UNIVERSITÉ DE MONTRÉAL

TOOL PATH GENERATION FOR ROUGHING INTEGRALLY BLADED ROTORS USING CUP MILL CUTTERS

ADRIAN AUGUSTIN PAUL
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
MAI 2007

© Adrian Augustin Paul, 2007.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-48934-5
Our file Notre référence
ISBN: 978-0-494-48934-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

TOOL PATH GENERATION FOR ROUGHING INTEGRALLY BLADED ROTORS USING CUP MILL CUTTERS

présenté par : PAUL Adrian Augustin
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment acceptée par le jury d'examen constitué de :

M. BARON Luc, Ph.D., présidente
M. BALAZINSKI Marek, Ph.D., membre et directeur de recherche
M. SASU Ioan, Ph.D., membre et codirecteur de recherche
M. CHÂTELAIN Jean-François, Ph.D., membre

ACKNOWLEDGEMENTS

Ideas only come as a product of interactions. I would like to begin by acknowledging those who have actively provided support to this research. In particular I would like to thank:

Marek Balazinski, my research manager at École Polytechnique de Montréal for all the support and understanding.

Ioan Sasu, my project coordinator at Pratt & Whitney Canada for giving me the opportunity to investigate a wide variety of topics and for his patience to share his technical knowledge.

Pratt & Whitney Canada, for all the material and financial support without which this project could not have been possible and for seeing a potential in this project.

Bertrand Turcott, and all the PWC technicians that helped me to bring this project to term by providing ideas and inspiration.

My dad, who helped me with the revision of this paper.

ABSTRACT

This paper presents a novel approach to rough integrally bladed rotors, using a new type of tool, the cup mill cutter. A particular attention is accorded to the kinematics of this tool. The currently used endmills at Pratt & Whitney Canada are fragile and, consequently, unproductive. In the same time, they are produced in-house, with relatively high manufacturing costs. Our proposed tool, on the other hand, is a rigid tool, more economic and possesses a good chip evacuation potential. Due to the particular geometry of this new tool and the different tool kinematics, this paper proposes also an algorithm capable to generate the tool path to rough compressor blades using the cup mill cutter. This algorithm is applicable to different part numbers, using different tool sizes. The 5-axis tool path generated permits to remove a maximum volume of material, using an efficient tool kinematics. Our work demonstrates the good potential of this new method for roughing IBRs to be used broadly in production.

RÉSUMÉ

Cette recherche propose une nouvelle méthode de dégrossissage de rotors aubage monobloc, en utilisant un nouveau type d'outil de fraisage cylindrique – cup mill. Une attention particulière est accordée à la cinématique de l'outil. Les outils présentement utilisés chez Pratt & Whitney Canada sont fragiles et, par conséquent, non productifs. De plus, ces outils son fabriqués directement par la compagnie avec des coûts de production élevés. Par contre, l'outil que nous proposons est rigide, économique et permet une bonne évacuation des copeaux. Dû à la géométrie particulière de cet outil et à la différente cinématique utilisée, cette recherche propose un algorithme capable de générer le trajet de l'outil pour usiner (dégrossissage) les pales de rotors aubage monoblocs. Cet algorithme est applicable à plusieurs types de compresseurs avec des outils de formes et dimensions différentes. Le trajet 5-axes de l'outil obtenu permet d'enlever rapidement un volume maximal de matériaux entre deux pales consécutives du rotor. Ce travail démontre le potentiel intéressant que cette nouvelle technologie qui peut être appliquée à grande échelle en production.

CONDENSÉ EN FRANÇAIS

Les fabricants de moteurs d'avions sont constamment à la recherche de nouvelles méthodes leur permettant d'augmenter la température interne des moteurs. Cela augmente la performance, diminue la consommation d'essence et réduit la pollution générée par les moteurs. Pour satisfaire à cette demande, de nouveaux matériaux ont étés développés pour la fabrication des composantes critiques du moteur. Parmi ces matériaux on note les alliages de titane et de nickel. Ces alliages ont la particularité d'être très légers, mais sont en même temps très résistants. De plus, leur dureté est préservée même à des températures élevées. Par contre, ils sont difficiles à usiner, car ils occasionnent une usure rapide de l'outil de coupe.

En particulier, les rotors aubage monobloc (IBRs, intergrally bladed rotors) sont généralement usinés avec un taux d'enlèvement de matière de plus de 90%. Leur usinage engendre des coûts de production élevés, car la vitesse d'enlèvement de la matière est basse. De plus, la méthode généralement utilisée, qui consiste à passer par tous les points sur la surface des pales avec la pointe de la fraise, est aussi responsable pour les coûts de production élevés, dû à son inefficacité. Cela s'explique par le fait qu'un grand nombre de passes de l'outil sont nécessaires pour usiner toute la surface de la pale (qui est une surface complexe).

Pratt & Whitney Canada a investit des efforts considérables pour développer et implémenter de nouvelles méthodes d'usinage des rotors aubages monobloc, méthodes qui sont plus productives et permettent, en même temps, de réduire les coûts de production. Une méthode récemment implémentée dans la compagnie est la méthode de *flank milling.* Cette méthode consiste à générer la surface de la pale avec une seule passe de l'outil. Dans ce cas, la fraise de forme conique (très allongée) usine avec son flanc et non seulement avec l'extrémité, comme dans le cas de la méthode d'usinage par points.

Cela a permis à Pratt & Whitney Canada de réduire de quinze fois le temps d'usinage des compresseurs monobloc.

Malgré les progrès réalisés suite à l'introduction en production de la méthode de *flank milling*, les opérations, surtout de dégrossissage, restent encore peu productives et coûteuses. La fraise de forme allongée ne peut soutenir les forces de coupe très importantes présentes lors du dégrossissage, ce qui limite la vitesse d'avance de l'outil et, par le fait même, affecte le taux d'enlèvement de la matière. De plus, l'espace entre deux dents consécutives de la fraise n'est pas suffisamment grand pour permettre une bonne évacuation des copeaux. Une augmentation importante de l'avance de l'outil causerait un encombrement des copeaux dans cet espace et l'outil casserait.

Dû aux inconvénients de la méthode actuelle de dégrossissage des rotors monobloc, des recherches ont étés entamées en 1995 par M. Ioan Sasu, du département de fabrication chez Pratt & Whitney Canada, pour développer une nouvelle méthode et un nouvel outil de coupe. Cet outil, appelé *Cup Mill*, comporte un corps de forme cylindrique, en acier 4140, tournant autour de son axe, sur lequel son attachés des pastilles qui effectuent l'usinage des pales du rotor. Le diamètre de l'outil varie entre 7 et 10 pouces et peut supporter une vingtaine de pastilles.

Le principal avantage de cet outil est sa grande rigidité, ce qui permet d'utiliser une plus grande avance lors de l'usinage et, par le fait même, d'augmenter la vitesse d'enlèvement de la matière. L'usure de l'outil est aussi moins importante comparée à celle des outils actuels, car le *Cup Mill* comporte entre 16 et 20 pastilles (dépendamment du diamètre de l'outil), comparativement à trois ou quatre pour l'outