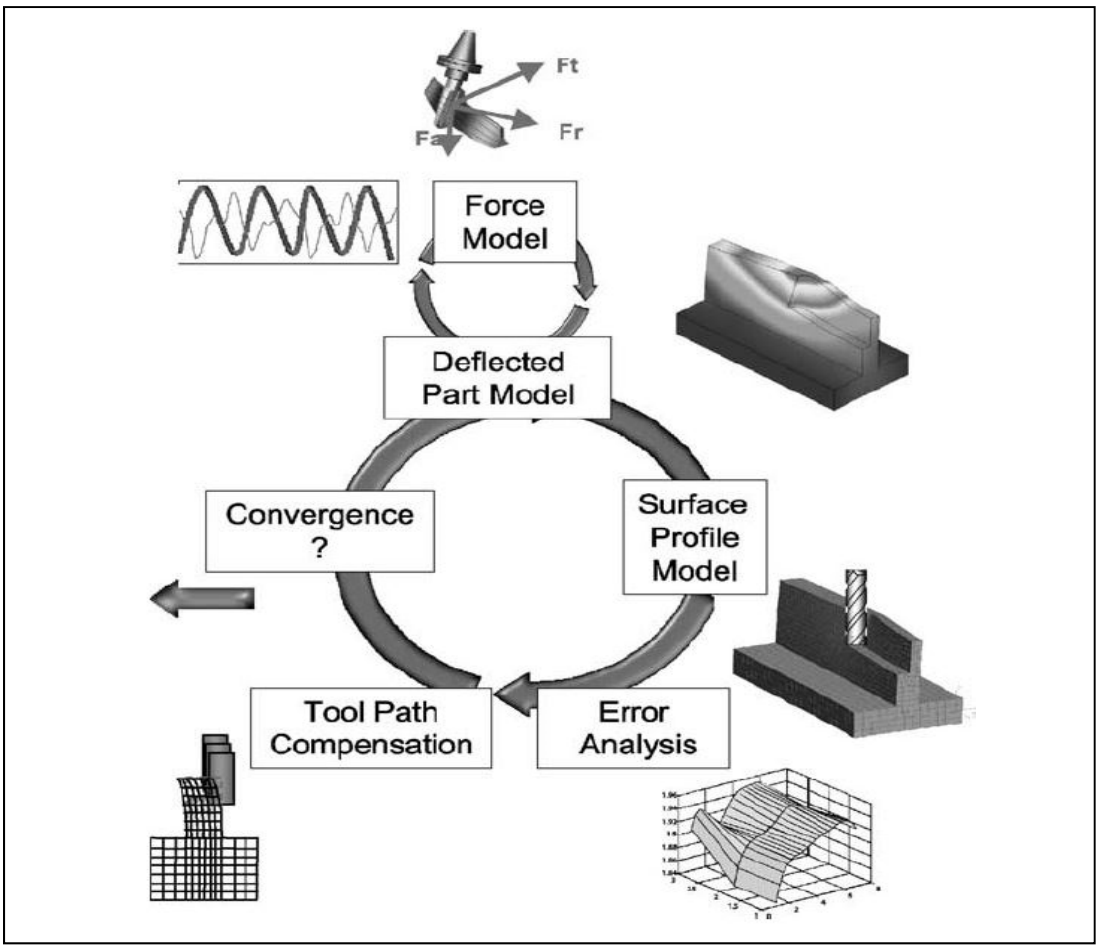


Figure 2-7 Methodology of prediction-compensation model of surface deflection (Svetan Ratchev et al., 2003)

The approach is based on using modified neural networks for the force prediction model and utilizing FEA for the part deflection model. On the other hand, a new voxel-based model (models based on three-dimensional volume elements for the purpose of shape generation) is proposed as a tool to design the process of material removal from the parts. The cutting forces are identified by using the neural network-based prediction model which has been designed only for low-rigidity parts. The predicted cutting forces are applied into an FEA model to predict the deflection of the part. The predicted deflected part then is used to determine the accurate material removal during machining which is the one close to reality instead of the

ideal one that results from static NC simulation program. To reduce the time of identification and prediction of the cutting forces and also decreasing the errors to an acceptable level, a genetic algorithm (GA) is introduced. It examines all the possible and expected geometries for the given data set. Later the user can decide which one is more suitable and all the input data is passed through that network. An iterative algorithm is used to analyze the errors and compensation. Consequently, corrective actions such as deflection control by changing the cutting parameters or deflection compensation by recalculating the tool path can be done. The force and deflection prediction models are experimentally validated by comparing the predicated with the measured profile surface in a few tests. With the help of force prediction algorithm, a new generation of NC verification method can be adopted in which the dynamic behavior of the tool and the workpiece are considered. Both force and deflection models were verified experimentally. However, the presented method is still dependent on manual optimization of the part and the simulation parameters.

Del Sol et al. (2019) in a review of available models and industrial approaches, investigated the dynamic and static problems of the machining process that are necessary to be controlled for the quality of the final part and avoiding the reprocessing steps. In the work, the authors presented analyzing machining different types of thin-walled light alloys, outlining numerical and computational models, revealing the reasons of instability and deformation in the models, and studying the solutions which are proposed by different authors. Figure 2-8 illustrates a flowchart for comprehensive machining process of thin-walled parts which addresses the dynamic and static problems of the machining operation.

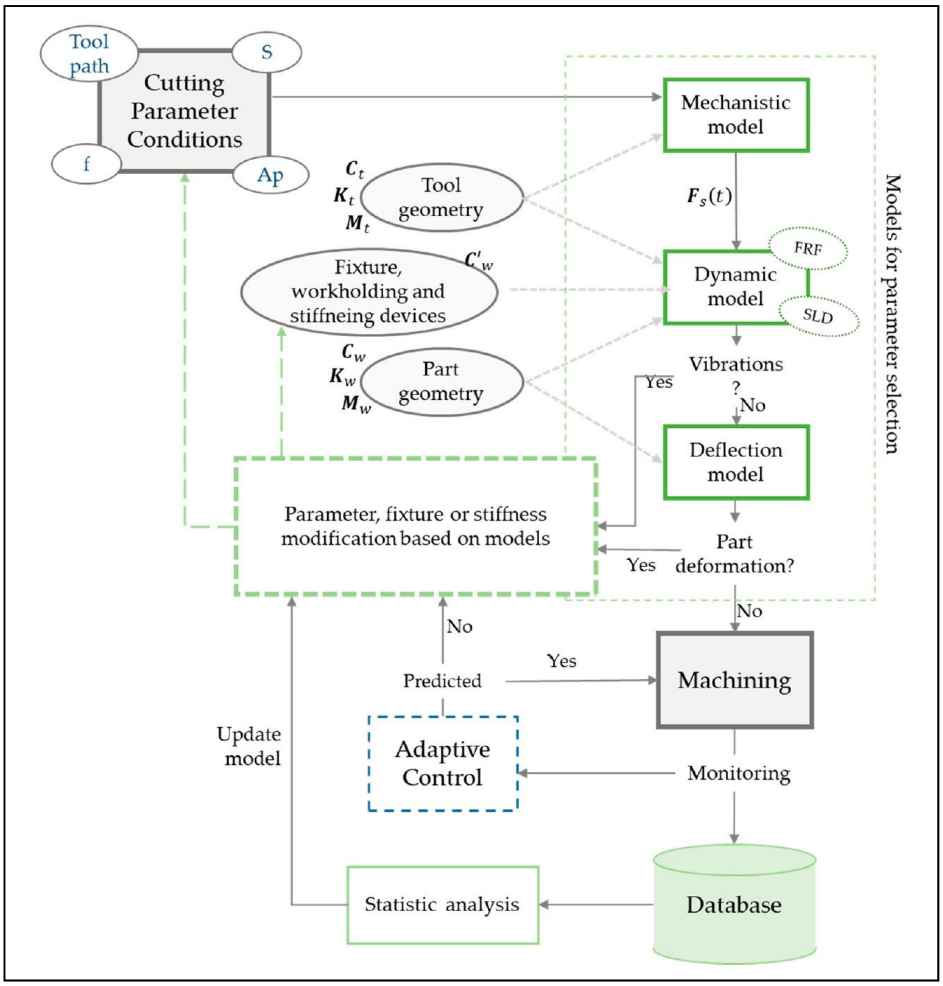


Figure 2-8 Thin-walled machining process flowchart
(Del Sol, Rivero, Lopez de Lacalle, & Gamez, 2019)

2.3 Error compensation model in machining thin components

In terms of predictions of deformation of low-rigidity parts when there is a milling process, most works and research have focused on single-cutting method without considering the fact that the deflection of previously machined layer of the workpiece affects the new one. Chen et al. (2009) presented a new methodology that predicts and compensates surface form errors caused by the deformation in different levels of a part with low rigidity. The author has mathematically developed an iterative formula to calculate the actual cutting depth. For a certain layer and cutting position, if the deformation results from FEA does not meet the

condition in formula (2.1), the FEA simulation will be repeated. This means, the cutting depth needs to be changed and subsequently new cutting forces will be generated and computed. This process goes on until the deformation results meet the condition in formula (2.1). The iterative termination formula is expressed as:

$$\left| \frac{{}^{v}\delta_i^{(j)} - {}^{(v-1)}\delta_i^{(j)}}{{}^{(v-1)}\delta_i^{(j)}} \right| < \varepsilon \tag{2.1}$$

Where, ${}^{v}\delta_i^{(j)}$ and ${}^{(v-1)}\delta_i^{(j)}$ are the computed deformations by the FEA program with DOC after v and $v-1$ times changes of position i in layer j respectively, ε refers to a very small and positive value, which can be determined based on the experimental situation in the machining operation. Once the condition in the formula is satisfactory, the related mesh elements are removed from the model (they represent the chips) and the next cutting position condition is checked until the machining operation is finished, and the deformation prediction process stops (see Figure 2-9). This paper considers only the deformation along the Z-direction (thickness direction) because it has a significant influence on the machining accuracy comparing to the other directions. For the deformation prediction model, a comparison between the predicted deformation and the measured one shows that the profiles are really close to one another. The difference of the two is noticeable where factors such as heat, and vibration were omitted in the FEA simulation. On the other hand, the deformation error compensation model is for decreasing surface errors resulted from the cutting forces. When the deformation errors are found, by comparing the nominal surface with the predicted one, it is feasible to modify a new tool path in order to compensate the deviation. Comparing to the full compensation model, the multilayer compensation method reduces the deflection of the surface finish for thin-walled components more effectively. The nominal cutting depth, cutting forces, spindle speed, vibration, and feed rate are the limitation parameters for this experiment. For the purpose of this paper, the author doesn't mean to exceed the limits which can lead to damage the tool or making the cutting process stop. In this study, only a specific type of milling tool was applied

(a single-tooth end) which does not involve all the cases in industry (multi-toothed tool). More teeth mean more cutting forces and more deformation. Therefore, a further investigation is required.

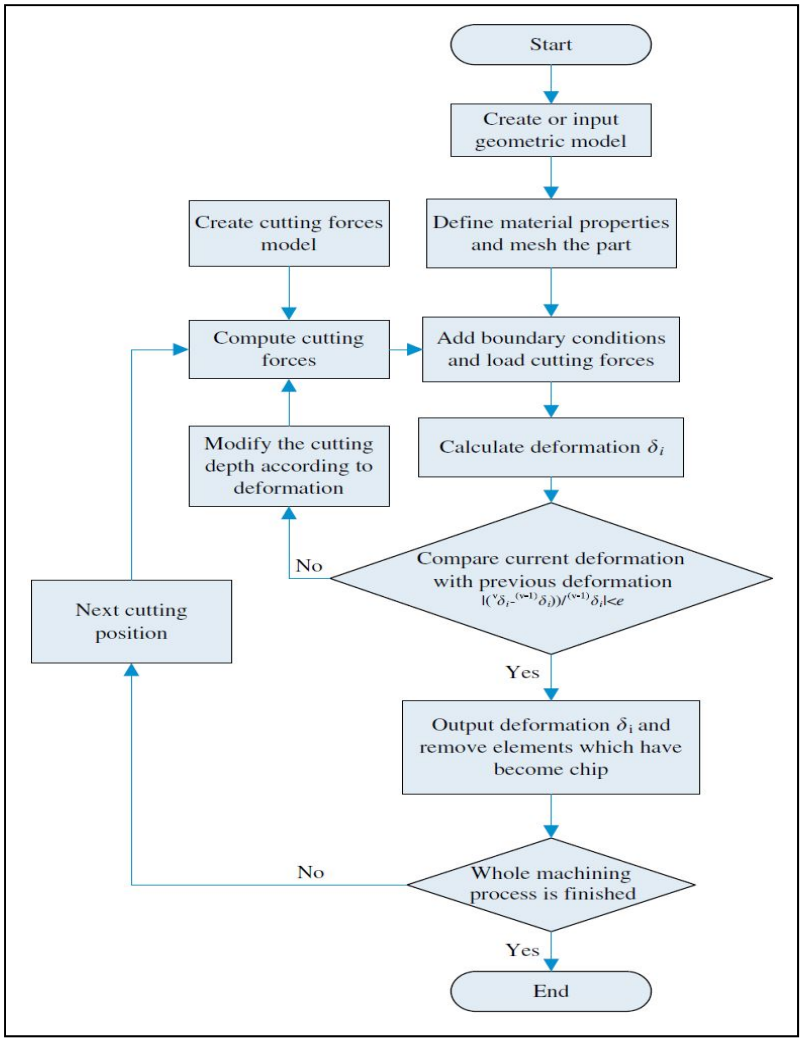


Figure 2-9 A multi-level optimization method
(Chen, Xue, Tang, Chen, & Qu, 2009)

In another work, S. Ratchev et al. (2005) presented an error compensation scheme to increase the accuracy of the machined components specifically targeting the challenging areas of low-rigidity complex parts. It reports on an advanced error prediction and compensation strategy addressing force-induced errors in machining of thin-wall structures. It predicts the machining

surface form errors by the use of a theoretical flexible force-deflection method. The predicted errors are then compensated by finding the optimal tool path prior to the actual machining experiment. The paper focuses on low-rigidity parts where force-induced deflection occurs. This issue is not considered by most of the existing machining simulation and optimization systems. The work is based on modeling and predicting the cutting forces, analyzing and predicting the deflection of the part during machining. The results for dimensional errors along with the related compensation model will be investigated then. To do so, the author developed a flexible theoretical force prediction model and then, used an iterative algorithm integrating an extended perfect plastic layer force model with a finite element part deflection model. Feeding the cutting force to the finite element model is the next step to predict the part's dynamic behavior when the cutting takes place. The milling error is computed iteratively using a trial-and-error method for the cutting force and deflection. This predicts a deflected part profile which is used to determine the real volume of the removed material during machining as opposed to the ideal volume that is currently implemented in adopting static NC simulations. When there is deflection in a milling operation, a compensation method can be considered which in this case, it is an optimization for the tool path. The simulation of the error compensation scheme is carried out by NC verification tools and is experimentally validated. The paper succeeds to predict the overall errors in flexible milling, using the proposed flexible force-deflection model with high accuracy. It also points it out that a large portion of the errors can be reduced by the proposed compensation method. It is worth noting that, although the suggested error compensation method highly reduces the cost of the milling process, a high computational power is required for modeling and extensive compensation iterations. Another aspect of this work which can be improved, is the use of a single-level compensation strategy. If a multi-level optimization approach could be implemented, it is feasible to reduce the errors even further.

Another research in this area has been done by Rai et Xirouchakis (2008). In high-speed milling of thin-walled parts, the main reasons of geometrical and dimensional deflections of the workpieces are the movement and the elastic-plastic deformation of the tool and workpiece which result from the selected tool path, fixture layout design and cutting variables. Available

software programs only can provide the tool-path simulation and do not take the dynamical effects of the part into account. Having a developed system by which simulating the whole process of machining and analyzing the causes of the errors is possible, helps to improve the quality of the machined part. This study proposes a finite element method (FEM) with a few related tools for milling process of thin-walled structures which considers the influences of fixturing, tool-path selection, operation sequence (the order and number of operations that are made) and cutting parameters in order to predict the deflection and elastic-plastic deformation of the part. The author took 4 main modules into account: 1. feature-based process-planner, 2. machining load computation module, 3. the FEM based milling simulation module and 4. the report generator. The machining feature characteristics such as operation type and tool path etc. are extracted from a file or given manually. Next, the created file is applied as an input to the force computation module where the cutting loads including cutting force components and the heat distribution, are calculated, and are saved in an ANSYS format file. Now the FEM module can extract the cutting attributes in the ANSYS file and make it possible for the user to create the fixture-workpiece model with applied boundary conditions for the setup. Then, the report generator module creates a hyper text markup language (HTML) report. For validating the numerical results such as the cutting forces, workpiece temperature distributor, part deflection and part residual stresses, an experimental analysis is carried out to measure theses values. To validate of the developed FEM model a few simple tests (pocket, slot, step) were considered to simplify the experimental setup design. By observing the results in terms of the thermo-mechanical response of the test part during machining of the pocket, the predicted and measured temperature results show similarities. The deflection profiles (measured and predicted) show that each axial pass that the tool takes, the amplitude of deflection increases. This is because of three reasons:

1. Elastic behavior of the workpiece
2. The rigidity of the workpiece decreases
3. Thermal growth situation during dry machining

Both measured and predicted profiles are close with a little gap over the entire cutting duration which results from the mesh size of the workpiece. By decreasing the size, it can be compensated partly. The finite element model can be the best available tool for analyzing the

root of the causes of the induced errors during machining of thin-walled structures. Knowing the results and predicting them before the actual machining allow manufacturers to improve their productivities and to skip the trial machining tests on the parts which are meant for designing the appropriate machining setup. In this work, due to the mass of large number of sensor cables, the fixture becomes instable and that is one of the main reasons for the changes in the wall thickness of the part. This plays an important factor which should be resolved in the future.

Typically, the surface dimensional error occurs when there is deflection in the tool and the workpiece during machining which affects the uncut and cut chip thicknesses. It may decrease or increase their values. To solve this problem, researchers developed another finite-element method for an end milling cutting-force model which analyzes surface dimensional errors of peripheral milling of thin-walled workpieces (Tsai & Liao, 1999). The design simulates the geometry and structural behavior of the cutter with the pre-twisted Timoshenko beam element to model the helical fluted end mill. The authors applied the modified Newton-Raphson method to solve the cutting system deflection and the cutting force distribution which is determined through an iteration algorithm. The thin-walled workpiece is modeled with a three-dimensional 12-node isoparametric element which can estimate the surface profile of the tool-part contact zone better and determine the changes of the workpiece thickness (Figure 2-10). For static equilibrium condition, a model is applied (the modified Newton-Raphson iteration) to determine the chip load, cutting forces and surface geometrical errors.

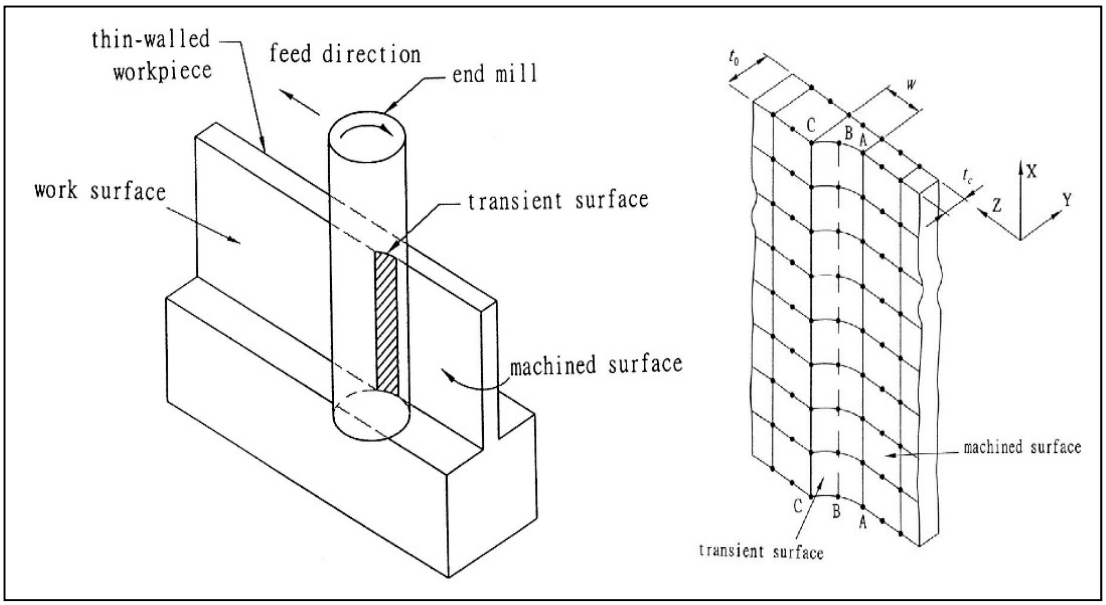


Figure 2-10 The peripheral down-milling of a thin-walled part
(Tsai & Liao, 1999)

The cutting force model follows the following assumptions; The chip thickness varies along the axis of the tool because of the helical shape of the tool. Thus, the cutting edge is divided to many small equidistant portions for each axial section. By summing up all the cutting forces for each section which is in contact to the workpiece, the instantaneous cutting forces acting on the whole tool can be calculated. These values are then used to compute the deflection of the tool and the workpiece. The model is considered as a rigid cutting model if their deflections (the tool and the workpiece) are ignored; and it is a flexible machining model, if it is otherwise. According to the author's assumptions, in order to predict the surface form error, it is needed to take the tool and workpiece deflections into account in only 2 directions (disregarding the tool's axial direction), because all the deflections are in these two directions. The machined surface form error can be achieved when the deflection of the tool and workpiece deformation are calculated by a simulation model. Since the forces on the tool and the workpiece are equal but in opposite direction, then the part must face an undercut or overcut material removal. Because it is assumed that the system is flexible, the boundary conditions must be changed. The reason behind is that the rigid cutting force model was used to model these forces, hence,

the current cutting force distribution does not include the tool-workpiece deflections. A new cutting force distribution is calculated through the corrected immersion boundaries and then, the CAE model reaches to a solution for the tool and workpiece deflections. To verify the proposed model and its capability of predicting the surface form errors caused by tool-workpiece deflection, three experiments were carried out and the results were compared to the simulation model. In the tests, the author somehow managed to apply a one-tooth cutter to eliminate the effect of the tool's run-out. The surface error distributions in the measured (experimental) and predicted models are showing close profiles. Since the deflection of the part at the central area is smaller than that at the free edges (the bottom of the workpiece is fixed), less errors can be observed around that area. However, the surface errors along the free edge of the final feed station are larger than those in other feed stations. This occurs because as the tool feeds to the final feed station, the thickness of the workpiece is reduced and as a result the workpiece becomes more resilient. By comparing the rigid and flexible models, the outcome reports that as the thickness of the workpiece becomes thinner, the results for the rigid model become worse in contrast to the flexible model which shows more aligned results to the experimental data. The dimensional errors achieved from the flexible model is smaller than the experimental one and the difference between them becomes larger for thinner workpieces. The present cutting force model for a rigid body is able to predict the results with a good accuracy if the workpiece is thick and does not deform seriously during milling. The proposed cutting force model for flexible body is similar to the rigid cutting force model with a difference of which the tool-part deflections influence is considered. To reduce the surface dimensional error, avoiding the tool and workpiece deflections is necessary by selecting appropriate cutting conditions and clamping type. The existent eccentricity between the axes of the tool and spindle results in a tool run-out (causes overcut) which affects the cutting forces. This effect was not considered in the investigation. To reduce the run-out effect, one of the two flutes of the helical end mill is removed to some extent so that only one flute will cut the workpiece.

In other studies, the usual approach to solve geometrical errors in machining low-rigidity parts is playing with the cutting conditions values. For instance, reducing the depth of cut or feed rate and increasing the number of teeth. However, the values manipulation results in extra costs

and longer machining time for the company. Researchers reported on an in-process compensation strategy by using a piezoelectric application (Diez et al., 2015). Piezoelectric actuators are known as devices with the ability of providing high movements time ratio in the nano-metric range. This application is based on comparing the measured and nominal cutting forces under the same cutting conditions obtained from rigid machining of thin-walled parts. The data collected from this comparison then is used to produce a motion command signal to the piezoelectric actuator with the aim of making a slight change in the width of cut that will eventually compensate the deviation caused by the workpiece deformation. To validate the work, the authors did the following: First, a reference model is needed to allow an accurate and fast calculations of the cutting forces. The authors' model is based on the average chip thickness which does not require an iteration method to calculate the cutting forces. Next, the influence of the part deformation on the cutting forces is analyzed. Based on the research, the deformation that occurs in milling thin-walled components is from the workpiece when the ratio between the tool diameter and the wall thickness is high. This deformation changes the radial depth of cut which affects the chips (generating smaller chips) and the entry angle of the cutting edge. Consequently, the cutting forces are changed. After estimating the cutting forces acting on the cutting edges based on the average chip thickness, the determination of the cutting coefficients must be done. Then, the model was verified experimentally, in other words, the estimated cutting forces calculated from the reference model must be in a good agreement with the measured ones in the experiment. The next step is to compensate the deviation caused by the deformation. This compensation system is according to the changes in the radial depth of cut (a_e) by changing its value (lower axial depth of cut results in lower cutting force Fy) to the point that the cutting force magnitudes match with the numbers in the nominal cutting conditions for the rigid machining. The in-process calculation for the compensation model is based on Figure 2-11.

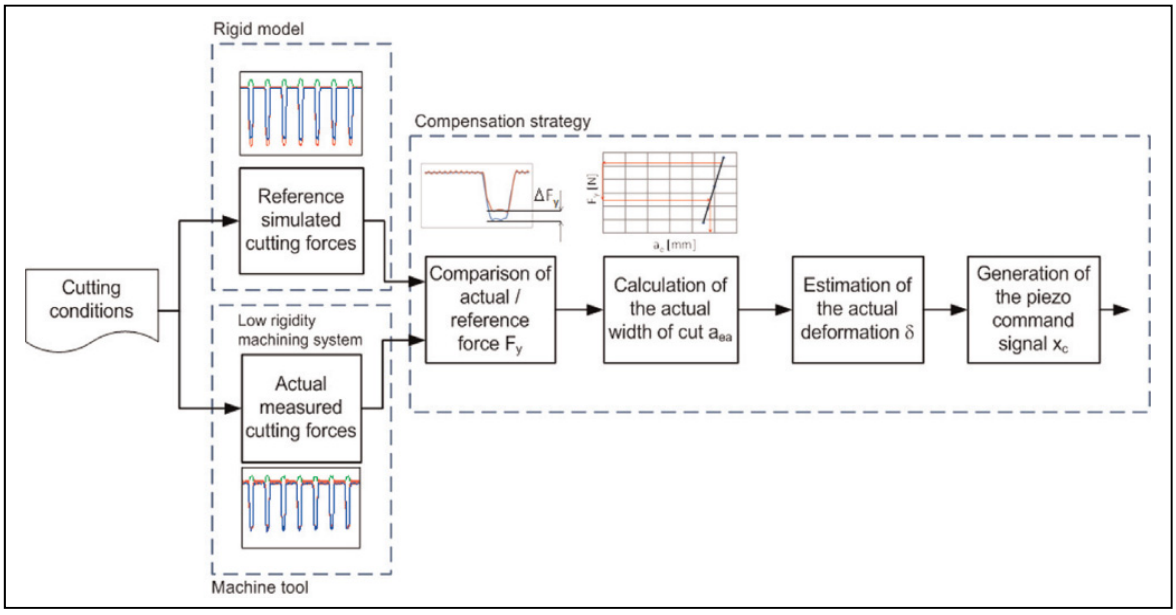


Figure 2-11 System to compensate for workpiece deformation
Taken from Diez, Perez, Marquez, et Vizan (2015)

In the machining process when the deformation happens for the workpiece, the axial depth of cut decreases and as a result, the cutting forces will decrease. The compensation strategy is about comparing the measured cutting forces for the current operation with the estimated cutting forces from the proposed model. Once they are found, then the actual width of cut and consequently the estimation of actual deformation will be obtained. By the help of piezoelectric actuator, the proper motion of the tool will be generated to fulfill the desired depth of cut. And finally, this strategy has been experimentally approved with a small difference between the deformation values. In end milling operations this control system, which is based on a reference model, works perfectly for dimensional errors caused by the part deformation. By applying the presented model, there is no need to use the iteration method along the flutes. The estimated forces must be accurate otherwise the compensation will not be effective. In this approach a piezoelectric actuator has been applied which is able to modify the relative motions between the tool and the workpiece to compensate the changes in depth of cut which causes the deformation. The system is independent of the machine tool control system, it does not manipulate the tool-path program and works parallelly with it.

In a recent work, slotting thin components using a flexible setup was investigated (Nguyen & Chatelain, 2014). These thin plates are difficult to clamp and machine due to the low rigidity (flexibility) and complex geometry. Attaching these flexible structures on the working table itself is another issue which should be focused on. Most of the setups' layouts, in spite of having costly processes, are dedicated and are not able to be applied for different parts with different shapes. In the study, a method for milling thin plates which are fixed into an adjustable fixturing setup is proposed. An end mill tool and aluminum plates were used in a few experimental tests. In contrast to the conventional full backplate support, the workpiece is sustained by four location pins on a flexible testbed setup. These pins go through four holes which were drilled before and are located in the corners of the plate. Four bolts then will be screwed on the top of the pins to hold the structure on the working table. During the machining tests, the axial cutting forces and the displacement of center point of the back face of the plate are measured. The machining process aims to create a rectangular pocket with certain dimensions. The proposed tool path follows a spiral curve which starts from the center of the part and gradually moves towards the external edges (see Figure 2-12). The more material is removed, the more the part tends to be deformed due to induced cutting forces. Therefore, the spiral path begins at the center when the part is at its stiffest condition.

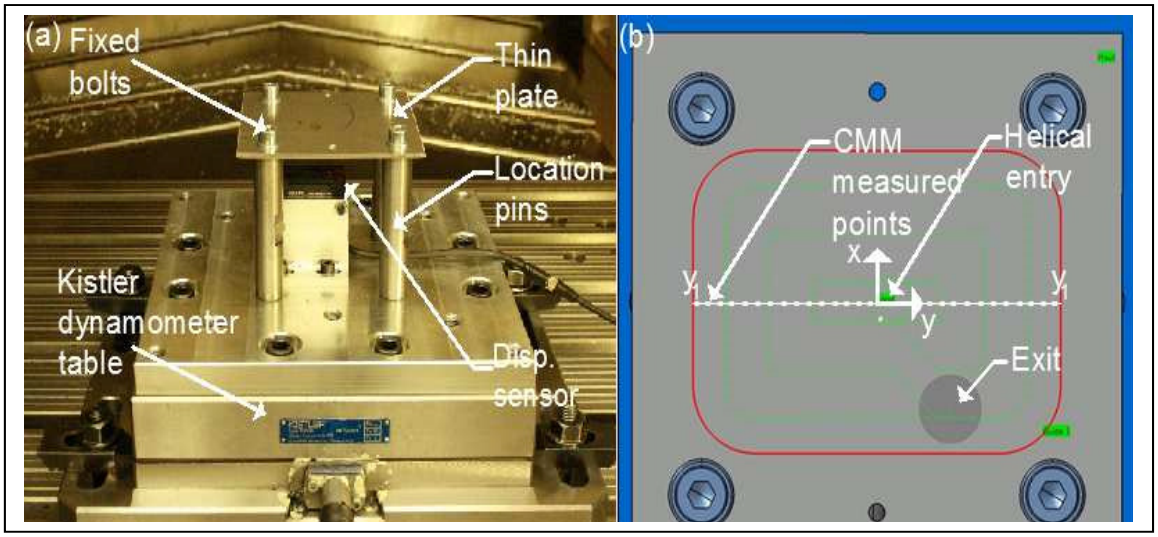


Figure 2-12 The machining setup and the spiral tool path
(Nguyen & Chatelain, 2014)

When machining is done, the thickness of the machined face was measured along the y-axis. The displacement of central point and the cutting forces were monitored during machining. The maximum deformation occurs at the beginning of the cutting operation (the cutter is positioned in the center) which the workpiece is at its most flexibility and the magnitude of generated cutting forces are high comparatively. By advancing the tool through the workpiece towards the positions of the pins, the displacement values decreased to zero approximately. Near to the support pins, the part's rigidity is at maximum, because the pins can bear all the cutting forces and the part does not deform much. Three tests were conducted with different cutting conditions to determine the performance and correlation between the cutting force, the displacement, and the cutting deflection. The cutting deviations resulted from the cutting forces in Z-direction, and it has direct relation with the axial cutting forces. The author concluded that in milling thin plates using a flexible setup configuration, geometrical errors cannot be avoided, and its magnitude depends on the position of the tool and the location of the pins during the material removal process. The desired tool path for different geometries can be designed in software programs and the results can be monitored faster. This approach suffers from a fact that a practical prediction model contains many limitations such as longer processing time and implementing extra resources.

In another research from the same authors, a simple slotting operation in a straight line was studied using the same flexible setup configuration as seen in Figure 2-13 (Nguyen, 2016). An FEA prediction model (Abaqus) in this work simplifies the process into a static analysis and computes the deformation of the part in several positions of the cutter on the part. For each position, a CAD file of the part is created and imported into Abaqus for analysis. Consequently, various analyses were made in Abaqus to obtain the displacement results.

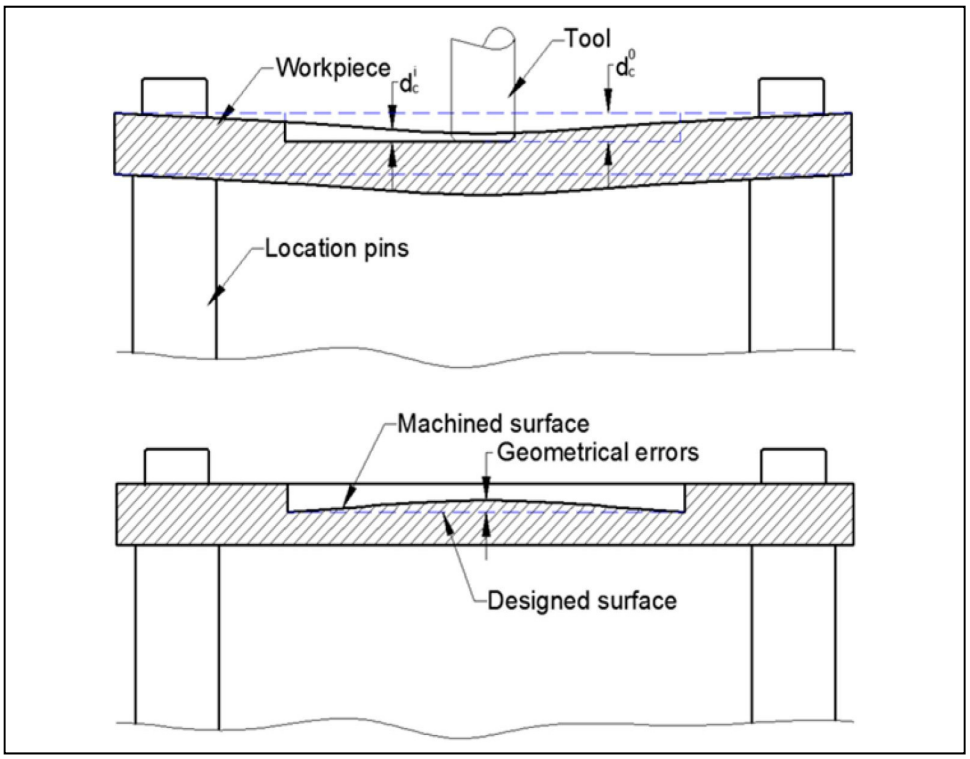


Figure 2-13 Material deformation during machining and geometrical error after machining

(Nguyen, 2016)

Given the fact the tool cuts less material than expected (undercut) due to the deformation, the part then was compensated by applying a method called mirror technique as shown in Figure 2-14. This method defines a new tool path in X-Z plane (vertical plane) with respect to the current deviation from the prediction model. In other words, it upgrades the depths of cut through the trajectory for different positions of the tool to achieve a flat surface finish as a result.

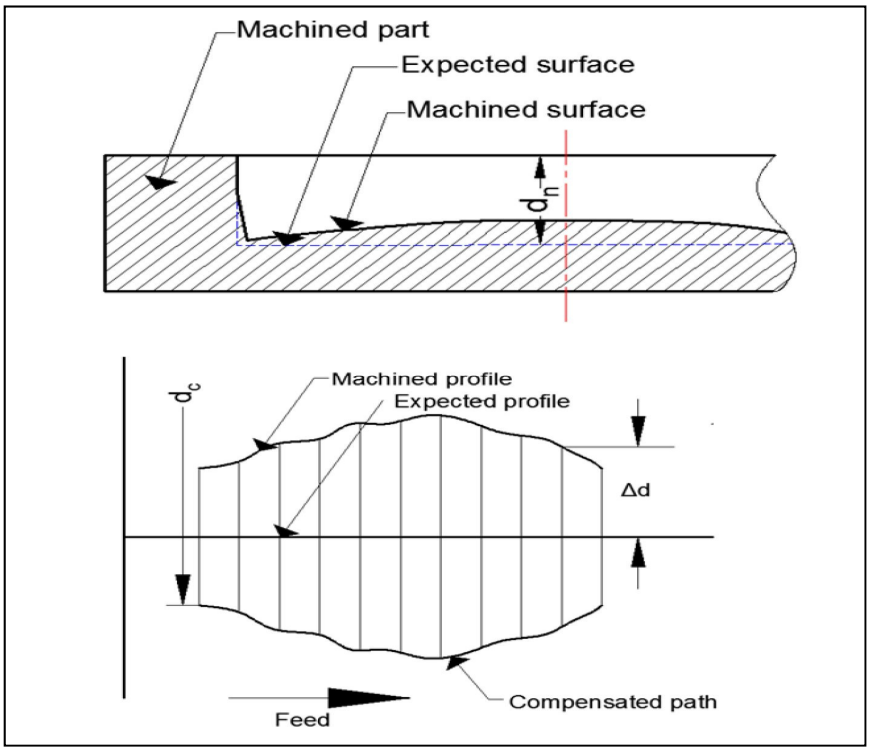


Figure 2-14 Mirror technique
(Nguyen, 2016)

Since the geometrical errors cannot be avoided in machining with the aforementioned conditions, thus, by reselecting the optimal tool-path profile based on the workpiece deformation for the compensation model in FEA, it is possible to reach the expected surface finish with a very low permissible tolerance. However, creating such a model in FEA requires iterative algorithms in both preprocessing and postprocessing that work parallelly with the program. To be able to apply these algorithms, the need of a comprehensive framework is essential. The present dissertation addresses this very important problem.

2.4 Summary of literature review

To summarize the previous works which have been done so far by the researchers, we created a table which points out the important aspects of their research projects as seen in Figure 2-15 and Figure 2-16 below.

Study Area	Reference	Description	Advantages	Limitations
Flexible fixtruing systems for machining thin parts	Cai et al. (1996)	Proposed an "N-2-1" locating principle for sheet metal fixturing	1. Valid for sheet metal fixturing	1. Finding the optimum distance between locators
			2. Finding optimum value for "N"	2. Constraining the locators to the workpiece
			3. Using OFixDesign software with ssatisfactory results	
	Prabhaharan et al. (2006)	Proposed a fixture layout optimization model using GA and ACA algorithms	1. Minimizing the dimensional and form errors	1. Vibration
			2. ACA showed better results than GA	
Geometrical error prediction models in machining thin parts	Y. G. Kang et al. (2014)	Applied a systematic simulation model in FEM, suitable for surface form errors prediction in peripheral milling considering tool/workpiece deflections	1. Modeling cuttig forces actin on the tool in Abaqus	1. Tool path can be modified
			2. Optimizing simulation time by 50%	
	Budak (1996)	Predicted the milling force coefficients and cutting forces usining orthogonal and oblique cutting conditions	1. Cutting coefficients can be used to calculate cutting forces for different machining types	1. Geometrical design of the tool which affect the cuttin forces, were neglected such as: rake face, helical angle, cutter radius, etc
			2. Helping to optimize the tool-path without chatter and tool breakage	
			3. Investigating the performance of various designs for the tool's geometry	2. Different cutting conditions (feedrate, cutting speed, etc)
	Denkena, Schmidt, et Krüger (2010)	Developed a simulation model to predict the thermo-mechanical loads in machining thin parts	1. Machining strategies include roughing and finishing in peripheral end-milling	1. Form deviation error along the tool-path (dimensional error)
			2. Waviness error reduction	
			3. The deviation history was recorded and	2. Lubrication to reduce the heat
	Ji et al. (2011)	A prediction model (FEM) for milling thin-walled part	1. Error prediction model	1. Assumption for a rigid tool
			2. Reducing machining deformation	
			3. Johnson-Cook material law for plastic behavior	
			4. Damage law for the elements in FEM	2. Simulating vibration in FEM was neglected
			5. Study on the influcence of feed rates	
			6. Higher productivity & reduction in material waste	
	Ma, Duan, et Tang (2010)	Studying on machining flexible thin parts considering the elastic-plastic deformation and the influence of bending springback in FEM	1. Calculating the deformation with & without springback	1. Tool's deflection
				2. Vibration
				3. Heat, coolant
	Izamshah R.A, Mo, et Ding (2010)	Creating a prediction model with FEA using Lagrangian mesh for milling thin-walled structures.	1. The part is modeled as an elastic-plastic body	1. Long processing time
			2. Helical end mill	2. Being expensive
				3. Lower productivity
	Y.-G. Kang et Wang (2013)	Studying on 2 flexible iterative algorithms for removing chips action	1. Using FEA to predict the surface deflection in peripheral milling of thin-walled parts	1. Other factors in cutting conditions were neglected in FEA such as: heat, vibration & risidual stress
			2. Decreasing the simulation time	
			3. flexible Tool/Workpiece system	
			4. FIAL model = good for determining cutting forces for flexible systems	
			5. DIAL model = good for predicting the max surface error for rigid parts	
	Svetan Ratchev, Govender, Nikov, Phuah, & Tsiklos (2003)	A method for prediction and compensation surface form errors caused by deflection in machining thin parts	1. Tool-workpiece deflection	1. Network neurons are more responsive when there is higher DOC
			2. Static-dynamic behavior	
			3. Flexible forces	
			4. Using the neural network-based prediction model	
			5. Using an artificial neural network	
	Del Sol et al. (2019)	A review of available models in the industry that investigates the dynamic & static problems of machining processes influencing on gemetrical errors	1. avoiding reprocessing steps	N/A
			2. Machining different types of thin light alloys	
			3. Instability and deformation causes	

Figure 2-15 Summary of literature review, page 1 of 2

Error compensation model in machining thin parts	Tsai & Liao (1999)	Developing an FEM to find the surface dimensional errors of perpheral milling of thin-walled workpieces using pre-twisted Timoshenko beam element to model the tool	1. Applying the modified Newton-Raphson iteration to solve the cutting deflection and the force distribution	1. Neglecting the deflection in the tool's axial direction (due to small value comparatively)
			2. Instantaneous cutting force model for the tool	
			3. Considering the deflections of the tool and workpiece	2. One-tooth cutter
			4. Undercut or overcut	3. Tool run-out
	Chen et al. (2009)	A prediction-compensation method for surface form errors caused by the deformation in different levels of a thin part	1. Iterative formula to find the actual cutting depth	1. Heat & vibration were omitted in the FEA
	S. Ratchev et al. (2005)	A prediction-compensation strategy addressing force-induced errors in machining thin-component	1. Predicts the machined surface errors by a flexible theoretical force-deflection method	1. Single-level compensation strategy
			2. Using an iterative algorithm integrating an extended perfect plastic layer force model	
	Rai et Xirouchakis (2008)	The authors developed an FEM to predict the deflection and elastic-plastic deformation of a thin-walled part in high-speed milling.	1. Using ANSYS to take the elastic-plastic deformation and the movements of the tool and the part into account	1. Large mesh size
	Diez et al. (2015)	In-process compensation strategy using piezoelectric actuators. The application is based on comparing the measured and nominal cutting forces under the same cutting	1. The compensation model is based on the changes in	1. Considering the average chip thickness
			2. Proper motion of the tool is generated during the operation without changing the tool-path	2. Lack of iteration algorithm
	Nguyen & Chatelain (2014)	Slotting thin components using a flexible configurable setup to find the best strategy for creating a pocket	1. Spiral tool path, beginning from the center is the best option	1. Longer processing time (a practical prediction model)
			2. Deformation in Z-direction (axial)	2. Implementing extra resources
	Nguyen (2016)	Prediction-compensation model for milling thin-components using flexible configurable setup	1. Static approach in FEA (Abaqus)	1. Too many CAD/CAE files
			2. Feeding CAD files to Abaqus	2. Lack of iteration algorithm in preprocessing and post-processing in Abaqus
			3. Deformation in Z-direction (axial)	3. Time-consuming
			4. Mirror technique	4. One-attempt compensation model

Figure 2-16 Summary of literature review, page 2 of 2

CHAPTER 3

PROBLEM DEFINITION AND DYNAMIC SIMULATION (IMPLICIT ANALYSIS)

3.1 Problem definition

As mentioned before, for thin-floor components, it is difficult to maintain dimensional accuracy and straightness, due to their low stiffness. In spite of the fact that several aspects contribute to the precision of the process, some simple guidelines could be taken into account. In various research projects, the prediction of the machined surface has been investigated as mentioned in the literature review section. This can help the manufacturer in terms of the productivity and finance. In this chapter, a method is introduced to predict the deformation and consequently the cutting deviation of low-rigidity components during a slot milling operation, using a flexible setup configuration. This setup partially supports the part in contrast to the traditional one which the bottom face of the plate is fully supported to a fixture and cannot be deformed vertically. To understand the benefits of the commercially available flexible setups in the industry, a simplified testbed is proposed to investigate the cutting deflections. The comprehension of the testbed mechanism, its strength points and weaknesses (i.e., deformation distribution) can help scientists to fully take advantage of the common industrial version of it (see Figure 3-1). Being able to adjust to complicated workpieces with complex geometries, makes this setup configuration flexible which is suitable in modern industry. Analyzing and predicting the part deflections on this cost-efficient and simplified testbed can pave the way for a multi-level optimization method in the future such as designing a compensation model which is able to optimize the depth of cut through the trajectory with respect to the part non-linear flexibility to achieve a surface finish with zero deflection approximately. The prediction model can be achieved with aid of Finite Element Analysis (FEA) software programs such as Abaqus, Deform, etc.

Figure 3-1 Universal holding fixture suitable for aerospace applications
(Canadian metalworking, 2010)

Although CAD and CAE programs are powerful separately, they lack iterative algorithmic features which are necessary in terms of optimization processes in both prediction and compensation models. For the readers convenience, we reiterate the research problems here:

1. In what way one can facilitate the experimentation on modeling machining processes?
2. Can the proposed solution be applied to an FEA for modeling the error prediction or compensation in terms of flexible milling?

Proposed solutions:

1. Creating a systematic semi-automated framework which is friendly to any type of machining simulation.
2. Yes, the framework automates preprocessing and postprocessing for modeling a flexible milling operation of thin components by using Python scripting in Abaqus.

The following flowchart illustrates the prediction model in Abaqus.

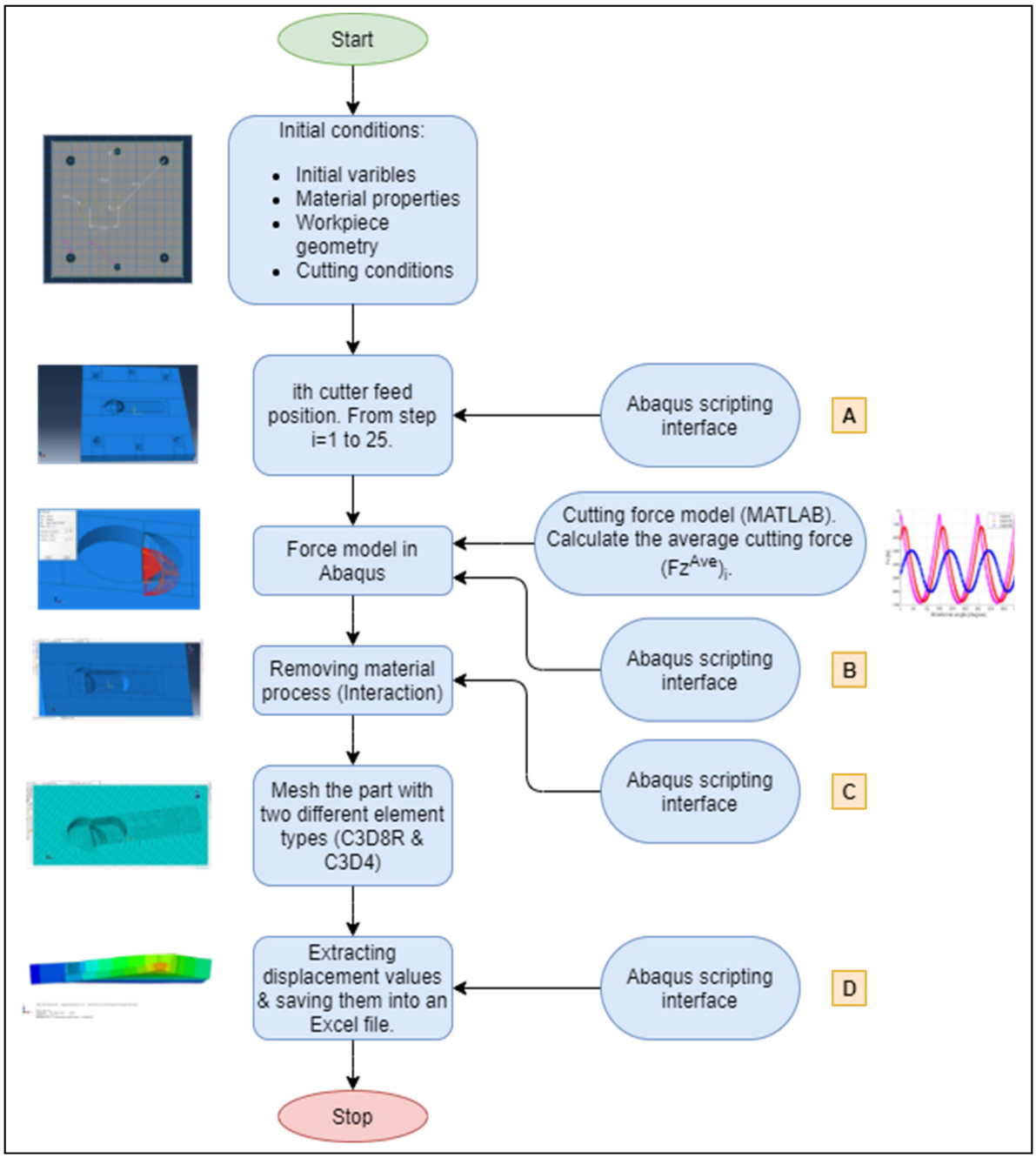


Figure 3-2 Simulation process flowchart

The proposed framework greatly speeds up the experimental loop. For instance, it reduces the chip cell partitioning process from 14 manual steps in Abaqus GUI to only 1 by running the script.

During the cutting process, the workpiece is deformed in all three main directions (X, Y and Z-directions) due to the cutting forces generated by removing materials in forms of chips. Deformation in Z-direction is the dominant reason of geometrical errors for milling thin-floor components using flexible configuration setup (Diez et al., 2015). Other reasons such as geometrical errors of the tool and fixture (i.e., tool wear), the number of flutes, positioning error, tool runout, heat and vibrations have been disregarded in this research project, in favor of the force-induced deflection influence, and for simplification.

In this chapter, a simplified CAE model (Implicit/Static analysis) has been developed to simulate the removing chip process and compute the part displacement along the cutting trajectory based on only the part's elastic deformation during the operation. The method is briefly described below and will be explained in detail in the next sections.

Since designing the cutting tool itself was never the focus of this work and only the effects of vertical cutting forces on the part was the purpose, then with respect to the feedrate (mm/rev) the workpiece has been partitioned into several cells in the machining region (see Figure 3-7). When the tool is advancing into the workpiece due to the feed, these cells represent the amount of material that it is assumed to be removed (see Figure 3-3). In this study, they are called chip cells (uncut chips). As seen in Figure 3-7, throughout the trajectory, 25 positions of the tool have been selected. At each position, the magnitude of average cutting forces in Z-direction is calculated according to the measured instantaneous cutting force model (Altintas, 2012) in MATLAB (see APPENDIX I) and is applied on the load (force) plane. The MATLAB script will be explained in section 3.5. The force plane is assumed to be parallel to X-Y plane with an offset equal to the nominal depth of cut $((DOC)_n)$ from the top-face of the part (see Figure 3-13). This is where all the forces from MATLAB calculation are applied. Once the analysis

is done and the displacement results for all tool positions are collected, it is possible to predict the geometrical profile of the plate on the trajectory after machining.

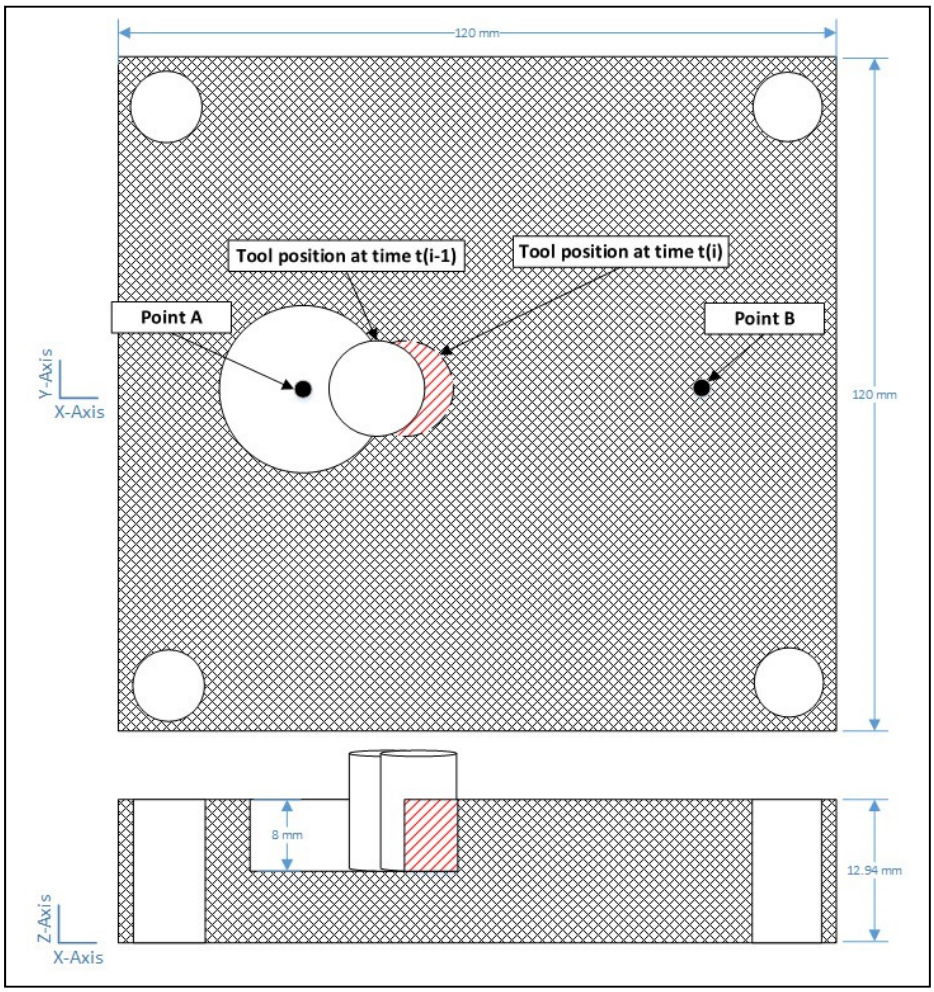


Figure 3-3 Schematic top and side views of tool-workpiece engagement

During both preprocessing and postprocessing in Abaqus, an application programming interface (API) has been applied to carry out some iterative actions which are time consuming for the user such as creating multiple chip cells, steps, surfaces and loads. “The Abaqus Scripting Interface is an application programming interface (API) to the models and data used by Abaqus. The Abaqus Scripting Interface is an extension of the Python object-oriented programming language; Abaqus Scripting Interface scripts are Python scripts.” (Smith, 2009) (see APPENDIX I to VII).

3.2 Cutting conditions

The part is fixed in the four corners and the cutting conditions follow as in Table 3-1. The movement of the tool's center is from point A (x= -25.4) to point B (x= +25.4) as seen in Figure 3-3.

Table 3-1 Cutting conditions (Nguyen, 2016)

Test	Thickness of plate [mm]	Cutting diameter [mm]	Depth of cut [mm]	Rotational speed of the tool [rpm]	Feed rate [mm/rev]
1	12.94	15.875	8	4584	0.3386

It is worth noting that Abaqus does not have any built-in system of units, thus, it is up to the user to define one and be consistent. In this work an SI (mm) system has been considered (see Table 3-2).

Table 3-2 System of units

Quantity	SI (mm)
Length	mm
Force	N
Mass	tonne (10^3kg)
Time	s
Stress	MPa (N/mm^2)
Energy	mJ (10^{-3}J)
Density	tonne/mm^3

3.3 Material property

Aluminum 6061-T6 is the material that is used for the workpiece whose mechanical properties are brought below in Table 3-3.

Table 3-3 Mechanical properties of Aluminum 6061-T6
This table is borrowed from Nguyen (2016)

Ultimate Tensile Strength	310 MPa	AA; Typical
Tensile Yield Strength	276 MPa	AA; Typical
Elongation at Break	12 %	AA; Typical; 1/16 in. (1.6 mm) Thickness
Elongation at Break	17 %	AA; Typical; 1/2 in. (12.7 mm) Diameter
Modulus of Elasticity	68.9 GPa	AA; Typical; Average of tension and compression. Compression modulus is about 2% greater than tensile modulus.
Notched Tensile Strength	324 MPa	2.5 cm width x 0.16 cm thick side-notched specimen, K_t = 17.
Ultimate Bearing Strength	607 MPa	Edge distance/pin diameter = 2.0
Bearing Yield Strength	386 MPa	Edge distance/pin diameter = 2.0
Poisson's Ratio	0.33	Estimated from trends in similar Al alloys.
Fatigue Strength	96.5 MPa	AA; 500,000,000 cycles completely reversed stress; RR Moore machine/specimen
Fracture Toughness	29 MPa-m½	K_{IC}; TL orientation.
Machinability	50 %	0-100 Scale of Aluminum Alloys
Shear Modulus	26 GPa	Estimated from similar Al alloys.
Shear Strength	207 MPa	AA; Typica

The mechanical material properties are defined in FEA as seen in Figure 3-4 and the density's value is 2.7×10^{-9} (tonne/mm^3). It is a solid and homogeneous thin plate which has an elastic behavior. In this work, the plastic deformation is disregarded due to its low influence on the results.

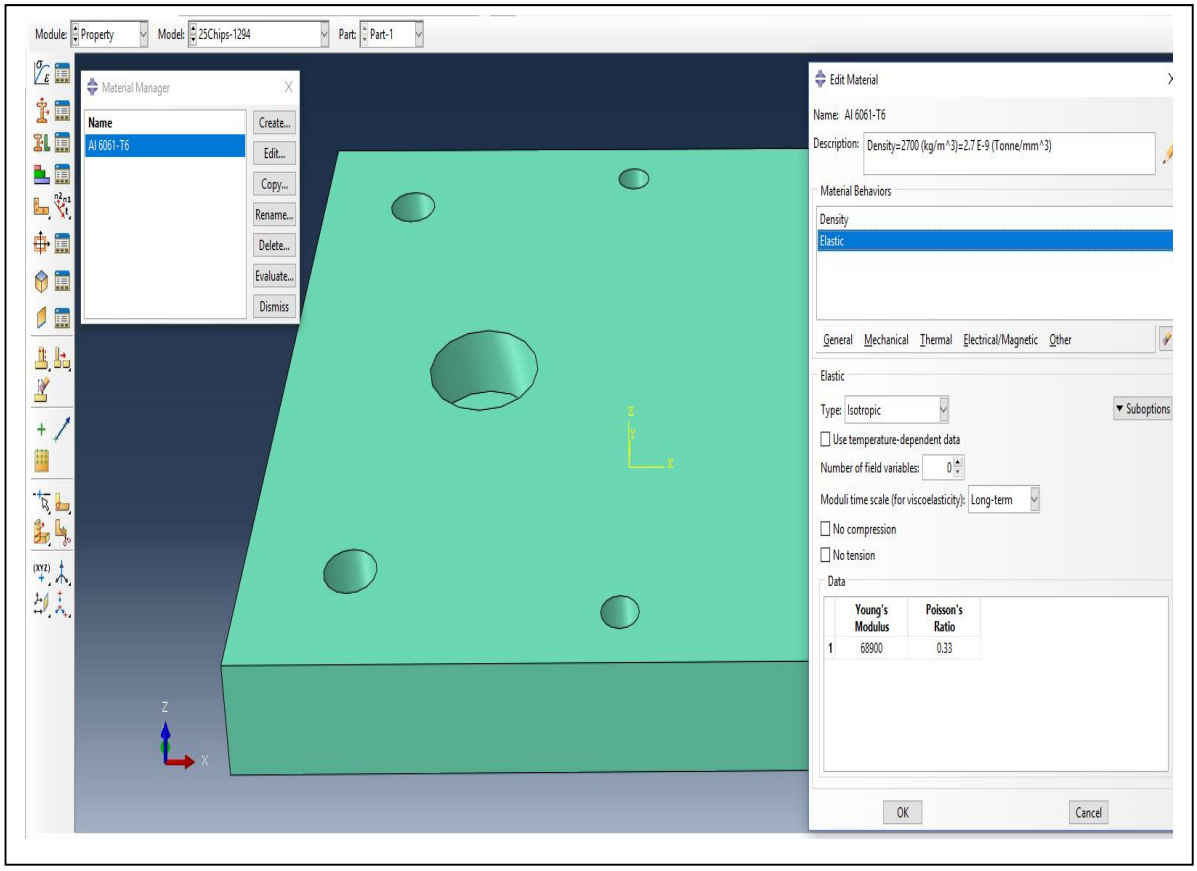


Figure 3-4 Material properties of Aluminum 6061-T6 in Abaqus

3.4 Geometry of 3-D model and partitions

Before explaining the geometry of the part and how to create it in Abaqus, it is worth knowing that how the experimental setup looks like (see Figure 3-5). The part contains 7 holes in total. The big hole in the middle of the part to the left side is the called lead-in pocket. This is created with respect to the nominal depth of cut ($(DOC)_n$) so that the cutter can be placed in the

desired position for the milling operation. Since the work is only about milling operation rather than drilling then, creating this hole in the preparation stage is essential. The four holes in the corners are for clamping the part by four support pins when being machined. And the two small holes located at Y-axis are designed for the measurement purposes.

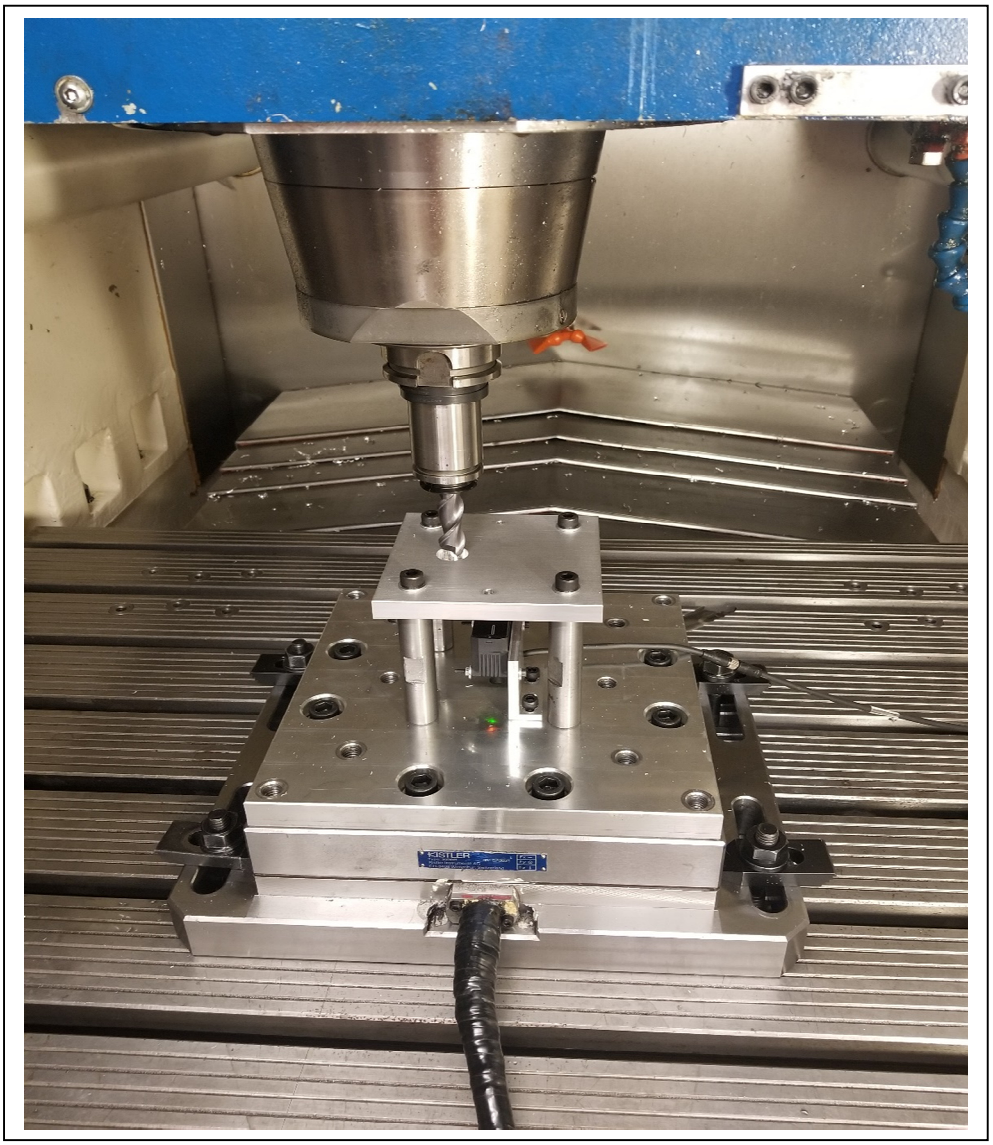


Figure 3-5 Experimental setup 1

The part geometry is a thin component with dimensions of 120 x 120 x 12.94 (mm). Four holes in the corners and two holes on Y-axis have been cut with radiuses of 4.2 (mm) and 3 (mm) respectively (see Figure 3-6).

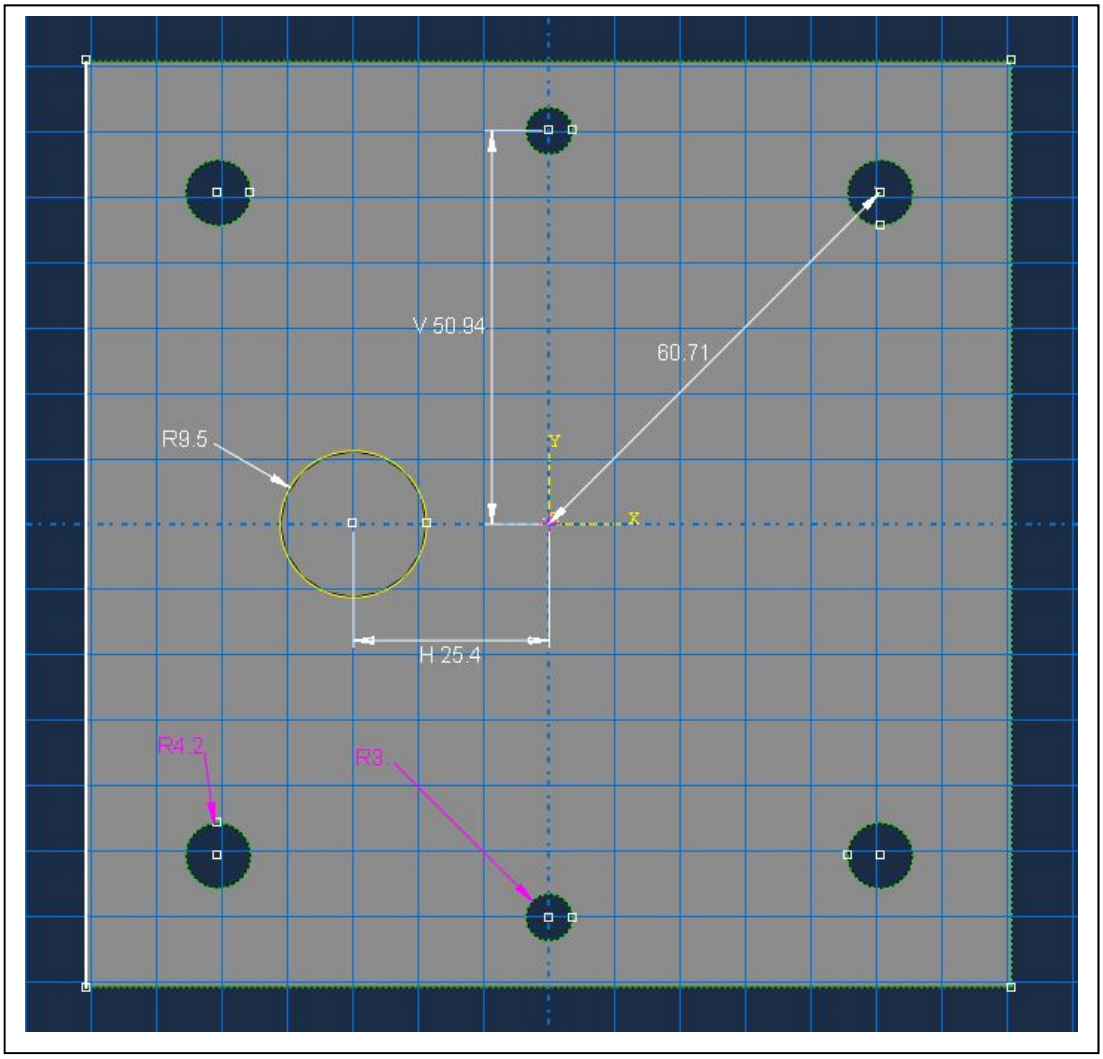


Figure 3-6 Part geometry, top view

The plate is partitioned as in Figure 3-7 into 25 uncut chip cells in the cutting region. The first few chip cells -located in the impact region of the tool and workpiece- have been sized with respect to the actual feedrate (0.3386 mm/rev). Thus, each cell represents the amount of material which is removed after one rotation of the tool. The rest of them are created unrealistically in larger sizes (3.2557 mm/rev) with equidistant positions of the tool's center towards to the end of trajectory (point B in Figure 3-3). Here to avoid the iteration in creating the chip cells in *ASAMBLY module, a Python script is developed to carry out the job which will be discussed in section 3.12 in detail. At the right side of the lead-in pocket and on the load plane, the circular face is partitioned by several straight lines parallel to Y-axis, so that creating surfaces (*SURFACE) for the related loads is possible. This will be explained in section 3.5.

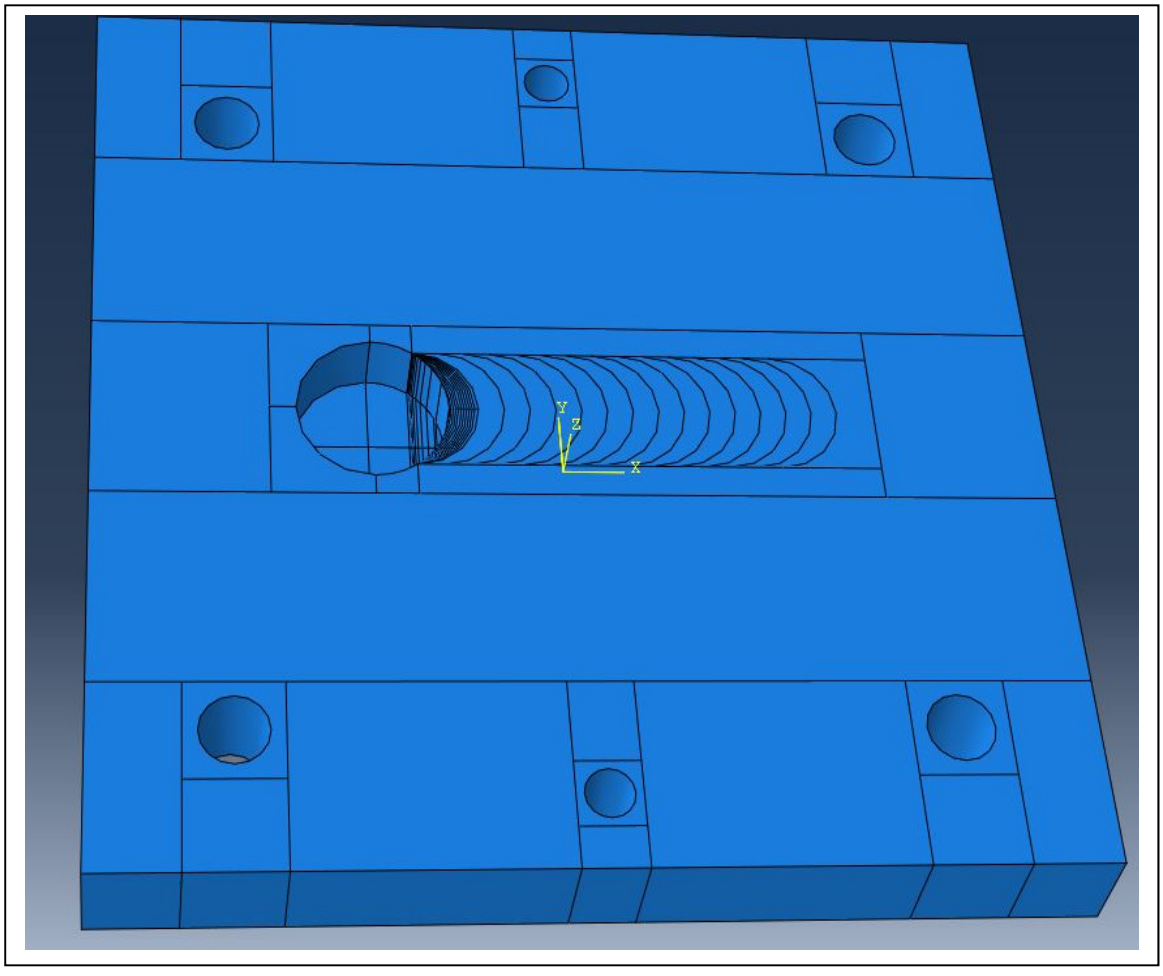


Figure 3-7 Assembly view, partitions

3.5 Cutting force prediction and assumptions:

1. In order to calculate the average cutting force at each position of the tool throughout the trajectory, a MATLAB script is applied (Engin, 2018) and it requires several input variables such as cutter entry, exit and pitch angles (φ_{st}, φ_{ex}, φ_{p}), cutting force coefficients and edge constants. The mechanics of milling process can be shown in Figure 3-8, where *b* is the radial depth of cut (also known as width of cut), *c* is the feedrate (mm/rev-tooth), φ is the instantaneous angle of immersion, *f* is the feed direction and *n* displays the rotation direction of the tool. Unlike the turning process, in milling the instantaneous chip thickness (*h*) varies periodically as a function of time and varying immersion. The approximate value of chip thickness variation is calculated as,

$$h(\varphi) = c\sin\varphi \tag{3.1}$$

Therefore, tangential ($F_t(\varphi)$), radial ($F_r(\varphi)$) and axial ($F_a(\varphi)$) cutting forces are expressed as below,

$$\begin{aligned} F_t(\varphi) &= K_{tc}ah(\varphi) + K_{te}a \\ F_r(\varphi) &= K_{rc}ah(\varphi) + K_{re}a \\ F_a(\varphi) &= K_{ac}ah(\varphi) + K_{ae}a \end{aligned} \tag{3.2}$$

Where, *a* is the edge contact length in Z-direction (axial DOC); K_{tc}, K_{rc}, K_{ac} are cutting constants, and K_{te}, K_{re}, K_{ae} are edge coefficients. All constants and coefficients are directly calculated from metal cutting experiments for a certain set of tool and workpiece (based on the literature. Nguyen, 2016).

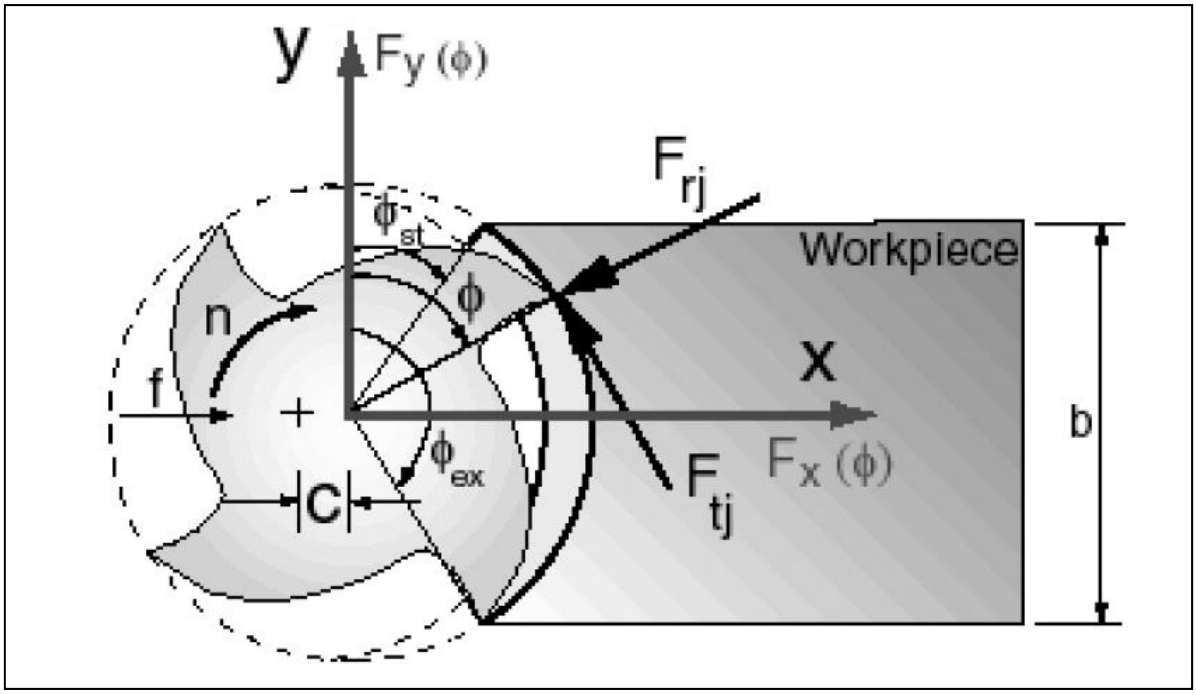


Figure 3-8 Geometry of milling process, (F_{aj} is normal to the paper and j is the flute counter)

This figure is borrowed from Engin (2018)

$$K_{ac} = -485.007\,, \qquad K_{ae} = -3.6147$$

Taken from Nguyen (2016)

(3.3)

An end mill with N number of teeth, helix angle of β, diameter of D, axial depth of cut of a is considered for the cutting forces calculation. The immersion angle is calculated from the normal y-axis and clockwise. In the MATLAB script, calculation of cutting forces for different entry angles is carried out (see APPENDIX I). "The accuracy of the cutting force prediction strongly depends on the selected digital integration interval. When the axial depth of cut is large in helical end milling operations, the differential element height in the axial direction must be very small in order to avoid numerical oscillations on the cutting force wave forms (da=0.01, in MATLAB script). When kinematics and certain properties of the milling process are

considered, it is possible to derive semi-analytical expressions for end milling forces." (Engin, 2018). Now, equation 3.2 can be rewritten as,

$$dF_{t,k}(\varphi,z) = [K_{tc}h_k(\varphi_k(z)) + K_{te}]dz$$
$$dF_{r,k}(\varphi,z) = [K_{rc}h_k(\varphi_k(z)) + K_{re}]dz$$
$$dF_{a,k}(\varphi,z) = [K_{ac}h_k(\varphi_k(z)) + K_{ae}]dz \tag{3.4}$$

Where $dF_{t,k}$, $dF_{r,k}$ and $dF_{a,k}$ are tangential, radial and axial forces acting on differential flute element respectively and with the height of *dz*. k is the flute number (first flute: *k*=1, second flute: *k*=2) and h_k is the instantaneous chip thickness for each flute. At an axial depth of cut *z,* the h_k is rewritten from equation 3.1 as,

$$h_k(\varphi,z) = c \ \sin\varphi_k(z) \tag{3.5}$$

Given h_k values with respect to the axial position, it is possible to measure all $dF_{a,k}(\varphi,z)$ values along the cutting edges for all active teeth and for 2 rotations (see Figure 3-9). Next step is to store these values in a matrix and take an average. Consequently, average cutting forces in Z-direction is calculated as,

$$d\varphi = 0.01 \text{ (Rad)}, \qquad K = 4\pi / d\varphi \tag{3.6}$$

$$Fz^{Ave} = \frac{\sum Fz_i}{K} \tag{3.7}$$

Where, Fz_i is the matrix which stores this data and K is the number of steps in angular integration (see APPENDIX I). The results of the proposed cutting force model in the MATLAB script can be seen in Figure 3-10.

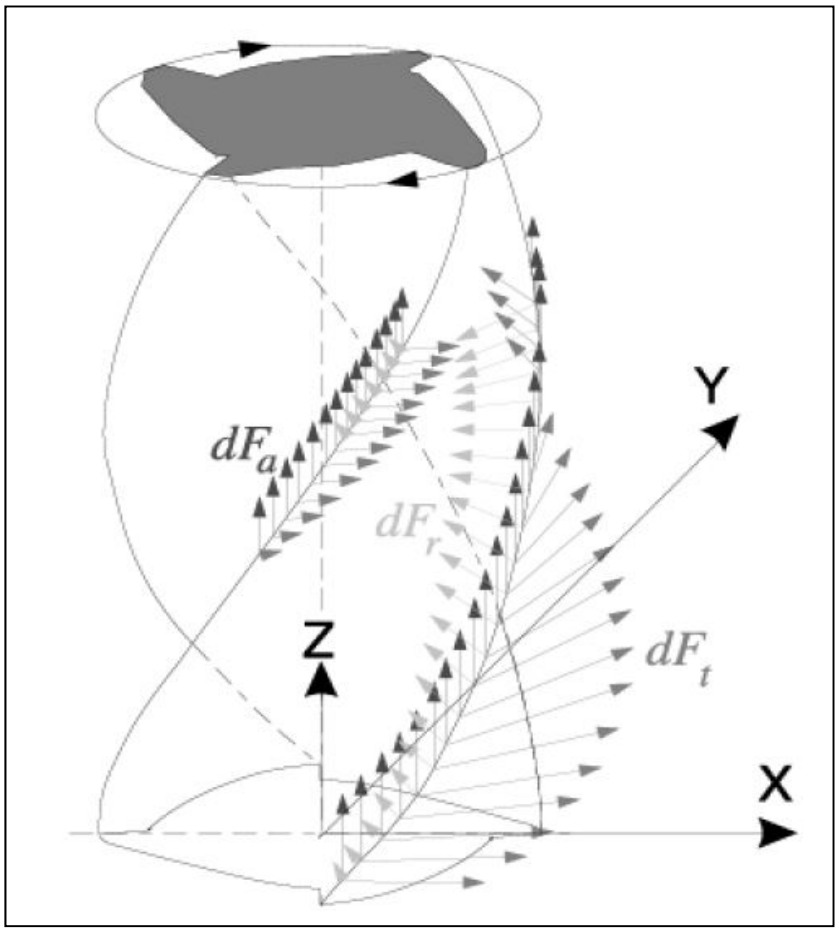


Figure 3-9 Force distribution on the cutting edges
This figure is borrowed from Engin (2018)

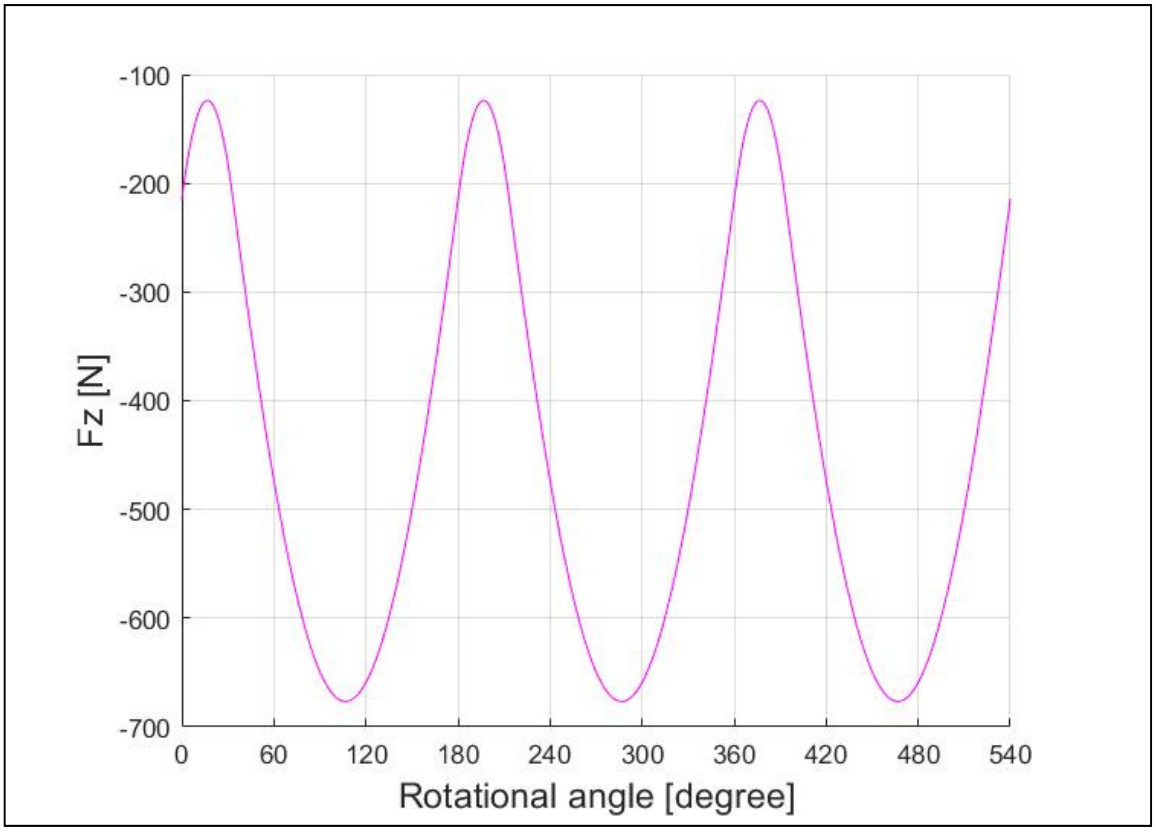
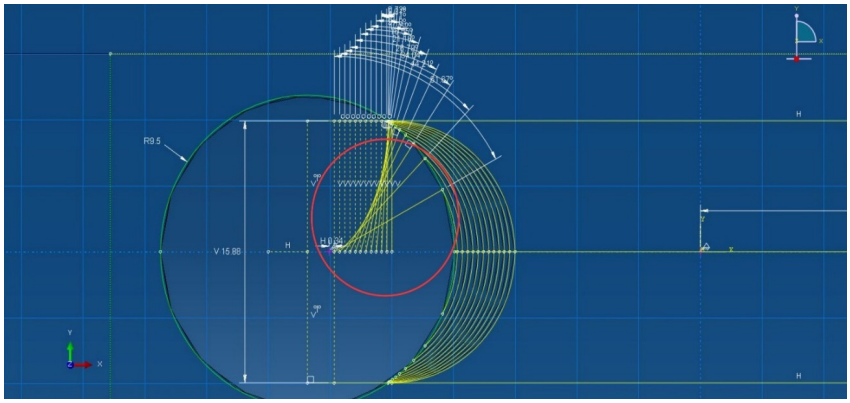


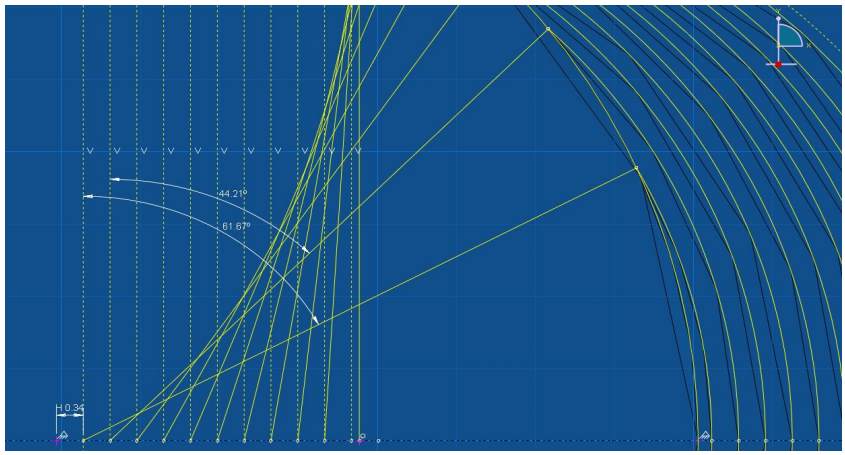
Figure 3-10 Predicted cutting forces using MATLAB

2. Figure 3-11 (a), is an advanced illustration of Figure 3-8 if we assume the feedrate is small and the cutter generates multiple chips with different sizes. In the figure, 12 uncut chips are shown from the top view with respect to their entry angles ($\varphi_{st,}$). An enlarged region from that figure can be seen in Figure 3-11 (b). As seen in Figure 3-11 (b), the radial depth of cut gradually increases from chip cell 1 to 12 as the cutter advances into the workpiece. Accordingly, the tool's rotational movement covers more cutting edge in the last chip cells (10, 11 and 12) and it results in generating more average cutting forces. The highest average force occurs at the 11^{th} chip cell and not 12^{th}. The reason behind it lies in the cutting force formula (Engin, 2018). Due to calculating the average cutting forces, each arc of the cutting edge is divided to many points (APPENDIX I, MATLAB script page 1, dphi=0.01; [Rad] Integration angle). Thus, with respect to the helix angle (β) and axial positions of the points (z), the axial force acting on a differential flute element with height of dz is calculated ($dF_{a,k}$). Based on the cutting

force diagram, the magnitude of the force is zero at angles of zero and π, however the depth of cut and helix angle can affect this. Therefore, the more we are getting close to these angles the more we are adding low-magnitude forces (Fz_i) to the average force formula ($Fz^{Ave} = \frac{\sum Fz_i}{K}$) while increasing K. In other words, the numerator does not change that much in value while the denominator increases. That is why Fz^{Ave} for 12th chip is slightly lower in value comparing to the 11th chip in Table 3-4. To understand how helix angle can influence on the cutting force diagram, section 4.4 is explaining this matter in more detail.



(**a**). Geometry of the first 12 chips and their entry angles



(**b**). Entry angles of chip 1 and 2

Figure 3-11 Geometry of uncut chips, top view

Table 3-4 Entry angles and average cutting forces

Chip num	1	2	3	4	5	6	7	8	9	10	11	12	13
φ_{st} [degree]	61.67	44.21	34.04	26.79	21.16	16.55	12.62	9.18	6.11	3.31	0.72	0	0
Fz^{Ave} [N]	-223.82	-337.99	-384.51	-408.08	-422.11	-430.73	-436.25	-439.82	-441.93	-442.94	-443.07	-442.96	-442.96

3. All forces are applied on a plane located 8 mm below the top-face of the workpiece ($(DOC)_n = 8mm$) for every chip removal analysis, and on a semicircular area depending on the projected cutting edges on X-Y plane. This is called the force plane in this dissertation. The forces directions are along z-axis towards top-face of the part as seen in Figure 3-12.

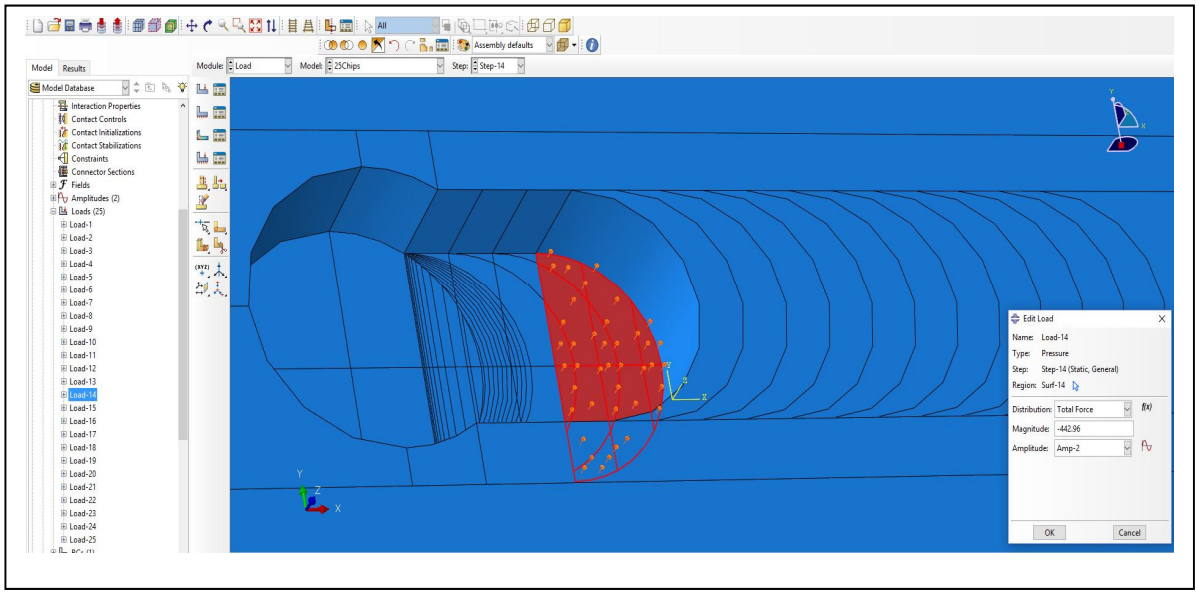


Figure 3-12 Applying cutting force for chip 14th

The *Steps will be explained in detail later in section 3.7, but here to help the readers to understand the purpose of creating several surfaces and defining various loads, an indication to Steps module is covered. We are trying to simulate the milling process in a static analysis. As mentioned before in section 3.4 (Geometry of 3-D model and partitions), 25 uncut chip cells are partitioned, representing 25 positions of the cutter on the workpiece along x-axis. For

each one, a complete and separate step will be defined. In every step, the related chip cell will be removed in the beginning and the relevant load will be applied. Then, the displacement of a relevant node on the force plane will be extracted and stored. If we wanted to design the part with respect to the actual feedrate, the part would have more than 150 uncut chip cells, surfaces and loads. Also, it would prolong the processing time for more than a day using a system with the following specifications: An Intel Xeon Processor E5 v2 Family (6 cores, 12 threads, 2.60 GHz Maximum Turbo Frequency, 15 MB of Cash), 40 GB of memory (RAM) and an NVIDIA Quadro K2000 (2 GB of VRAM) graphics card.

The first chip removals take place in different radial depths of cut (RDOC) but symmetrically (see Figure 3-13 and Figure 3-14). The RDOC constantly changes in the beginning of the tool-workpiece engagement up to a certain position of the tool on x-axis in which it reaches its maximum (to the tool's diameter). That is why the first chip cells are created with respect to the actual feedrate (0.3386 mm/rev) so that the different average cutting forces for each cell can be taken into account. At 12^{th} chip, the first full-immersion of cutting (maximum RDOC) occurs (see Table 3-4). Since after chip number 12, it is always a full-immersion cutting condition and also it is an implicit/static analysis, the rest of the chip cells are created in bigger sizes (unreal feedrate of 3.2557 mm/rev) just to reduce the processing time and simplify the mesh (see Figure 3-14). Creating all the chip cells with the actual feedrate can only help us to have a smoother machined profile as a result in Figure 3-25. The Python scripts to create multiple surfaces and loads for every step is discussed in section 3.12.

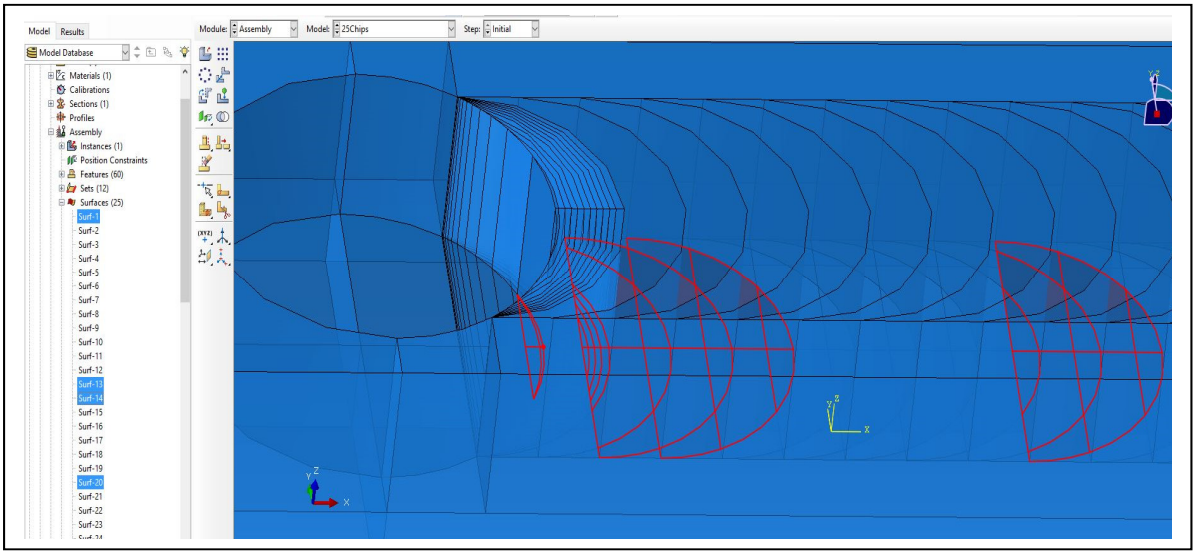


Figure 3-13 Creating surfaces for loads on the force plane for different positions of the tool

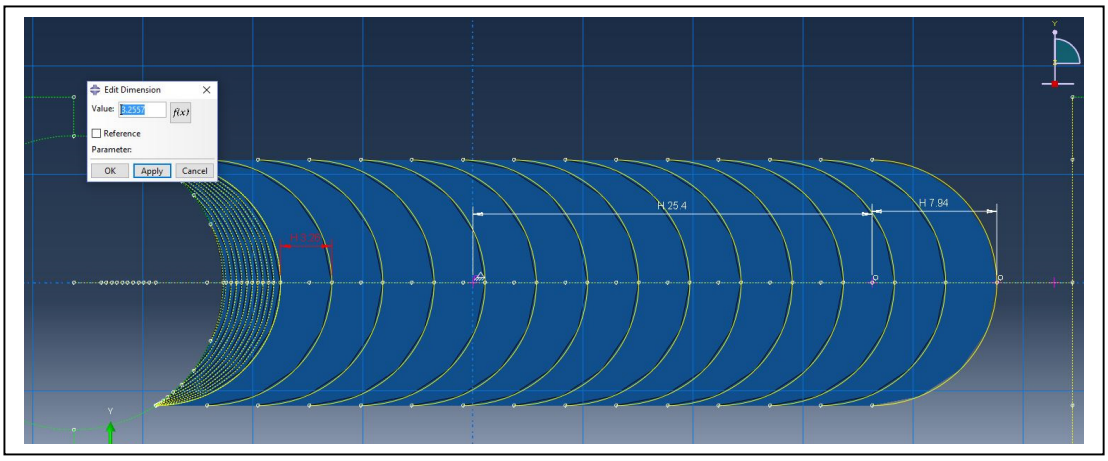


Figure 3-14 Top view of chip cells geometry

In order to be able to mesh the whole model in Abaqus with a mesh size equal to one (it remarkably reduces the running time), it is assumed the full immersion happens at chip 11th with an entry angle equal to zero since its value is so close to zero according to Table 3-4. Therefore, the radial thickness of chip cell 11^{th} is slightly thicker than the real feedrate and chip 12^{th} now is partitioned with the unreal feedrate. Consequently, instead of 12, there are 11 small chip cells on the left side of the trajectory.

3.6 Boundary conditions

The plate is clamped in four corners to the support pins by four bolts. The pins go through the holes and are fastened on top by the bolts as shown in Figure 3-5. Accordingly, the holes movements are restricted in all directions and U1=U2=U3=UR1=UR2=UR3=0. Where, U1, U2 and U3 are movements in X, Y and Z directions, and UR1, UR2 and UR3 are rotations around x, y and z axes (roll, pitch and yaw) respectively (see Figure 3-15).

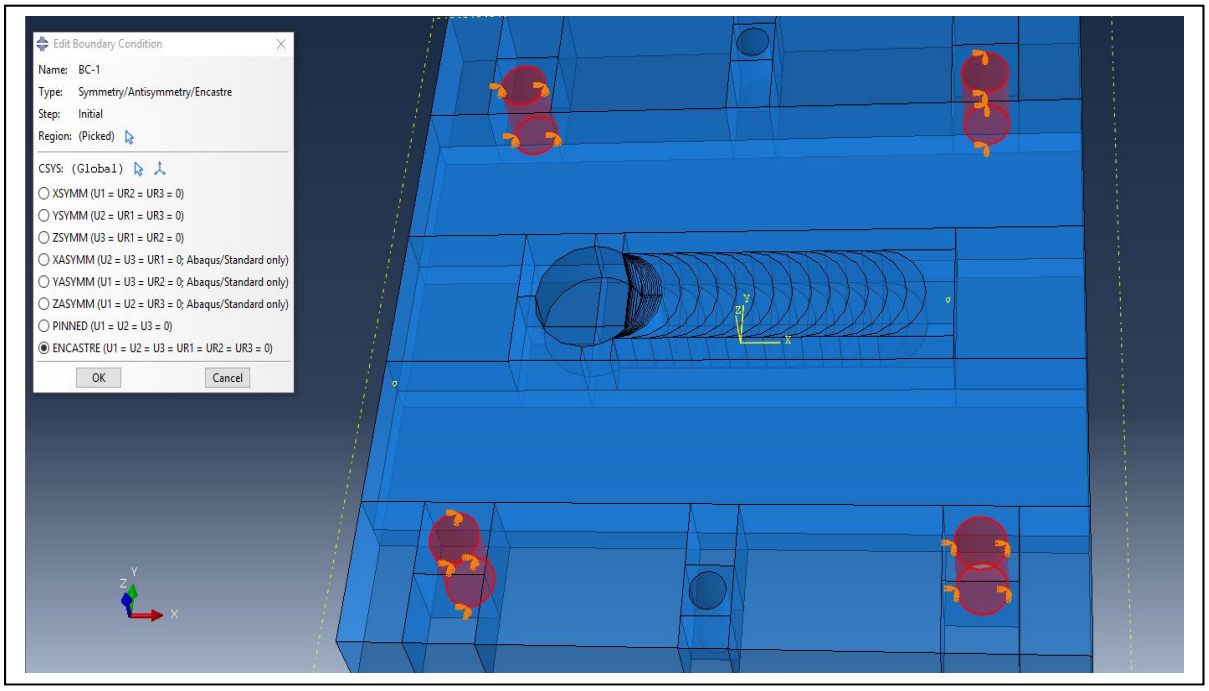


Figure 3-15 Boundary conditions

3.7 Steps

The model contains an initial step which includes the boundary conditions, and also 25 steps of removing material. At every step, the following happens; First, the related chip cell is removed from analysis (by using *INTERACTION), then a constant force throughout the step is applied on the force plane (by using *LOAD) except step-1 (for removing the first chip cell) which its amplitude gradually increases from zero to its maximum (see Figure 3-16), because it is where the first interaction of tool and workpiece occurs. When the next step begins (the

next chip removal), the load from the previous step is deactivated for the rest of the analysis (see Figure 3-17). In order to create multiple steps without using Graphical User Interface (GUI), a script was written which is explained later in section 3.12.4.

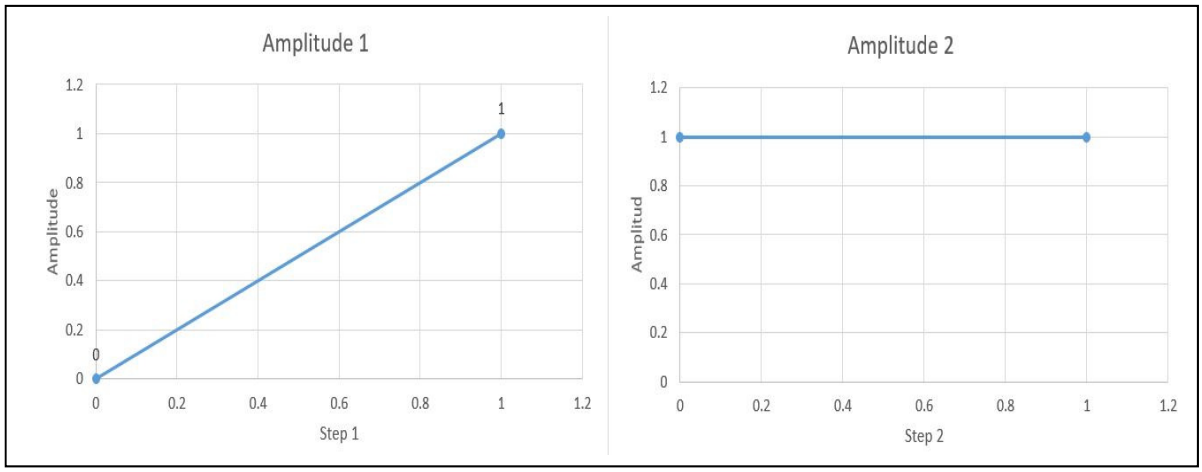


Figure 3-16 Amplitude, a coefficient for the load value in Abaqus
Amplitude 1 and 2 are defined for step 1 and all other steps, respectively

Inactivating the previous loads is considered in the model because when the tool advances through the part, the position of the cutting edges -where the forces are being generated- will change accordingly. The cutting edges are where the teeth are in contact to the workpiece. Since the previous chip has been already generated and separated from the part; then, there is no reason to keep its related load active for the current step of chip removal. However, if we are going to include the influences of the previous chip removal deformation into the current one, we need to keep a few of them active. That way their deformations would be active too. Considering this dynamical behavior, the results of the prediction model can be improved and will be discussed in CHAPTER 5.

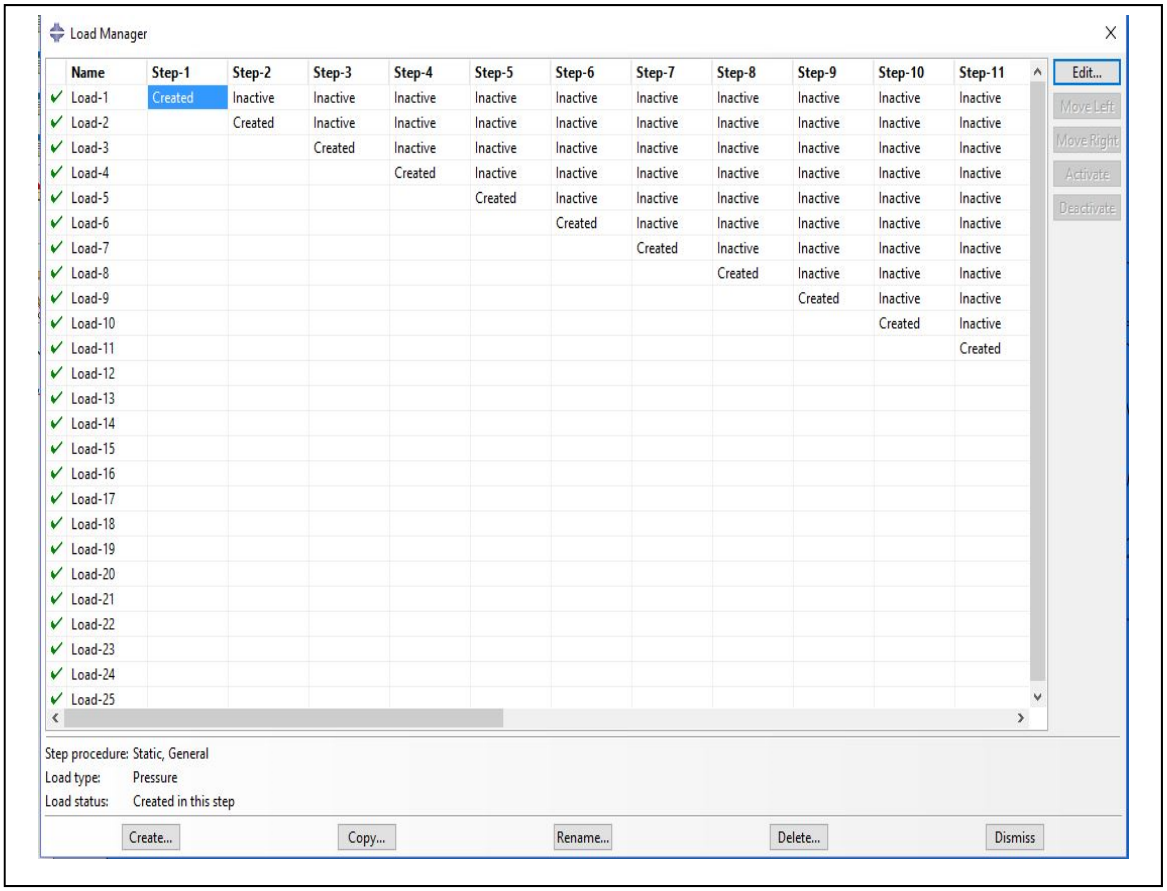


Figure 3-17 Load manager window displays all active and inactive loads throughout all steps

3.8 Renaming partition cells

It is recommended to rename the important partition cells after creating them because tracking them will be easier when it is needed. All chip cells have been renamed since these cells are essential for the future scripts. Relevant details will be presented in section 3.12.2.

3.9 Interaction

The interaction module is for defining various objects including mechanical and thermal interactions between regions of a model. To remove segments of a part in a simulation for a limited period of time or permanently, one can use a model change interaction which deactivates and reactivates a certain set of elements or geometries during analysis (Smith 2009). In the interaction module, the removing material action occurs. In model change interaction feature, the uncut chip cell geometry is selected in GUI for every step and is excluded from the relevant step and all the future steps during the analysis. This type of object only exists in static/implicit and dynamic/implicit analyses. Now, it is possible to write a loop for this object (*MODEL CHANGE) and let Python carry out the job for us and create 25 interactions as seen in Figure 3-18. The logic behind the related code is discussed later in section 3.12.5.

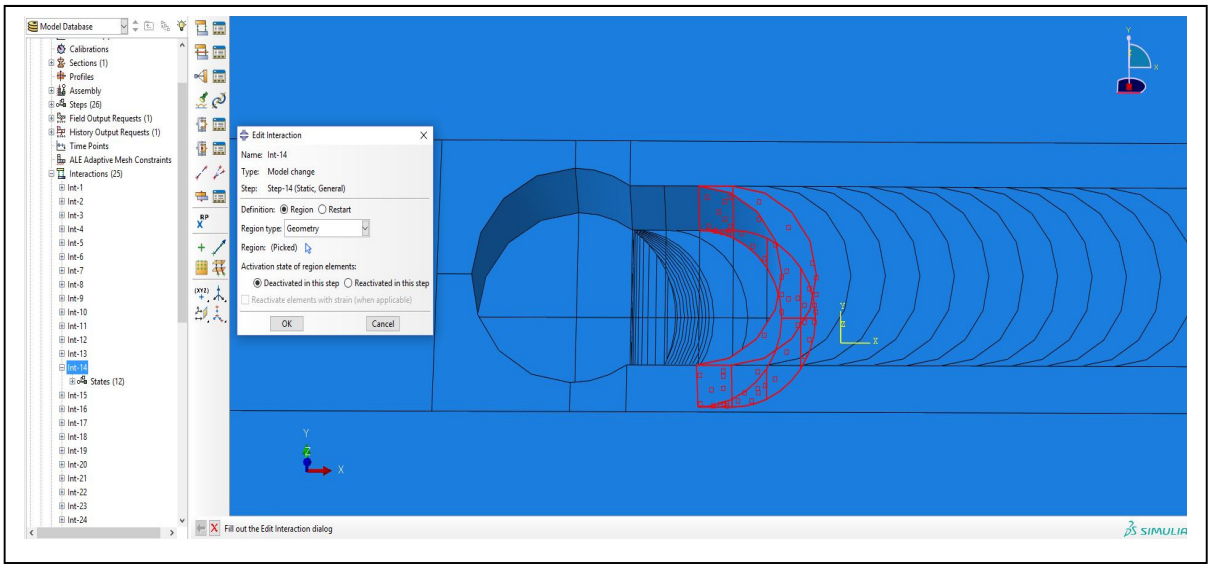


Figure 3-18 Interaction module (*MODEL CHANGE)

3.10 Meshing

As seen in Figure 3-19, the part is mostly meshed with linear hexahedron element shape (C3D8R) excluding the cutting region and the holes areas which are meshed with linear tetrahedron elements (C3D4). Because of the complex geometries in these areas, the tetrahedron elements can show the deformation more realistically. With a proper partitioning and a small mesh size, it is feasible to mesh the whole part with only hexahedron elements, however it also requires a powerful compute infrastructure such as that provided by cloud computing to enable high speed processing.

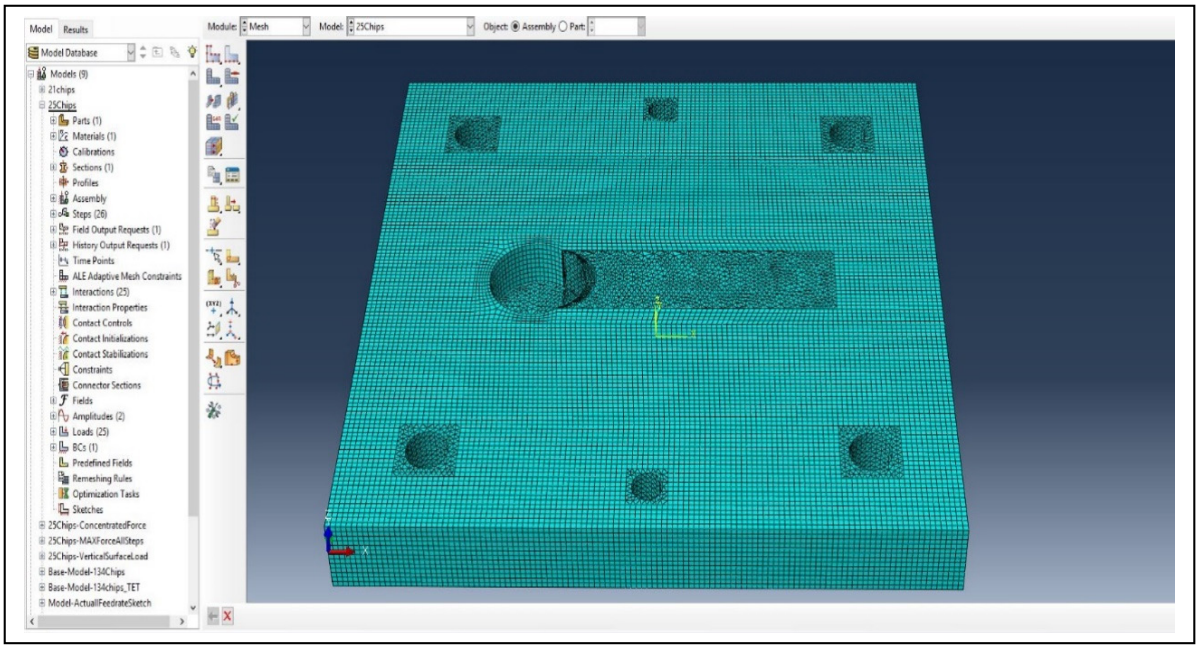


Figure 3-19 Mesh geometry of the workpiece

3.11 Displacement

When preparing the model is completed and a job (*JOB) is created and submitted, the analysis takes approximately 9 hours to process based on the system specification: An Intel Xeon Processor E5 v2 Family (6 cores, 12 threads, 2.60 GHz Maximum Turbo Frequency, 15 MB of Cash), 40 GB of memory (RAM) and an NVIDIA Quadro K2000 (2 GB of VRAM) graphics

card. Once it is done, the results can be shown in the visualization module (see Figure 3-21). 25 nodes have been selected to read their vertical displacement values in the last frame [1]of the related steps (see Figure 3-20). Each step contains several frames based on the incrementation input. Abaqus tries to converge the calculations at each step incrementally until it reaches to a solution in the last frame. Thus, only the last frames' results of the steps have been considered for the displacements data. These nodes are located on the force plane beneath the related chips at the right side of the semicircles and on a line parallel to x-axis (see Figure 3-22). They are assumed to be the representative of the deformed machined surface underneath the tool's position. To read the displacement values for these nodes, there are a few ways in Abaqus GUI including using "Probe Value" dialog box. However, instead of doing this multiple times, a Python script is written to complete the task which will be explained in section 3.12.6. Since the tool's positioning error has been disregarded, then the tool cuts all volumes of material which have deformed in Z-direction with positive displacements. Therefore, the cutter removes more material than is expected (overcut). This will be discussed more in detail in results and analysis section 3.13.

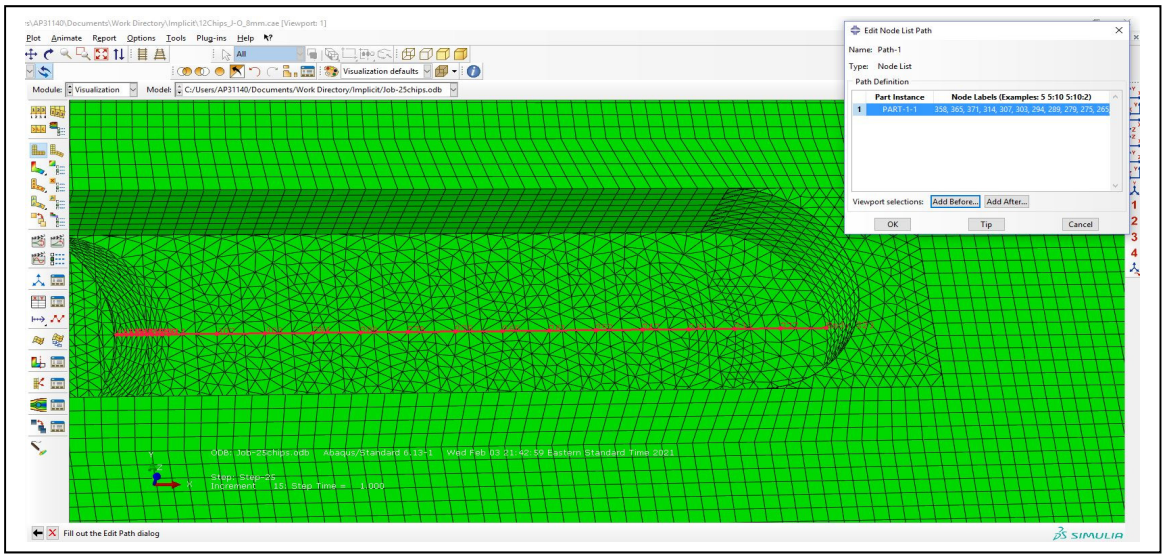


Figure 3-20 Node labels and their positions

[1] According to the Abaqus user's manual, the increments at which the user selects to be written to the output database file are called frames (Smith, 2009).

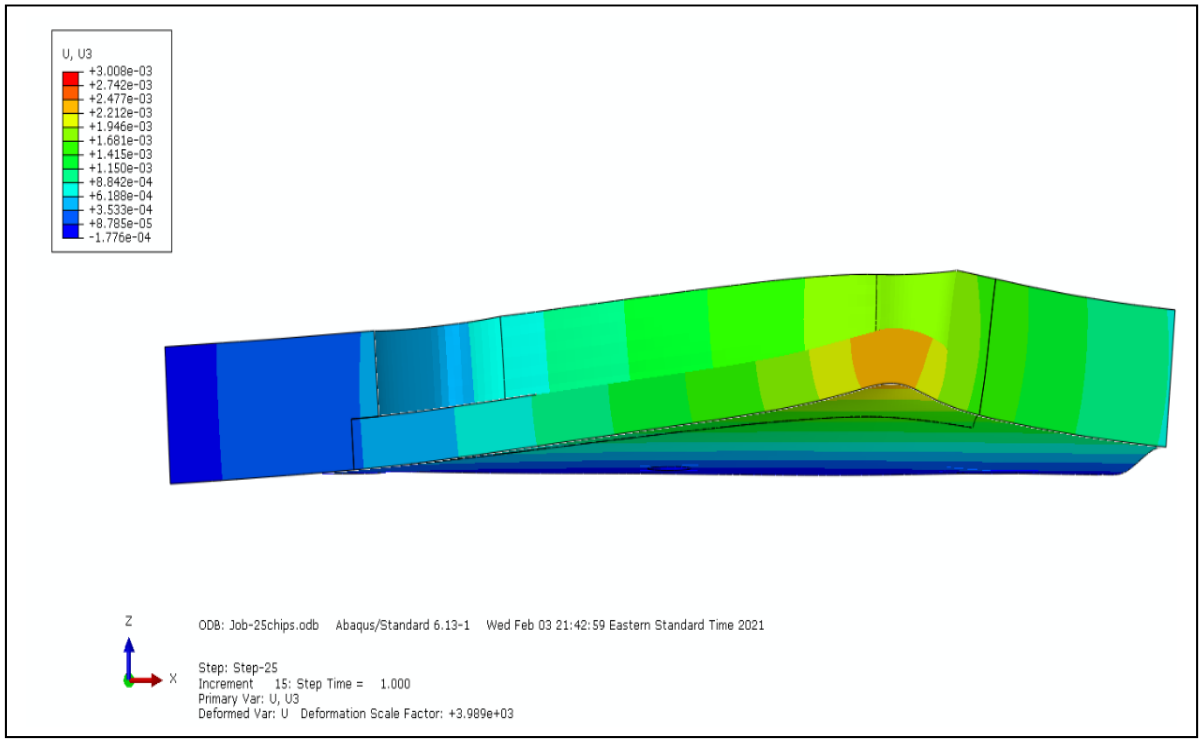


Figure 3-21 Side cut view of the workpiece during deformation (displacement spectrum)

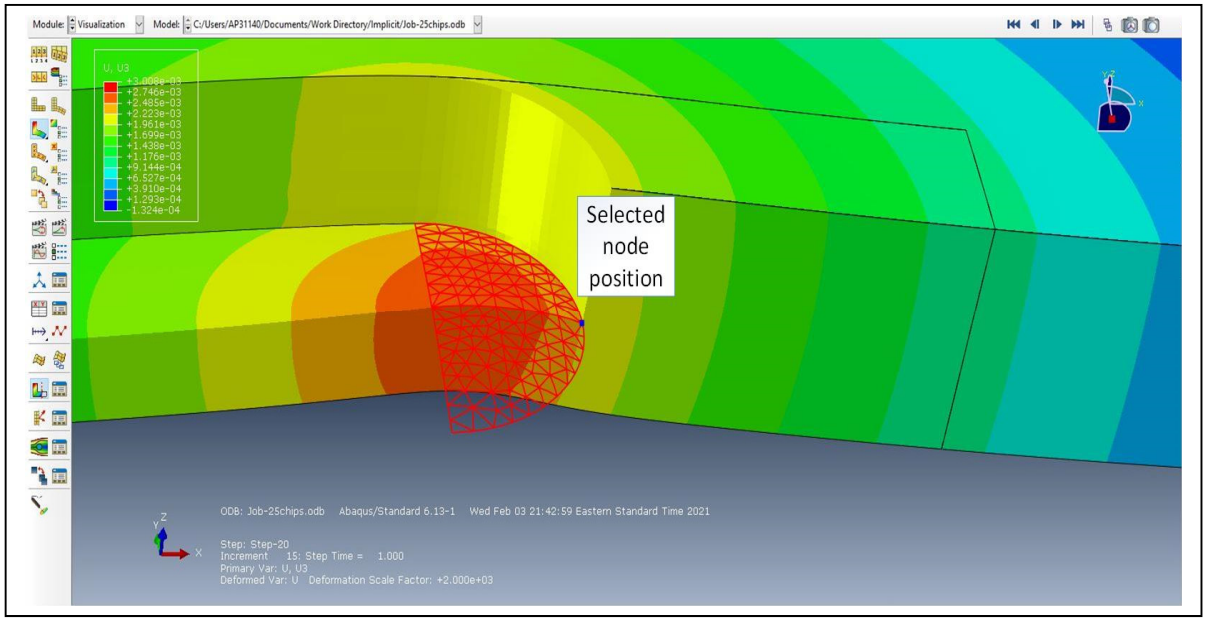


Figure 3-22 Node position

3.12 Abaqus Scripting Interface

In order to facilitate the process of creating the model in Abaqus/CAE and avoid unnecessary manual iterations such as creating multiple chip cells (partitioning), surfaces, loads, steps and even reading the results, various scripts in Python have been developed to conduct these tasks (please see the APPENDIX section for future details and the relevant codes). The term preprocessing is employed for the steps taken in preparing the model in Abaqus before starting the analysis, and the term postprocessing refers to reading and visualizing the results after the analysis is done. The Mdb object is the Abaqus model database which saves analysis controls and models. The Odb object (output database) on the other hand, stores all the data related to the results and is saved in an .odb file. A few files are generated by Abaqus as soon as the model is saved in the work directory. Files with the extensions .cae, .jnl, .odb and .rpy are the critical ones in this dissertation. Everything that a user does in the Abaqus Graphical User Interface (GUI) environment (see Figure 3-23) related to the model database will be saved in the .jnl file as lines of code. Therefore, it is possible to see how Abaqus transfers the actions in GUI into lines of code and saves them in a separate script for the future manipulations. For instance, when a model is being created and developed in the GUI window, we can open the .jnl file in a text editor and track all taken actions. The Sublime text editor has been used for this purpose throughout the present study. Knowing the names of the objects and their functionalities makes it easier for users to manipulate these data. Hence, a basic knowledge of both Python and Abaqus is vital.

Prior to arriving at creating the model, a line of code as seen in Figure 3-23 is entered in the script window and we press enter. No immediate effects are observable to user as a result of running the command, but this line will help us later with reading the .jnl file (JulianBauer, 2017). Enter the line as follow:

```
session.journalOptions.setValues(replayGeometry=COORDINATE,
recoverGeometry=COORDINATE)
```

Because Python language does not include all the functionalities of Abaqus, various modulus of Abaqus must be imported by the user in the scripts (see APPENDIX II). There are a few ways to run a script in GUI such as using “Run script” option or by entering all the lines of code directly into the script window as illustrated in Figure 3-23.

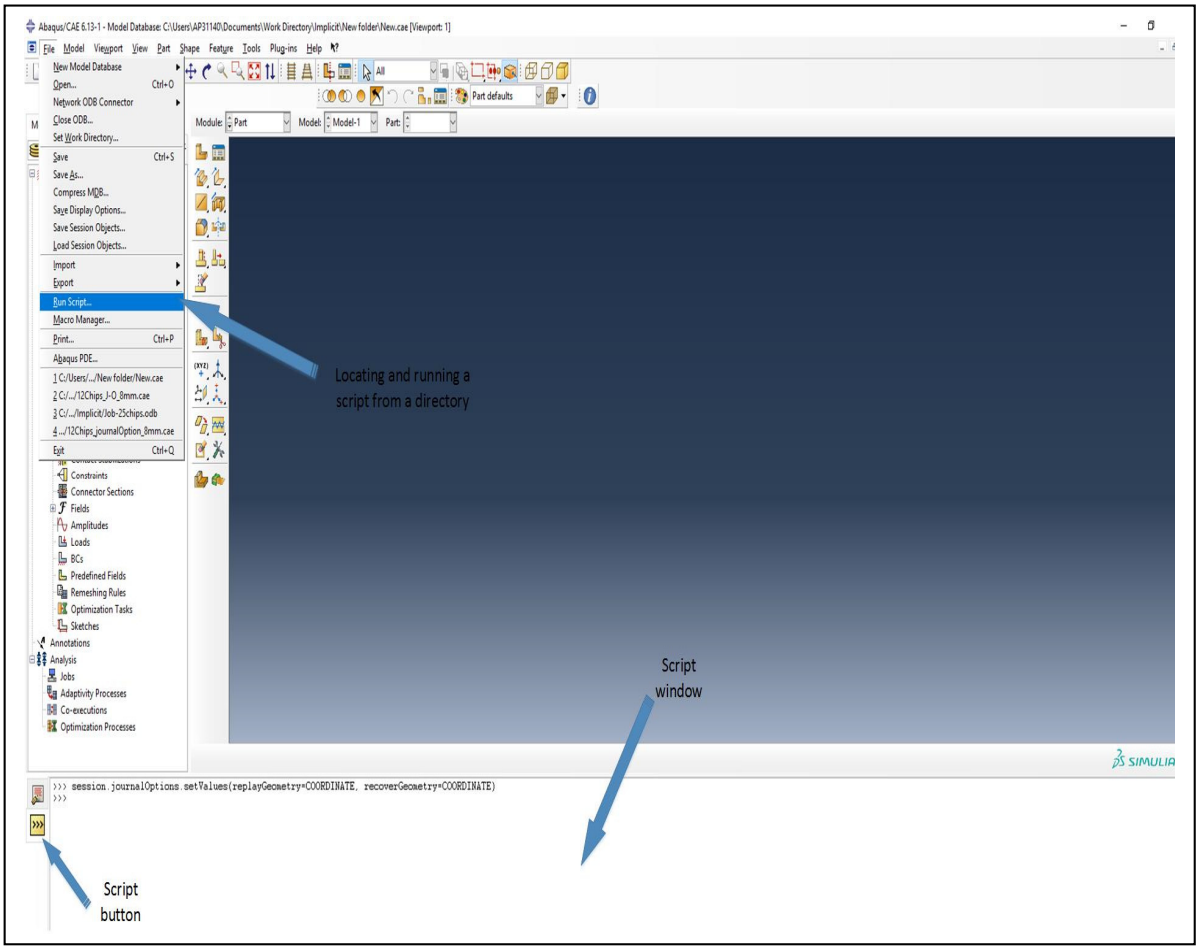


Figure 3-23 Main Abaqus GUI window

3.12.1 Partitioning the part (fragment A of the flowchart in Figure 3-2)

Partitioning the part occurs in the assembly module. We create a chip cell by using the Extrude/Sweep edges object as shown in Figure 3-24. Meanwhile, the .jnl file is open in Sublime and the generated codes can be seen. This approach repeats for all the scripts in this dissertation to determine a way for iterations. Once the desired object’s name and its arguments

are shown, writing the script can begin. As seen in APPENDIX II, the object "PartitionCellBySweepEdge" contains 3 arguments [2]named cells, edges and sweepPath. For argument "cells", the position of the initial cell is wanted which is highlighted in Figure 3-24. The argument "edges" takes the positions of the edges that we want to sweep. The edges must be defined as a closed perimeter. In this case, there are 6 edges for every C-shape (crescent) area on the top-face of the part that we have already face-partitioned. And, the argument "sweepPath" requires a path (an edge or a line) to sweep. Since all cells share a same path to sweep, then only the other arguments must be manipulated. An edge from a corner parallel to z-axis with a top-to-bottom direction was selected as the sweep path for creating all chip cells. Knowing the position of the 6 edges and the unreal feedrate (3.2557 mm/rev) makes it possible to create an "if" loop. It is worth mentioning that the object "findAt[3]" is the replaced object of "getSequenceFromMask" in .jnl file, due to the initial line of code that was entered in the script window before creating the model.

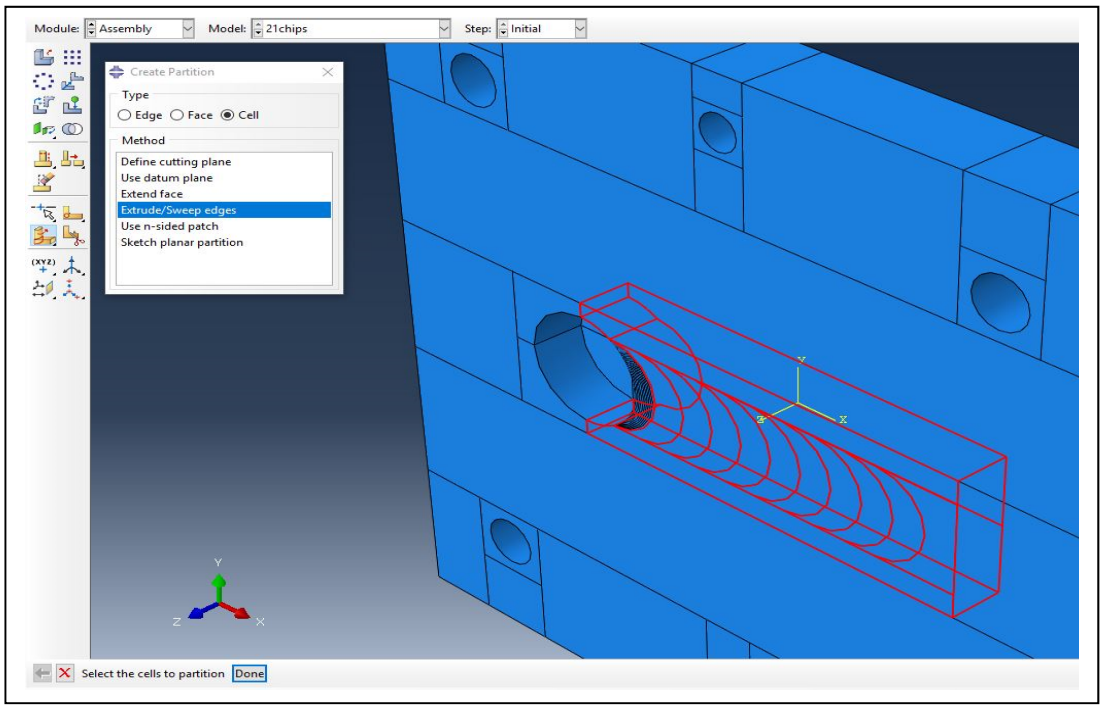


Figure 3-24 Partition cells, Extrude/Sweep edges

[2] Argument: information that are passed into a function.
[3] Object findAt helps to locate edges, surfaces, cells or any other geometry type.

3.12.2 Renaming partition cells

Any new cell which is created in GUI is saved with a default name (i.e., Partition cell-1). As seen in APPENDIX III, “changeKey” is an object to rename any cell in the assembly module. There are 2 arguments for this object. One is the current name of the cell and the other one is the desired one. Knowing the default names of the cells can help the user to optimize the script. That is why, it is recommended to rename the cells right after being created.

3.12.3 Creating multiple surfaces (fragment B of the flowchart in Figure 3-2)

25 surfaces have been created for the purpose of defining the loads in Abaqus. If we wanted to keep the actual feedrate throughout the trajectory, the number of surfaces, loads, chips and steps would exceed 150. Then, having a script to do the iteration for us is needed. As mentioned before in section 3.5 and seen in Figure 3-13, a pattern is essential in order to create a loop for surfaces. This does not happen before surface 14th as seen in the figure. Hence, the qualified surface for this job is surface 14th. Surface 13th and 14th can be located in Figure 3-13 in the middle and among 4 highlighted surfaces. As been shown in the figure, each semicircle surface contains a few faces. Surface 13th has a few smaller faces in its surface-set which will never repeat in the next ones. These small faces exist because of the partitioning process for the previous chip cells with the lower feedrate. Therefore, to avoid making the loop more complex, the loop starts at surface 14th. APPENDIX IV shows that the object “Surface” contains the surface name and all selected faces as its arguments. For instance,

```
Surface(name=’surf-14’, side1Faces= (POSITIONS OF THE FACES))
```

That is why surface 13th is not a good pick to start the loop with. Because those small faces in its surface set will never appear in the next sets of surfaces. Therefore, the loop starts at chip number 14 to 25. Knowing that the positions of y and z coordinates of these selected faces are constant along the trajectory, we can write a loop by increasing the x-coordinate and locate the position of next set of faces for the next surface.

3.12.4 Creating steps and loads (fragment B of the flowchart in Figure 3-2)

The challenge here is to find a way to add more steps and create their related loads and deactivate the previous load without using GUI environment. As shown in APPENDIX V, the objects we are seeking are: “StaticStep”, “Pressure” and “deactivate”. “StaticStep” refers to the type of step and includes various arguments such as name and previous. “Pressure” object on the other hand, takes arguments including the type of distribution, the magnitude, the region to apply it (which is the surfaces discussed in the previous section) and the name of the load. Taking into account the names of the “steps”, “surfaces” and the “loads”, the user is able to define and utilize a counter variable to iterate through them. The object “deactivate” refers to skipping a load or several loads during a certain step. Then in line 28, we are trying to deactivate a load from the previous step in the current step.

3.12.5 Interaction loop (fragment C of the flowchart in Figure 3-2)

APPENDIX VI shows how this interaction works in a loop to remove a cell throughout a step and the following steps. First, finding the x-coordinate of an initial chip cell is needed. To do so, a point anywhere inside the cell is required. Then, with respect to the feedrate and the steps’ names, the positions of the following cells will be determined. Every cell can be located in “ModelChange” object by a point using “findAt” object. This point must be located inside of the cell. Once its coordinates are known, by increasing the value of x-coordinate in a loop, the position of the next cells can be found. At the end of the script, a “job” is created and submitted for analysis.

3.12.6 Extracting U3 for specific nodes and storing them (fragment D of the flowchart in Figure 3-2)

In all the previous sections for scripting, the .jnl file was used to read the relevant lines of code for each particular action we took in GUI. When the analysis is done, all data related to the

results are written into the .odb file. To be able to read the data, we open “abaqus.rpy” file and switch to the visualization module in Abaqus. In APPENDIX VII, the displacement value in Z-direction (U3) for each node in the last frame of every step is extracted and written in an excel file. For this loop, objects such as “frames”, “fieldOutputs” and “nodeLabel” are crucial in order to address the data we are seeking properly. Last but not least, scripting using the .odb file requires new modules to be imported which are shown in APPENDIX VII as well.

3.13 Results and analysis

In this chapter a simplified static analysis was developed in Abaqus to predict the elastic behavior and eventually the deflection of a thin Aluminum plate during the slot milling using a flexible setup configuration. The cutting deflection was estimated according to the material deformation in different positions of the tool along the trajectory. This deformation was caused by an applied cutting force model in Z-direction, in the prediction model which represents the generated cutting forces by removing material (chips) from the workpiece. As mentioned before, other reasons of errors such as geometrical errors of the tool (i.e., tool wear) and fixture, positioning error, tool runout, heat and vibrations have been disregarded. Figure 3-25 shows the deviations of the cutting process based on the Abaqus model. In other words, it is the cut surface profile in x-z plane predicted by FEA comparing to the perfectly flat ideal surface.

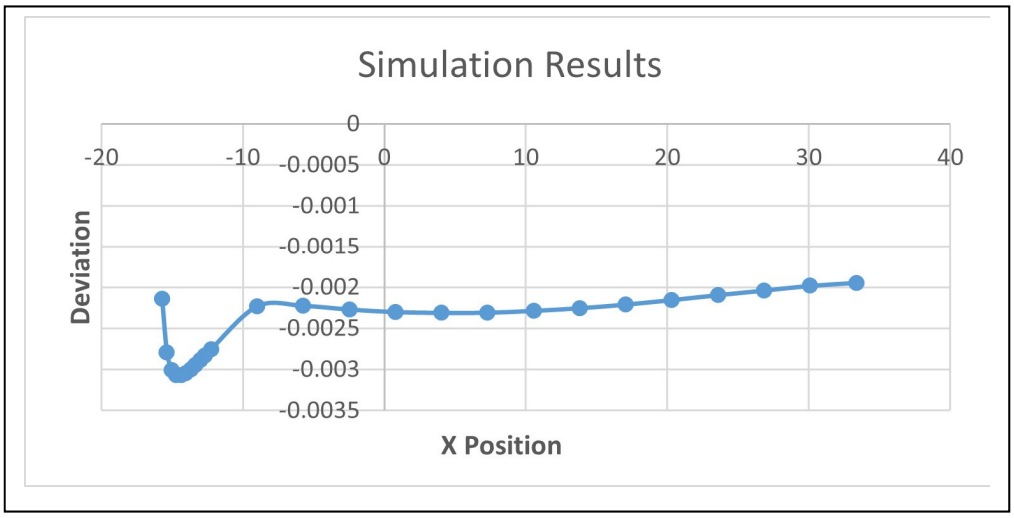


Figure 3-25 Machined profile of the cutting region

The range of X position is between -15.7 to 33.34. It is because only 25 nodes were defined to read their displacement results as seen in Figure 3-20. Defining any other node out of this range would be unnecessary, since no step was created for somewhere with no defined load. Please see section 3.11 for more information about selecting these specific nodes. X position =0 have been shown in many figures before such as Figure 3-6 and Figure 3-7. And it is the center of the part where the origin is located. The deviation value for the simulation model is expressed as,

$$(\varepsilon_i)_s = -(U3)_i \tag{3.8}$$

Where, i is the step index and $(U3)_i$ is the vertical displacement value of the relevant node through the trajectory at the end of each step. The points in Figure 3-25 represents the surface form errors of the selected nodes from chip number 1 to 25 on the machined floor parallel to x-y plane. As it is assumed the tool's vertical position is constant (tool as a cantilever beam) throughout the cutting process; then, the deviation and displacement have an indirect relationship. As seen in Figure 3-21 and Figure 3-22, all displacement values are positive because of the load direction. This means the tool has cut more material as it advances through the workpiece (overcut). The first X positions of the nodes are closer to each other because the first uncut chip cells were created with a lower feedrate (0.3386 mm/rev). The maximum deflections happen first when the impact of the tool-workpiece occurs and next, is somewhere close to the center of the workpiece ($X \approx 5\,\text{mm}$) where, the part is at its maximum flexibility. The Fz^{Ave} value reaches to the highest prior to the radial full-immersion condition and after that it slightly decreases to a constant value along the trajectory. From now on, the deformation distribution relies only on the positions of the applied loads. This was explained in detail in section 3.5.

CHAPTER 4

EXPERIMENTAL STUDY

4.1 Introduction

In this chapter, an experimental test with respect to the same cutting conditions as in simulation model is conducted to validate the simulation results regarding the machined thickness of the workpiece. A method of applying flexible setup configuration is considered for the slotting process. This testbed is a simplified version of the ones available in the industry. The industrial version is compatible to sustain more complex geometry due to the flexibility of its setup configuration. Figure 4-1 shows the setup preparation. In contrast to the full back plate support machining, in this approach, the part is supported by four adjustable pins in the four corners. Thus, the plate is able to be deformed freely vertically. The part has been drilled at its corners and the location pins are fixed at these positions by fastening bolts on top of them. There are two smaller holes located at y-axis of the workpiece which are designed for measurement purposes. The machined part then will be measured by help of a Coordinate Measurement Machine (CMM) and the results will be collected and compared to the prediction model.

4.2 Preparing raw material

The material is a thin block of aluminum 6061-T6 with dimensions of 120 x 120 x 12.94 (mm). There are six holes and one pocket on the part which have been milled with different diameters. Four holes of 8.4 (mm) to clamp the plate to the pins on the testbed, two holes of 6 (mm) as references for the measurement process and a lead-in pocket with diameter of 19 (mm) for the tool to be placed at its initial position with respect to the nominal depth of cut ($(DOC)_n = 8\ mm$) are created. The plate has been milled using the flexible configuration setup

when creating the lead-in pocket. This could result in an initial deviation on that region before the slotting operation which will be discussed in section 4.6.

4.3 Machining process using the flexible configuration setup

The milling machine which conducted the experiment is a three-axis Huron K2X10 CNC. An end-mill Niagara Φ5/8'' -2F-TiCN coated was utilized as the cutting tool with a pitch angle of $\angle 180^{\circ}$ and helix angle of $\angle 30^{\circ}$. It is worth pointing out that the helix angle was estimated by the lab technician. This information is not public by the tool manufacturer. A 3-D Kistler 9255 type dynamometer table and a Keyence displacement sensor (an IL-1000 sensor amplifier) were also applied to record the total cutting forces acting on the support pins and measure the displacement of the center of the bottom face of the plate during machining respectively (see Figure 4-1). Later on, in the measurement section, we noticed neither the displacement sensor nor the two smaller holes on y-axis proved useful. The cutting conditions for the experiment are the same as illustrated in Table 3-1. Figure 4-2 illustrates the part geometry when the test is done.

Figure 4-1 Experimental setup 2

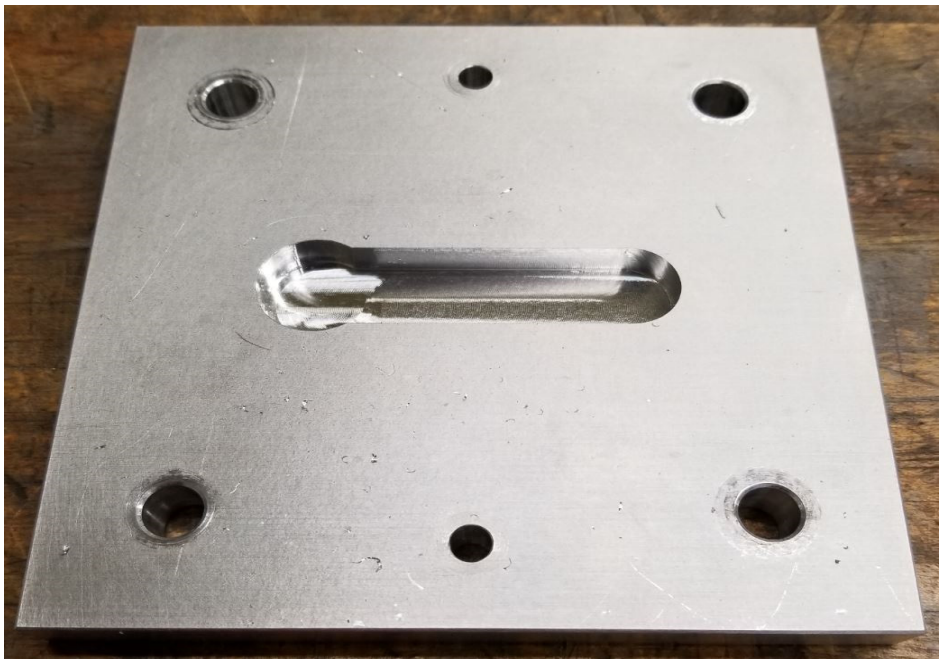


Figure 4-2 Machined part after the slotting process

4.4 Vertical cutting forces

Figure 4-3 shows a comparison between simulation and experimental results for generated cutting forces in Z-direction. As covered before, section 3.5 provides the details of the calculated cutting forces by MATLAB. The cutting force profiles do not fit on each other completely. Generally, it is because of the reasons below.

First, the cutting force model (Altintas, 2012) that is utilized in this dissertation is a simplified one and lacks a few other variables in the calculation (such as rake angle, lead angle, etc). Second, the actual depth of cut (DOC) in the experiment is different from the nominal one in the numerical model (MATLAB script) due to the part deformation in this particular case. Thus, if more DOC is occurring, then more cutting forces are generated and also, more deformation happens and vice versa. Third, different helix angle can influence on minimum and maximum values of cutting forces in Z-direction in one rotation and is able to shift the positions of these points (peaks and valleys) as well. For instance, as mentioned before in this

work, only a helix angle of $\angle 30^{\circ}$ is assumed for the calculation in Altintas cutting force model. If this value were zero, the peak and valley would occur close to $\angle 0^{\circ}$ and $\angle 90^{\circ}$ respectively. It is because these angles define the lowest and highest chip thicknesses for a generated chip. The more chip thickness means the more cutting forces in all directions are required (Altintas, 2012). In Figure 4-4, when keeping the other variables unchanged in the cutting force calculation (APPENDIX I), the influence of different helix angles is substantial (Izamshah, Yuhazri, Hadzley, Ali, & Subramonian, 2013). This helps readers to understand why knowing the exact angle is important for calculating the forces or the magnitude of average cutting forces correctly.

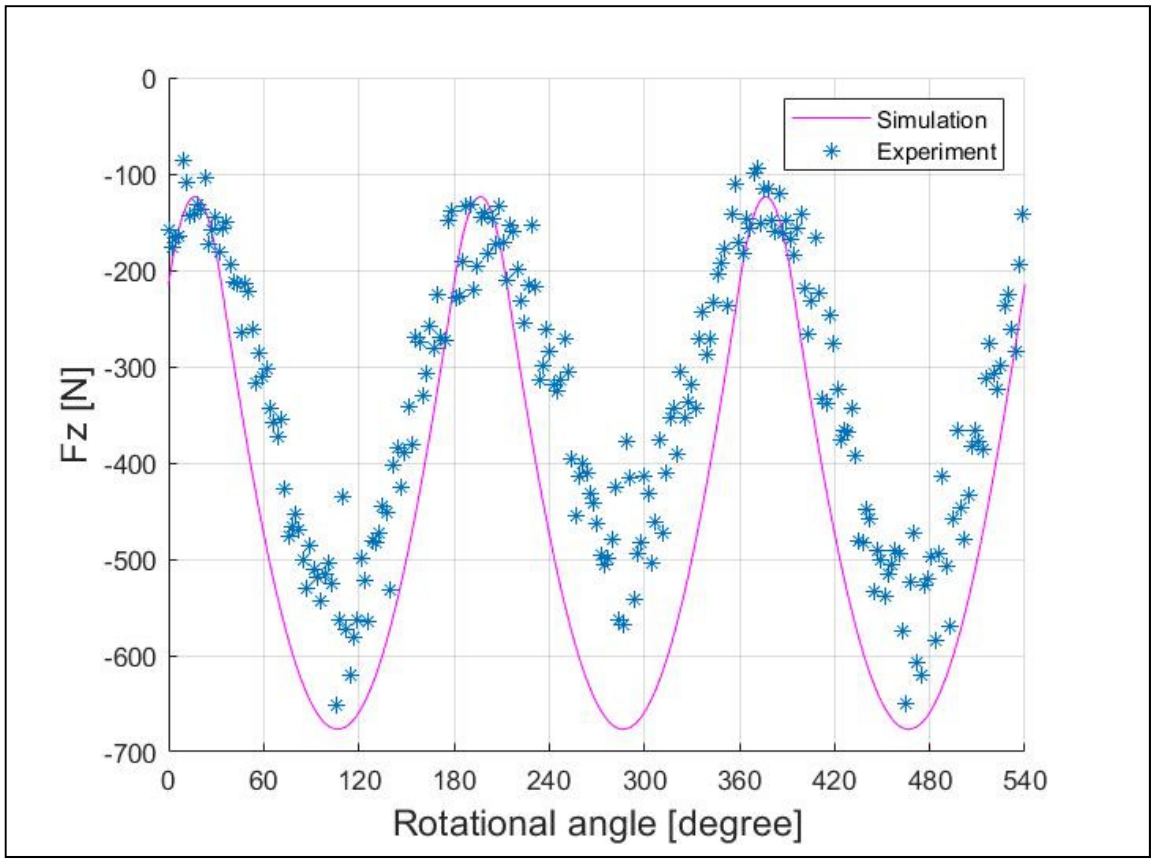


Figure 4-3 Comparing numerical and experimental cutting forces

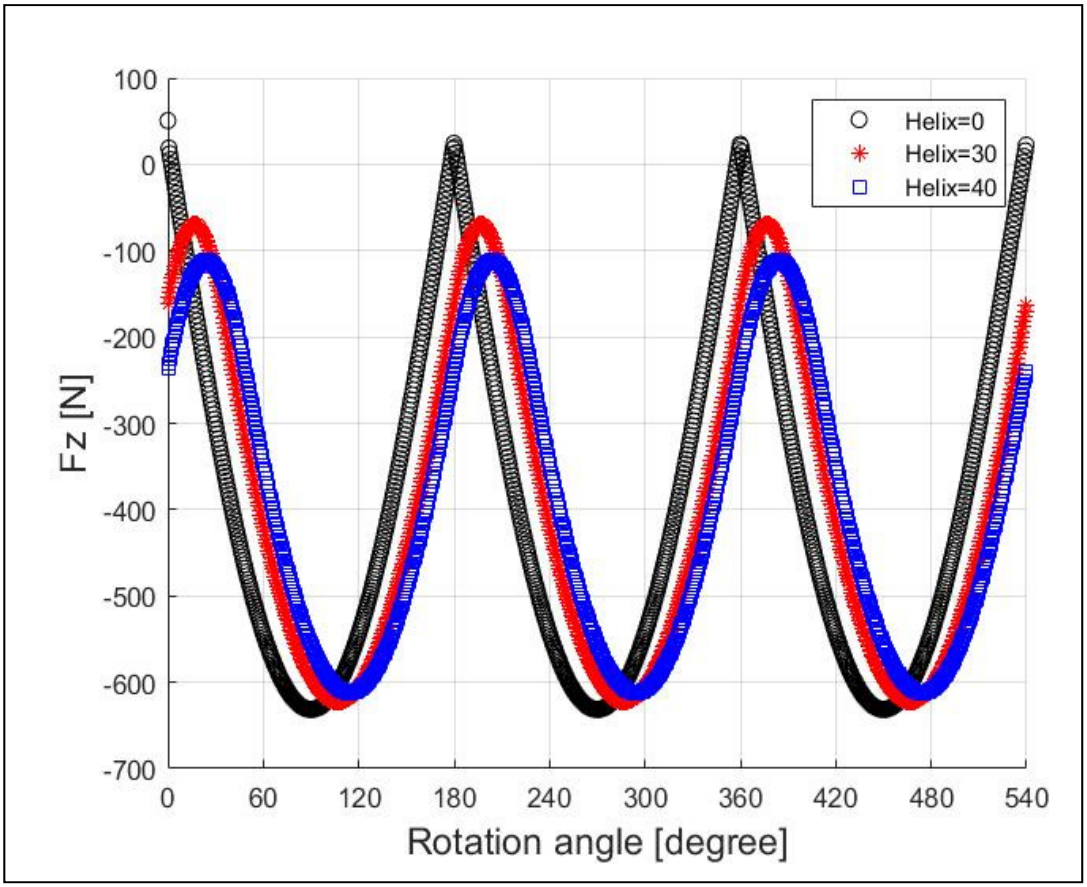
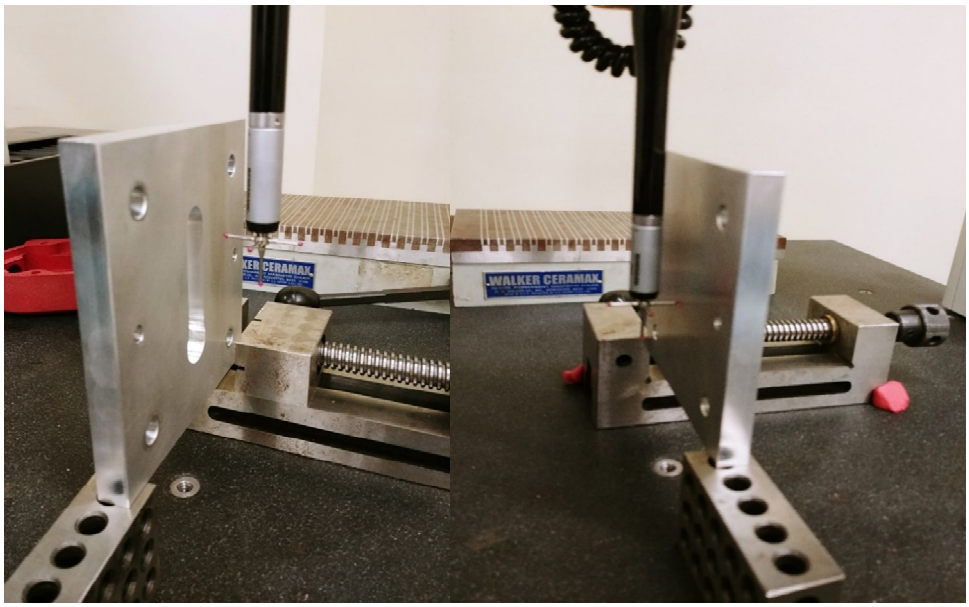


Figure 4-4 Influence of different helix angles

4.5 Measurement of the part thickness

A Mitutoyo Bright Strato 3-D Coordinate Measurement Machine (CMM) is utilized to measure the dimensions of the machined workpiece (the thickness throughout the trajectory) in order to define the geometrical profile of the cutting region by its touch signal probe (see Figure 4-5). For the sake of measurement, the part is clamped to the CMM table as illustrated in Figure 4-6.

Figure 4-5 Mitutoyo Bright Strato 3-D Coordinate Measurement Machine



(**a**). Probing from the machined face (**b**). Probing from the bottom face

Figure 4-6 Clamping and probing by CMM

The CAD file of the plate is created with the help of six holes in CMM software (PolyWorks). After calibrating, a 5-way stylus probe (a high-accuracy touch-signal probe) is used to do the probing process. Two lines of points on both sides (bottom and machined faces) at $y = 0$ and along X-direction through the trajectory have been selected to extract the points' positions as shown in Figure 4-7. Every point that CMM measures from the thin plate's surfaces is identified by its position (x, y, z coordinate) with respect to the coordinate system. The local coordinate system origin has been placed at center of the semicircle at the end of trajectory (near point B) on the top-face of the workpiece and it will eventually be corrected and moved to the center of the part and all points will be identified with respect to the new origin. This coordinate system was picked for the measurement purpose only because it had a clean surface finish at the location (at the end of trajectory) comparing to the other locations. Once all points are generated, the vertical distance among each two aligned points is measured to find the machined thickness at each position. Figure 4-8 shows this process in CMM software program in which the origin has been corrected in directions but not moved to its final location yet. The thickness of the machined part at certain position along the trajectory ($(H_i)_m$) is calculated as:

$$(H_i)_m = (z_i^b)_m - (z_i^c)_m \tag{4.1}$$

Where, $(z_i^b)_m$ and $(z_i^c)_m$ are the z-coordinates of the points at a certain position on the bottom and the cut face of the plate after the machining process, respectively.

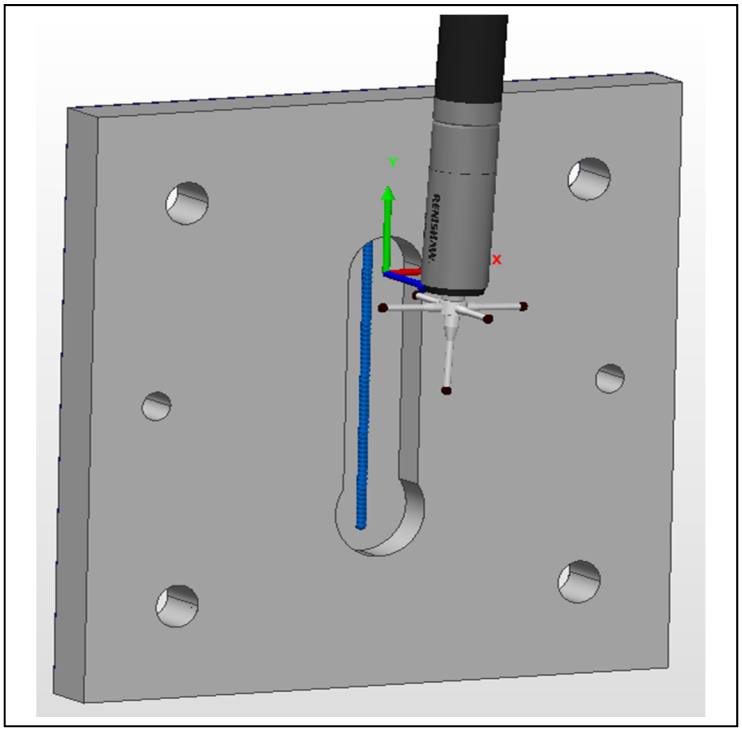


Figure 4-7 Generated points on the machined face by probing in CMM program (PolyWorks)

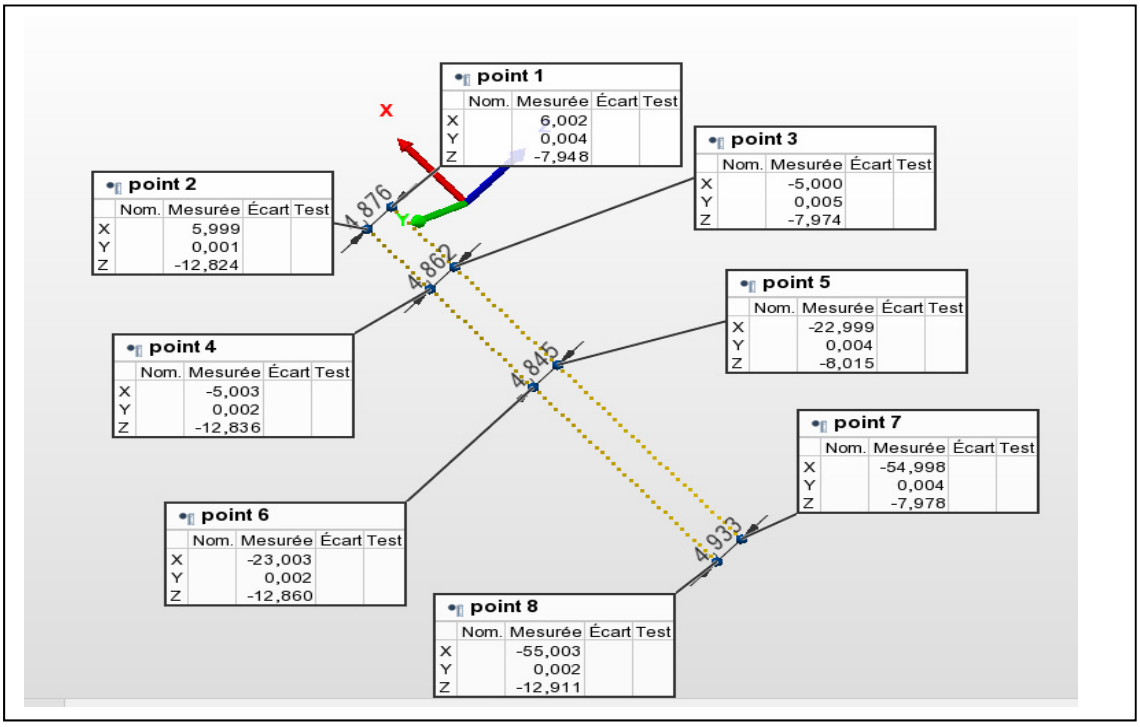


Figure 4-8 Measuring the machined thickness in CMM software program (PolyWorks)

4.6 Results and analysis

Now that both nominal and actual thicknesses of the plate in the cutting region are identified, the geometrical error at a certain position ($(\varepsilon_i)_m$) through the trajectory can be calculated as below,

$$(\varepsilon_i)_m = (H_i)_m - H_n \tag{4.2}$$

Where, $(H_i)_m$ is the actual machined thickness at a certain position and H_n is the nominal machined thickness of the workpiece. Positive value for $(\varepsilon_i)_m$ in equation (4.2) means that there is more material left on the cutting surface (i.e., the tool has cut less, undercut) and negative value means there are less material left (i.e., the tool has cut more, overcut) when the machining process is done. In other words,

$$\text{If } (\varepsilon_i)_m > 0 \Rightarrow D_r < D_n \text{ ,} \qquad \text{If } (\varepsilon_i)_m < 0 \Rightarrow D_r > D_n \tag{4.3}$$

Where, D_r and D_n are the real and nominal depth of cuts, respectively. Figure 4-9 shows the measurement results according to equation (4.2). The points in range of -30 to -20 in X-direction represent the deviation on the lead-in pocket surface which was not involved in the slotting process. However, a small portion of the right side of it (from $x \approx -20$ to -15.9), was technically involved due to the generated cutting forces in early steps of material removal. If the diameter of the pocket is considered on X-axis, it covers a range of $x = -34.9$ (mm) to $x = -15.9$ (mm). When the cutter-workpiece engagement starts, the RDOC (radial depth of cut) successively increases which results in an increase in the average cutting force value, up to a certain position (also known as maximum RDOC). For further details, please see section 3.5 (assumption number 2).

As mentioned before, creating the lead-in pocket while the component is clamped to the testbed setup could cause an initial deviation on this area due to the part flexibility. This can also aggravate the upcoming deformation in the area because of two reasons. First, in every machining operation, the material's removal process induces inevitable residual stress which in this case intensifies the deformation for the upcoming slotting operation (Fergani, Lazoglu, Mkaddem, El Mansori, & Liang, 2014). And second, the deflection of previous layer of the machined surface affects the deviation for the new one (Chen et al., 2009). As seen in Figure 4-9, around X Position= -15.9 (mm), the impact occurs and that is where the first valley can be observed.

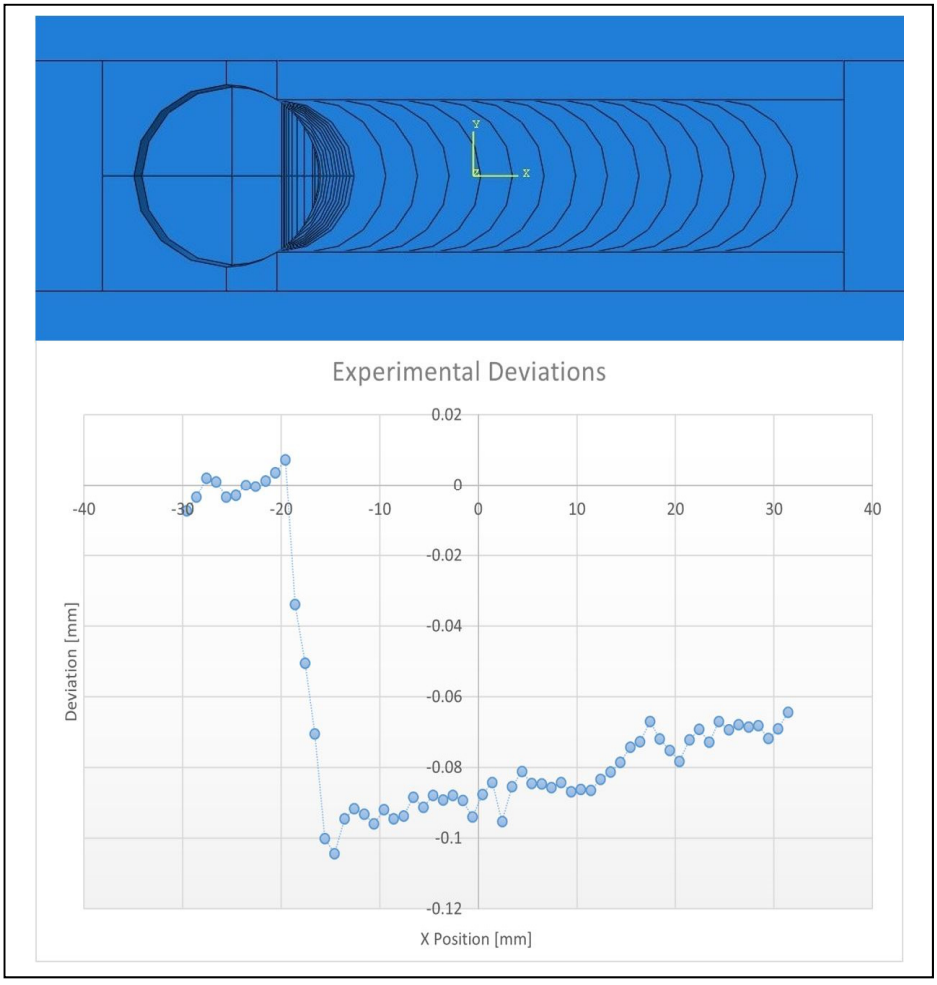


Figure 4-9 Experimental deviation

The center of the workpiece on x-axis is the next important valley, given the fact that the part is at its most flexibility at that position. By extracting both theoretical and experimental results, it is now possible to compare the deviation profiles. As shown in Figure 4-10, both profiles follow the same shape but with a great difference in value. We believe the static analysis that we chose to address this issue is the main reason of having a huge gap between the simulation and experimental results. In static analysis and in every step or removing material, Abaqus tries to converge the calculation in a way to bring the part to a steady-state while in a dynamic analysis after each chip removal, the part does not have the time to go to the steady-state and the deformation of the previous chip removal step does influence on the current one.

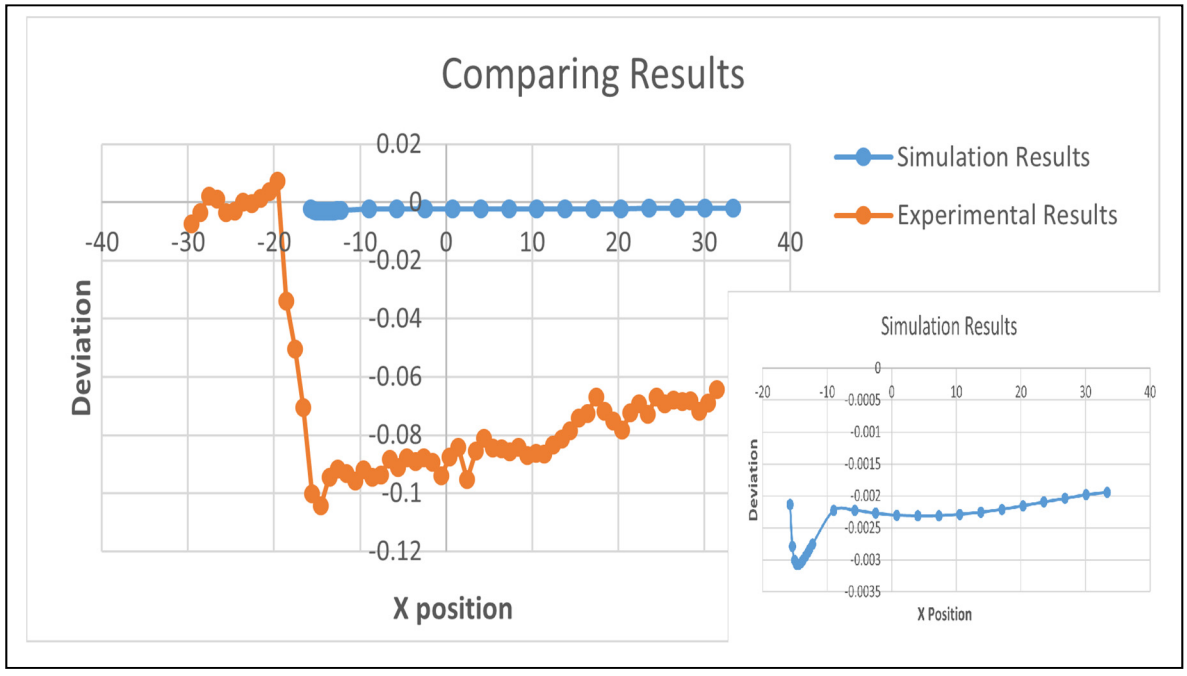


Figure 4-10 Comparing prediction and experimental results

Additionally, the material property is simple and only considers the elastic behavior of the plate. The force model in Abaqus on the other hand, plays an important role on the simulation results. Facts such as choosing a semicircle area for the distribution of the loads and having a constant average force throughout the step can result in lower displacement. Because in the experimental model, the forces act on the cutting edges periodically as the tool advances in

feed direction. Then, both assumptions for the load surfaces and constant forces do not represent this action perfectly. Subsequently, the maximum deformation in every step in Abaqus, occurs with a little distance to the actual cutting edges zone (see Figure 3-21 and Figure 3-22) and therefore, the part's behavior is different for these two force models (theorical and experimental models).

Moreover, it is not observable since the gap is too big to notice this; but, not knowing the actual value for the helix angle can also result in different value for the average cutting forces which leads to measure incorrect displacement as seen in Figure 4-4. The next chapter presents a few attempts to reduce the aforementioned gap between the theoretical and experimental results.

CHAPTER 5

FUTURE DEVELOPMENT

5.1 Introduction

As discussed in the previous chapter, static approach for a dynamic behavior is the main reason for the gap between the theoretical and experimental results. In this chapter, a few improvements are discussed which should be taken into account to reduce the gap and we will compare the results for more analysis. Basically, the Abaqus prediction model can be improved considering implicit or explicit analyses. It is recommended to utilize the explicit analysis to avoid converting a complex dynamic behavior of the part to a static one. However, since an implicit approach was studied in this work due to the simplification, mostly we cover the future works related to this matter in this segment such as considering the deformation of the previous steps of chip removals and modifying the force model in Abaqus.

5.2 Modifying the force model in Abaqus

As mentioned before in CHAPTER 3, the represented surface for each chip removal locates on the force plane parallel to X-Y plane. This caused the deformation concentration to appear with a small distance to the actual cutting region (cutting edges location). In the experiment, the cutting edges area is on a cylindrical surface as seen in Figure 5-1, where the cutter is in contact with the workpiece. Therefore, if the force model could be developed by defining a cylindrical and vertical surface for the cutting load, the deformation and the displacement results would be improved as well. The force direction in this new force model (*SURFACE TRACTION) is parallel to the surface rather than normal to it. The contact region is deformed more similar to the experiment and also comparatively with a higher value as seen in Figure 5-2. In this developed model we only manipulated the force model in Abaqus. The rest of the assumptions stay untouched. However, the improvement in terms of the displacement value,

is yet to come close to the experimental data. The approach was able to increase the displacement value of the node by166% although still far from the experimental results as seen in Table 5-1.

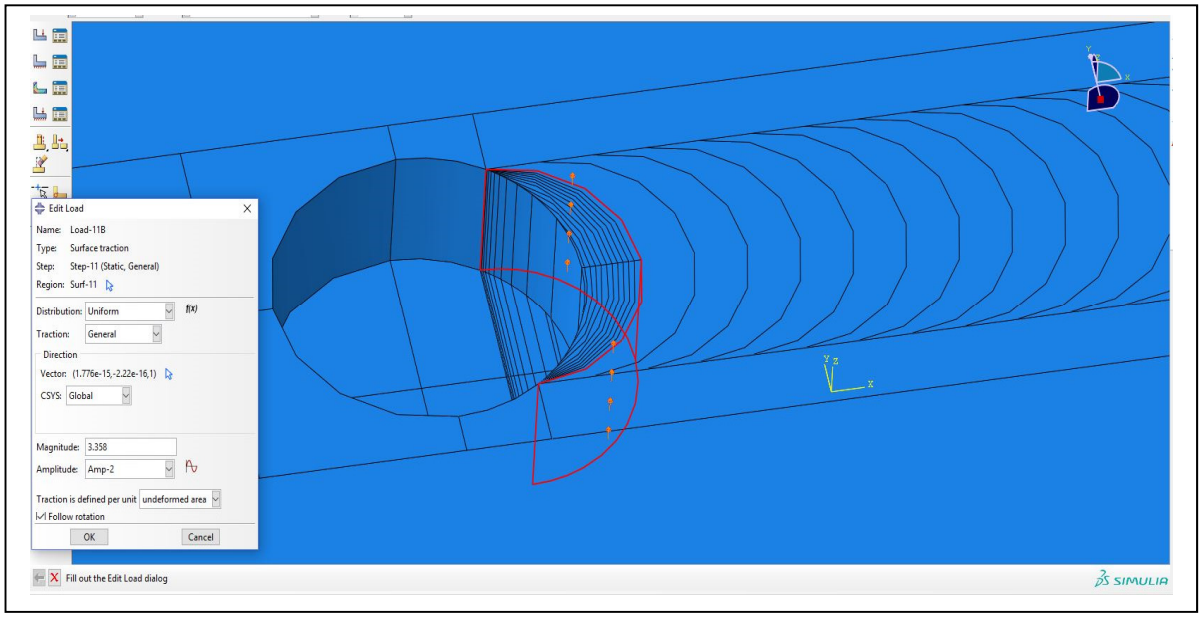


Figure 5-1 Vertical surface for defining the load

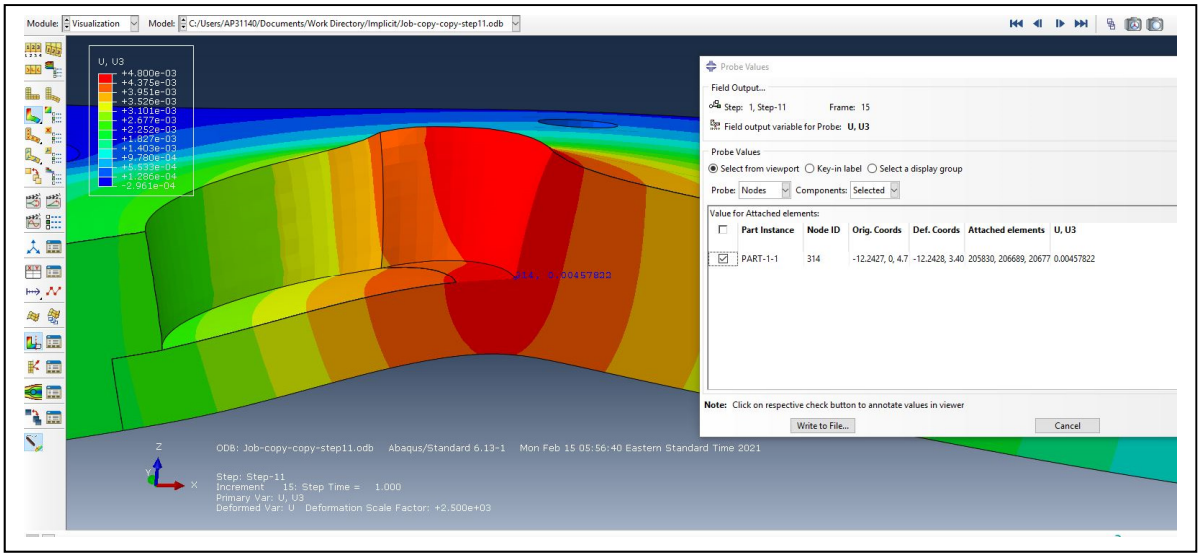


Figure 5-2 Deformed shape of the part at step 11, last frame

Table 5-1 Deviation improvement for the vertical surface approach at chip number 11

Approach	Step	Node label	X position (mm)	Vertical displacement in Abaqus (mm)	Experimental deviation at X position= -12.581 (mm)
Horizontal surface	11	314	-12.243	-0.0027	-0.0916
Vertical surface	11	314	-12.243	-0.0045	-0.0916

5.3 Dynamic behavior

In this section, we keep all the assumptions for the prediction model in CHAPTER 3 unchanged, except one. This time, we allow the previous loads in the previous steps to stay active for the first 11 steps of chip removals (see Figure 5-3). We decided to do the study only for the first 11 chips because they are partitioned with the real feedrate. Applying this assumption for the whole part, requires partitioning the part with the real feedrate throughout which can be done in the future studies.

Considering this change, we rerun the prediction model once again only for the first 11 uncut chips. Figure 5-4 shows the deviation result for these chips in a comparison with the experimental data.

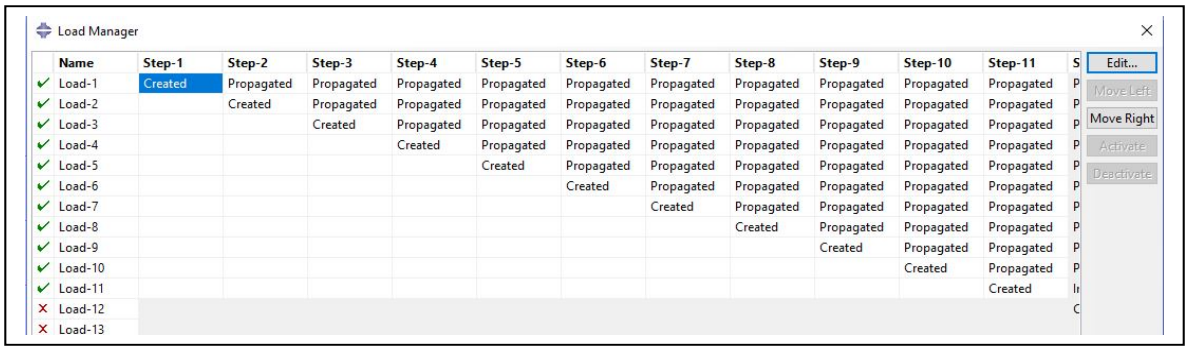


Figure 5-3 Load manager, loads stay active for the next steps

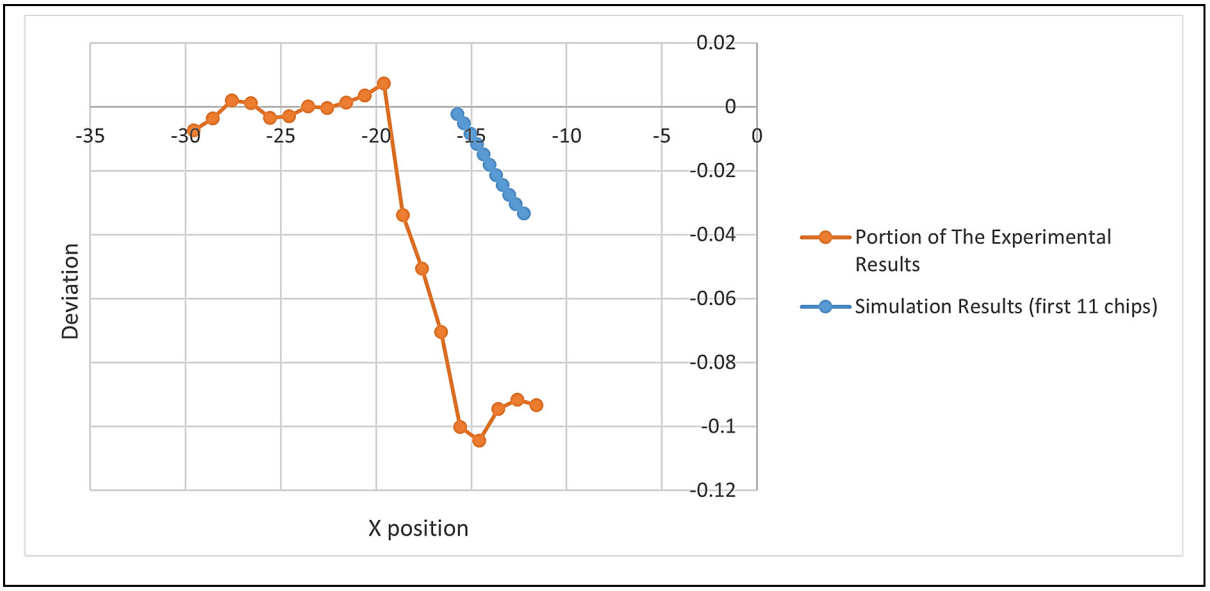


Figure 5-4 Results comparison for the developed model (dynamic behavior)

As can be seen in Figure 5-4, the improvement is remarkable. We were able to reduce the gap by 33% for step 11. If we could read and extract the displacement values for the nodes in steps 1 to 10, when they are having the maximum deformation among all available steps (by using a Python script), we would be able to further improve the deviation results for these positions as well.

5.4 Results and analysis

As seen in the sections above, both approaches can refine and accurize the results. Combining them would be also a great approach for the future works. According to the primary assumptions, each chip cell is removed at each step, and in the vertical-surface force model, the existence of the upcoming uncut chip cell is essential to define the load surface, however the previous chip cells do not exist anymore due to the chip removal action (*INTERACTION-*MODEL CHANGE). Without a surface, creating the load is impossible. Then, it is impractical to define the vertical loads for the previous steps, let alone keeping them active. Thus, for all previous steps, the horizontal force model (the one in CHAPTER 3), and for only the current step, the vertical force model (the one in section 5.2) should be defined. On the

other hand, for the dynamic behavior, we assumed all the previous deformations are at their maximum which is in contrast to the wave-like motion of the part during machining. In the experiment, the type of motion that the part is facing during the operation along the trajectory (in X-Z plane) is a wave-like one as shown in Figure 3-21 and Figure 5-2. Hence, this can be modified to a partial deformation for the previous steps (i.e., 90%U3, 80%U3, etc.) with respect to their distances with the cutting edges and the slop of the wave-like profile. Simulating that motion using a static approach is not an easy task to fulfill. Accordingly, utilizing Python as a tool is a crucial factor to carry out the work. All in all, the proposed FEA model needs to be improved in terms of applied force model in Abaqus and the type of analysis. An Implicit/dynamic or explicit/dynamic model is a more preferable option comparing to an implicit/static solution. Also, reducing the number of steps (if it is an explicit analysis) and defining a moving force along the trajectory by leveraging Python (i.e., subroutines namely Dload or Vload) can bring this model behavior closer to the accurate version.

Current practices in compensation methods are mainly divided by two approaches. The first one is that the part is treated with repetitive machining operations through feeding with different nominal depths of cut manually until it reaches to the desirable surface accuracy. This results in long processing time and lowers productivity. The second one is through FEA and NC programs which target on modifying the tool path, based on the deflection in the prediction model. But they both have a big limitation, and it is the lack of repeatability for the sake of automating optimization. In other words, they both follow a single level error compensation strategy. The proposed models were able to reduce the cutting errors by 60%. To go further than that number, an automated compensation model is needed. With a simple development and modification in the offered framework, it is possible to build and then upgrade the prediction model from a one-attempt compensation model to a fully automated process which predicts, inspects, and compensates the deformation, multiple times and iteratively until an admissible tolerance is reached (i.e., $(\varepsilon_i)_s \approx 0.001$) in one or a few scripts. Extensive compensation iterations have a high computational cost which can benefit much from modern compute infrastructures such as distributed computing and cloud computing.

CONCLUSION

This study addressed a systematic semi-automated framework for modeling a flexible milling operation of thin structures in both preprocessing and postprocessing using Python Abaqus API. In the previous works, the prediction and compensation models for geometrical errors in FEA were designed manually with help of Abaqus modules in GUI. This is a time-consuming process specially if an optimum compensation model is involved. To reach the desired surface finish with a very low tolerable surface form error, a reliable framework which considers iterative algorithms in both preprocessing and postprocessing is essential. The proposed framework facilitated the prediction modeling process in Abaqus by using several Python scripts in both preparing the model and reading the predicted results.

A Thin plate of Aluminum was designed in Abaqus. The cutting and boundary conditions were defined. Removing material process and applying the generated cutting forces with respect to the tool's positions were simulated. Several Python scripts aided to facilitate these processes. The deformation and displacement values for specific nodes on the machined area at certain positions of the tool were collected automatically in an Excel file by a Python script.

Next, an experimental test with the same cutting conditions as in Abaqus model was carried out to validate the numerical results. The workpiece is fixed on a flexible setup configuration for slotting operation using an end mill tool by a CNC machine. Then, the thickness of machined surface was measured along the trajectory by a Mitutoyo Bright Strato 3-D Coordinate Measurement Machine (CMM) using its touch signal probing technique. Both data from simulation and measurement were collected to define the geometrical errors of the cutting region after the machining test.

The prediction model then was developed considering two important factors. First, a vertical surface for the force model in Abaqus was studied. And then, a dynamic behavior assumption

was applied to the prediction model to take into account the influence of the deformation of the previous chip removals. This aspect was able to diminish the gap between the numerical and experimental results significantly. The modified force model in Abaqus could add 160% to the displacement magnitude and the dynamic behavior approach was able to improve the results by 33%.

APPENDIX I

MATLAB SCRIPT

Cutting_force_different_phi.m

```
clc;
clear all;
clear variables;
%Input
dataset=xlsread('8mmDOC_Copy','sheet2','e9:i244');
a=8;          %depth of cut [mm]
N=2;          %numbers of flutes
f=0.3386;     %feedrate [mm/rev]
c=f/N;        %feedrate [mm/tooth]
%s= 4584;     %Spindle Speed [rpm]
D=15.875;     %cutter diameter  [mm]
beta=pi/6;    %helix angle (30)
K_ac=-485.007;  K_ae=-3.6147;   %Kac=pi*Fzc/N*a, Kae=2*Fze/N*a   %cutting coefficients related to edge rubbing and chip shearing mechanisms,respectively.
dphi=0.01;       %Integration angle [Rad]
da=0.01;         %Integration height [mm]
%%
%Variable
phip= 2*pi/N;    %Cutter pitch angle
K= 4*pi/dphi;    %number of integration angular steps
L=a/da;          %number of integration height steps
%Variables for the 1st 12 chips
ps=(pi/180)*[61.67, 44.21, 34.04, 26.79, 21.16, 16.55, 12.62, 9.18, 6.11, 3.31, 0.72, 0]; % 12 entry angles up to full immersion [Rad] - Radial immersion
p=pi*ones(1,12);
px=p-ps;          % exit angles= pi-(entry angle)  [Rad]
%%
for angle_i = 1:12    % Angle_i is the loop index. It is the cutter entry immersion angle for the very first chip removals up to full-imerssion
    phist=ps(angle_i);  % Entry angle
    phiex=px(angle_i);  % Exit angle
    Fxii=0;Fyii=0;Fzii=0;

    for i=1:K
        phi(i)= phist+(i-1)*dphi;    %Cutting edge positions
        phidegree(i)= phi(i)*180/pi; %converting from Radian to Degree
        Fx(i)=0;Fy(i)=0;Fz(i)=0;

        for k=1:N                    %Calculating the force contributions of all TEETH
           phi1= phi(i)+(k-1)*phip;  %Phi1 (immersion angle) for each flute

           while phi1>2*pi, phi1=phi1-2*pi;  % Keeping phi1(i) value between 0 and 2*pi if it passes 2*pi
           end

           for j=1:L    %Integrate along the axial depth of cut
           aa(j)=j*da;  %Axial position
           psi = 2*tan(beta)*aa(j)/D;   %Psi: Radial Lag angel caused by the local helical angle
           phi2= phi1-(psi);            %Phi2: instantaneous Radial immersion angle (Update the immersion angle due to helix)
           % a cutting edge point that is axially z above will have an immersion angle of (phi-psi)
           % The lag angle "Psi" is the cutting edge projected position on XY plane because of the
           % helix angle at the level z.
            while phi2>2*pi, phi2=phi2-2*pi;     % Keeping phi2(i) value between 0 and 2*pi if it passes 2*pi
            end
               if (phi2>=phist)&&(phi2<=phiex)  %if the edge is cutting at this height with phi2 immersion angle then calculate the force
               h=c*sin(phi2);                   %instantaneous undeformed chip thickness at this point, C=feed [mm/tooth]
               %dFt=da*(K_tc*h+K_te);
               %dFr=da*(K_rc*h+K_re);
               dFa=da*(K_ac*h+K_ae);
               %dFx=-dFt*cos(phi2)-dFr*sin(phi2);
               %dFy=dFt*sin(phi2)-dFr*cos(phi2);
               %dFz=-dFa;
               %Fx(i)=Fx(i)+dFx;  %summing the cutting forces contributed by all the active edges for each axial immersion
               %Fy(i)=Fy(i)+dFy;  %Fx,Fy,Fz are variabales containing all the cutting forces for each immersion angle after the loop is done
               Fz(i)=Fz(i)+dFa;
               else
               end
```

Figure A-1 Instantaneous cutting force model, page 1 of 2

```
    for i=1:K
        phi(i)= phist+(i-1)*dphi;    %Cutting edge positions
        phidegree(i)= phi(i)*180/pi; %converting from Radian to Degree
        Fx(i)=0;Fy(i)=0;Fz(i)=0;

        for k=1:N                    %Calculating the force contributions of all TEETH
           phi1= phi(i)+(k-1)*phip;  %Phi1 (immersion angle) for each flute

           while phi1>2*pi, phi1=phi1-2*pi;  % Keeping phi1(i) value between 0 and 2*pi if it passes 2*pi
           end

           for j=1:L    %Integrate along the axial depth of cut
           aa(j)=j*da;  %Axial position
           psi = 2*tan(beta)*aa(j)/D;   %Psi: Radial Lag angel caused by the local helical angle
           phi2= phi1-(psi);            %Phi2: instantaneous Radial immersion angle (Update the immersion angle due to helix)
           % a cutting edge point that is axially z above will have an immersion angle of (phi-psi)
           % The lag angle "Psi" is the cutting edge projected position on XY plane because of the
           % helix angle at the level z.
            while phi2>2*pi, phi2=phi2-2*pi;     % Keeping phi2(i) value between 0 and 2*pi if it passes 2*pi
            end
                if (phi2>=phist)&&(phi2<=phiex)  %if the edge is cutting at this height with phi2 immersion angle then calculate the force
                h=c*sin(phi2);                   %instantaneous undeformed chip thickness at this point, C=feed [mm/tooth]
                %dFt=da*(K_tc*h+K_te);
                %dFr=da*(K_rc*h+K_re);
                dFa=da*(K_ac*h+K_ae);
                %dFx=-dFt*cos(phi2)-dFr*sin(phi2);
                %dFy=dFt*sin(phi2)-dFr*cos(phi2);
                %dFz=-dFa;
                %Fx(i)=Fx(i)+dFx;  %summing the cutting forces contributed by all the active edges for each axial immersion
                %Fy(i)=Fy(i)+dFy;  %Fx,Fy,Fz are variabales containing all the cutting forces for each immersion angle after the loop is done
                Fz(i)=Fz(i)+dFa;
                else
                end
           end
        end
    %Fxii=Fxii+Fx(i);    %suming the cutting forces for all angular increments after 2 rotations (4pi)
    %Fyii=Fyii+Fy(i);
    Fzii=Fzii+Fz(i);
    end
%ave_force_X=Fxi/K;
%ave_force_Y=Fyi/K;
ave_force_Z(angle_i)=Fzii/K;    % Average cutting force
ave_force_Z
end
%% Plotting MATLAB script
hold on;
%plot(phidegree,Fx);
%plot(phidegree,Fy,'red');
plot(phidegree,Fz,'magenta')
xlim([0 540]);
xbounds = xlim();
set(gca, 'xtick', xbounds(1):60:xbounds(2));
grid on;
xlabel('Rotational angle [degree]','FontSize',14);
ylabel('Fz [N]','FontSize',14);
%% Plotting experimental results
xx=dataset(:,5);  % Rotaional angle [degree]
yy=dataset(:,1);  %Fz   [N]
plot(xx,yy,'*');  %angle,Fz in 8mmDOC_Copy.xlsx
hleg = legend('Simulation','Experiment');
set(findobj(gcf,'Type','text'),'FontSize',14)
```

Figure A-2 MATLAB script, page 2 of 2

APPENDIX II

CHIP CELLS

```
#-------------------------------------------------
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# The X-positions of the edges for Chip-12th
edge1_x13= -19.366296
edge2_x13= -17.142671
edge3_x13= -13.886971
edge4_x13= -9.591227
edge5_x13= -12.846927
edge6_x13= -19.366296

MDL_RA=mdb.models['26Chips'].rootAssembly

for i in range(14):     # Total chips=25  (25-11=14)  The C-shape edges for the first 11 chips are not equidistant.
                        # I do the first 11 chips manually in ABAQUS CAE and in *ASSEMBLY interface.
                        # The loop here is for the rest of them
        edge1_x = (i*3.2557) + edge1_x13
        edge2_x = (i*3.2557) + edge2_x13
        edge3_x = (i*3.2557) + edge3_x13
        edge4_x = (i*3.2557) + edge4_x13
        edge5_x = (i*3.2557) + edge5_x13
        edge6_x = (i*3.2557) + edge6_x13

        MDL_RA.PartitionCellBySweepEdge(cells=          #Object to creat a cell in *ASSEMBLY
        MDL_RA.instances['Part-1-1'].cells.findAt(
        ((37.,0.,10.7), )), edges=(
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge1_x, 7.9375, 12.7), ),
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge2_x, -7.333294, 12.7), ),
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge3_x, -7.333294, 12.7), ),
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge4_x, 3.03755, 12.7), ),
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge5_x, 3.03755, 12.7), ),
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (edge6_x, -7.9375, 12.7), )), sweepPath=
        MDL_RA.instances['Part-1-1'].edges.findAt(
        (-60.0, -12.0, 6.7), ))
MDL_RA.regenerate()
```

Figure A-3 Partitioning chip cells

APPENDIX III

RENAMING PARTITIONS

```
# Renaming partition Cells:

from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

MDL_RA=mdb.models['Model-Cells'].rootAssembly
Cellname= "Chip-Cell-"
ii=0

for i in range(30,165):             #changing defult partition names to something trackable (partition cell-30 => Chip-Cell 12),
                                    # range(31,165) => if we take the actual feedrate
    MDL_RA.features.changeKey(fromName='Partition cell-'+str(i), toName=Cellname+str(12+ii))
    ii=ii+1
```

Figure A-4 Renaming default partition cells (for chip cells)

APPENDIX IV

MULTIPLE SURFACES

```
'''
mdb.models['25Chips'].rootAssembly.Surface(name='Surf-15', side1Faces=
    mdb.models['25Chips'].rootAssembly.instances['Part-1-1'].faces.findAt(((
    -10.152702, -2.55773, 4.7), ), ((-10.152702, 2.55773, 4.7), ), ((-10.04161,
    7.678482, 4.7), ), ((-3.641302, 2.55773, 4.7), ), ((-6.897002, 2.55773,
    4.7), ), ((-9.999118, 6.106214, 4.7), ), ((-6.897002, -2.55773, 4.7), ), ((
    -9.999118, -6.106214, 4.7), ), ((-10.04161, -7.678482, 4.7), ), ((
    -3.641302, -2.55773, 4.7), ), ((-5.175748, 0.533252, 4.7), ), ((-4.565882,
    -2.046364, 4.7), ), ))
'''
face1_x14=-10.152702   # X-Corodinates of selected faces for Chip Number 14 as a reference.
face2_x14=-10.152702   # Since Y and Z-Coordinates of these faces dont change then no need to put them in a loop.
face3_x14=-10.04161
face4_x14=-3.641302
face5_x14=-6.897002
face6_x14=-9.999118
face7_x14=-6.897002
face8_x14=-9.999118
face9_x14=-10.04161
face10_x14=-3.641302
face11_x14=-5.175748
face12_x14=-4.565882

MDL_RA= mdb.models['25Chips'].rootAssembly
ii=14
for i in range(12):                                 # Creating Surfaces from chip-14 to chip-25
        face1_x = (i*3.2557) + face1_x14
        face2_x = (i*3.2557) + face2_x14
        face3_x = (i*3.2557) + face3_x14
        face4_x = (i*3.2557) + face4_x14
        face5_x = (i*3.2557) + face5_x14
        face6_x = (i*3.2557) + face6_x14
        face7_x = (i*3.2557) + face7_x14
        face8_x = (i*3.2557) + face8_x14
        face9_x = (i*3.2557) + face9_x14
        face10_x = (i*3.2557) + face10_x14
        face11_x = (i*3.2557) + face11_x14
        face12_x = (i*3.2557) + face12_x14

        MDL_RA.Surface(name='Surf-'+str(ii+i), side1Faces=
            MDL_RA.instances['Part-1-1'].faces.findAt(((
            face1_x, -2.55773, 4.7), ), ((face2_x, 2.55773, 4.7), ), ((face3_x,
            7.678482, 4.7), ), ((face4_x, 2.55773, 4.7), ), ((face5_x, 2.55773,
            4.7), ), ((face6_x, 6.106214, 4.7), ), ((face7_x, -2.55773, 4.7), ), ((
            face8_x, -6.106214, 4.7), ), ((face9_x, -7.678482, 4.7), ), ((
            face10_x, -2.55773, 4.7), ), ((face11_x, 0.533252, 4.7), ), ((face12_x,
            -2.046364, 4.7), ), ))

MDL_RA.regenerate()
```

Figure A-5 Creating multiple equidistance surfaces

APPENDIX V

STEPS AND LOADS

```python
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

MDL= mdb.models['25Chips-Copy']
MDL_RA= mdb.models['25Chips-Copy'].rootAssembly

for i in range(12,26):
    #creating the remaining steps (the first 11 steps for the first chip cells were done in GUI)
    MDL.StaticStep(initialInc=0.01, maxInc=0.1, maxNumInc=10000,
    minInc=1e-60, name='Step-'+str(i), previous='Step-'+str(i-1))
    # creating the load with restpect to the step's number and also deactivating the previous load
    MDL.Pressure(amplitude='Amp-2', createStepName='Step-'+str(i),
        distributionType=TOTAL_FORCE, field='', magnitude=-442.955, name='Load-'+str(i)
        , region=MDL_RA.surfaces['Surf-'+str(i)])
    MDL.loads['Load-'+str(i-1)].deactivate('Step-'+str(i))

MDL_RA.regenerate()
```

Figure A-6 Steps and loads

APPENDIX VI

INTERACTION

```
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

MDL= mdb.models['25Chips-Copy']
MDL_RA= mdb.models['25Chips-Copy'].rootAssembly

XCell_12= -17.8     #X-cordinate of chip cell 12th

for i in range(14): #The chip cells follow a pattern in size and volume after chip cell 11th up to the end.
                    #Then it is possible to make a loop with respect to their X-cordinates at the same
                    # y and z positions.
    XCell = (i*3.2557) + XCell_12

    MDL.ModelChange(activeInStep=False, createStepName='Step-'+str(i+12),
    includeStrain=False, name='Int-'+str(i+12), region=Region(
    cells=MDL_RA.instances['Part-1-1'].cells.findAt(
    ((XCell, -7.736098, 12.7), ), )))

MDL_RA.regenerate()
#Creatin a Job, submitting and waiting till the analysis is done.
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='25Chips-Copy', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Job-25chips-Copy', nodalOutputPrecision=
    SINGLE, numCpus=11, numDomains=11, numGPUs=0, queue=None, scratch='', type=
    ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Job-25chips-Copy'].submit(consistencyChecking=OFF)
mdb.jobs['Job-25chips-Copy'].waitForCompletion()
```

Figure A-7 Interaction loop (*MODEL CHANGE) and submitting the *Job

APPENDIX VII

DISPLACEMENT

```python
# ================================================Extract U at some nodes=====================================
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=219.796875,
    height=178.784255981445)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
import odbAccess
import csv

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)
odb = session.openOdb(
    name='C:/Users/AP31140/Documents/Work Directory/Implicit/Job-25chips-Copy.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=odb)

#Create a variable that refers to the node labels (The nodes are in order, from chip 1 to 25)
My_node_set=[318, 328, 332, 338, 342, 348, 352, 358, 365, 371, 314, 307, 303, 294, 289, 279, 275, 265, 261, 252, 247, 237, 233, 227, 223]
Z_list=[]

for i in range(25):
    # Steps
    S='Step-'+str(i+1)
    #===Creating a variable that refers to the last frame of each step====
    lastFrame=odb.steps[S].frames[-1]

    #===Creating a variable which refers to disp in last frame of each step===
    displacement= lastFrame.fieldOutputs['U']

    #===Creating a variable that refers to the Z values of the node set in the last frame of each step===
    u= displacement.values
    for v in u:
        if v.nodeLabel == My_node_set[i]:
            #x = v.data[0]      #U1
            #y = v.data[1]      #U2
            z = v.data[2]       #U3 (Displacement in z-direction)
            #mag = ( x**2 + y**2 ) **0.5        #Magnitude
            #print (v.nodeLabel, z)
            Z_list.append(z)
joined_list = list(zip(My_node_set, Z_list))
# "with open" creates an Excel file and stores the displacements in a colomn in front of their node labels
with open(r"C:\Users\AP31140\Desktop\Usefull files\Coding Python\results.csv", mode="w") as f:
    wr = csv.writer(f, quoting=csv.QUOTE_ALL)
    wr.writerows(joined_list)
```

Figure A-8 Displacement in Z-direction (U3)

LIST OF BIBLIOGRAPHICAL REFERENCES

Adetoro, M., & Wen, P. H. (2008). Simulation of end milling on FEM using ALE formulation. Dans *Abaqus user's conference* (pp. 1-19).

Aoyama, T., & Kakinuma, Y. (2005). Development of Fixture Devices for Thin and Compliant Workpieces. *CIRP Annals, 54*(1), 325-328. doi: 10.1016/s0007-8506(07)60114-0

Asada, H., & By, A. (1985). Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE Journal on Robotics and Automation, 1*(2), 86-94.

Bao, Y., Wang, B., He, Z., Kang, R., & Guo, J. (2021). Recent progress in flexible supporting technology for aerospace thin-walled parts: A review. *Chinese Journal of Aeronautics*. doi: 10.1016/j.cja.2021.01.026

Bolar, G., & Joshi, S. N. (2017). Three-dimensional numerical modeling, simulation and experimental validation of milling of a thin-wall component. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231*(5), 792-804. doi: 10.1177/0954405416685387

Budak, E., Altintas, Y., & Armarego, E. J. A. (1996). Prediction of milling force coefficients from orthogonal cutting data.

Cai, W., Hu, S. J., & Yuan, J. X. (1996). Deformable sheet metal fixturing: principles, algorithms, and simulations. *118*(3). doi: 10.1115/1.2831031

Canadian Metalworking. (2010*). Kostyrka universal holding fixture (UHF)* [Photograph]. Canadian Metalworking. https://www.canadianmetalworking.com/canadianmetalworking/product/metalworking/universal-holding-fixture

Chen, W., Xue, J., Tang, D., Chen, H., & Qu, S. (2009). Deformation prediction and error compensation in multilayer milling processes for thin-walled parts. *International Journal of Machine Tools and Manufacture, 49*(11), 859-864. doi: 10.1016/j.ijmachtools.2009.05.006

Del Sol, I., Rivero, A., Lopez de Lacalle, L. N., & Gamez, A. J. (2019). Thin-Wall Machining of Light Alloys: A Review of Models and Industrial Approaches. *Materials (Basel), 12*(12). doi: 10.3390/ma12122012. Repéré à https://www.ncbi.nlm.nih.gov/pubmed/31234596

Denkena, B., Schmidt, C., & Krüger, M. (2010). Experimental investigation and modeling of thermal and mechanical influences on shape deviations in machining structural parts. *International Journal of Machine Tools and Manufacture, 50*(11), 1015-1021. doi: 10.1016/j.ijmachtools.2010.06.006

Diez, E., Perez, H., Marquez, J., & Vizan, A. (2015). Feasibility study of in-process compensation of deformations in flexible milling. *International Journal of Machine Tools and Manufacture, 94*, 1-14. doi: 10.1016/j.ijmachtools.2015.03.008

Escamilla, I., Zapata, O., Gonzalez, B., Gámez, N., & Guerrero, M. (2010). Finite Element Simulation Of The Milling Process Of A TI6AL4V Alloy. Dans *SIMULIA Customer Conference*.

Feng, H. Y., & Menq, C. H. (1996). A Flexible Ball-End Milling system model for cutting force and machining error prediction.

Fergani, O., Lazoglu, I., Mkaddem, A., El Mansori, M., & Liang, S. Y. (2014). Analytical modeling of residual stress and the induced deflection of a milled thin plate. *The International Journal of Advanced Manufacturing Technology, 75*(1-4), 455-463. doi: 10.1007/s00170-014-6146-3

Fields, A., Youcef-Toumi, K., & Asada, H. (1989). Flexible fixturing and automatic drilling of sheet metal parts using a robot manipulator. *Robotics and Computer-Integrated Manufacturing, 5*(4), 371-380. doi: 10.1016/0736-5845(89)90010-0

Izamshah, R., Yuhazri, M. Y., Hadzley, M., Ali, M. A., & Subramonian, S. (2013). Effects of End Mill Helix Angle on Accuracy for Machining Thin-Rib Aerospace Component. *Applied Mechanics and Materials, 315*, 773-777. doi: 10.4028/www.scientific.net/AMM.315.773

Izamshah R.A, R., Mo, J., & Ding, S. L. (2010). Finite Element Analysis of Machining Thin-Wall Parts. *Key Engineering Materials, 458*, 283-288. doi: 10.4028/www.scientific.net/KEM.458.283

Ji, W., Qin, G. H., & Ye, H. C. (2011). An Advanced FEA Based Analytical Strategy for the Milling of Overall Thin-Walled Workpieces. *Advanced Materials Research, 308-310*, 499-502. doi: 10.4028/www.scientific.net/AMR.308-310.499

JulianBauer. (2017, November 14). Abaqus Surface getSequenceFromMask [Msg 2]. Message posted to website address. https://stackoverflow.com/questions/47293330/abaqus-surface-getsequencefrommask

Kang, Y.-G., & Wang, Z.-Q. (2013). Two efficient iterative algorithms for error prediction in peripheral milling of thin-walled workpieces considering the in-cutting chip.

International Journal of Machine Tools and Manufacture, 73, 55-61. doi: 10.1016/j.ijmachtools.2013.06.001

Kang, Y. G., Yang, G. R., Huang, J., & Zhu, J. H. (2014). Systematic Simulation Method for the Calculation of Maximum Deflections in Peripheral Milling of Thin-Walled Workpieces with Flexible Iterative Algorithms. *Advanced Materials Research, 909*, 185-191. doi: 10.4028/www.scientific.net/AMR.909.185

Lowell, W. F., 1982, "Geo-metrics II: Dimensioning and Tolerancing," ANSI/ ASME Standard, Y13.5M, pp. 35-52.

Ma, H., Duan, H., & Tang, A. (2010). Modeling and simulation of deformation of milling thin-walled part. Dans *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)* (Vol. 2, pp. 433-436). IEEE.

MTorres. (n.d.). [TORRESMILL 5-axes gantry DNC-CNC high speed milling machine]. Retrieved May 10, 2021, https://www.mtorres.es/en/aeronautics/products/carbon-fiber/torresmill

Nguyen, S. Q. (2016). *Prediction and compensation of geometrical errors in milling process of thin components using a flexible configuration setup* (École de technologie supérieure).

Nguyen, S. Q., & Chatelain, J. F. (2014). Study of the Milling Process for Thin Components Using a Flexible Setup Configuration. *Applied Mechanics and Materials, 704*, 27-31. doi: 10.4028/www.scientific.net/AMM.704.27

Prabhaharan, G., Padmanaban, K. P., & Krishnakumar, R. (2006). Machining fixture layout optimization using FEM and evolutionary techniques. *The International Journal of Advanced Manufacturing Technology, 32*(11-12), 1090-1103. doi: 10.1007/s00170-006-0441-6

Rai, J. K., & Xirouchakis, P. (2008). Finite element method based machining simulation environment for analyzing part errors induced during milling of thin-walled components. *International Journal of Machine Tools and Manufacture, 48*(6), 629-643. doi: 10.1016/j.ijmachtools.2007.11.004

Ratchev, S., Govender, E., Nikov, S., Phuah, K., & Tsiklos, G. (2003). Force and deflection modelling in milling of low-rigidity complex parts. *Journal of Materials Processing Technology, 143-144*, 796-801. doi: 10.1016/s0924-0136(03)00382-0

Ratchev, S., Liu, S., & Becker, A. A. (2005). Error compensation strategy in milling flexible thin-wall parts. *Journal of Materials Processing Technology, 162-163*, 673-681. doi: 10.1016/j.jmatprotec.2005.02.192

Ratchev, S., Liu, S., Huang, W., & Becker, A. A. (2006). An advanced FEA based force induced error compensation strategy in milling. *International Journal of Machine Tools and Manufacture, 46*(5), 542-551. doi: 10.1016/j.ijmachtools.2005.06.003

Schmitz, T. L., Bayly, P. V., Soons, J. A., & Dutterer, B. (2001). Prediction of surface location error by time finite element analysis and euler integration. Dans *Proceedings of the 17th Annual ASPE Meeting, October 20* (Vol. 25, pp. 132-137). Citeseer.

Sela, M. N., Gaudry, O., Dombre, E., & Benhabib, B. (1997). A reconfigurable modular fixturing system for thin-walled flexible objects. *The International Journal of Advanced Manufacturing Technology, 13*(9), 611-617.

Serafettin, Engin. (2018). *Metal Machining and Surface Technology* [Syllabus]. Montreal, QC: Department of Mechanical And Industrial Engineering, Concordia University.

Smith, M. (2009). *ABAQUS/Standard User's Manual, Version 6.9*. Dassault Systèmes Simulia Corp.

Tsai, J. S., & Liao, C. L. (1999). Finite-element modeling of static surface errors in the peripheral milling of thin-walled workpieces. *Journal of Materials Processing Technology, 94*(2-3), 235-246.

Y. Altintas, *Manufacturing Automation*. 2012: Cambridge University Press

Zhang, Y., Gao, S., Yang, N., Jiang, X., Zhao, G., & Liu, X. (2021). doi: 10.21203/rs.3.rs-603290/v1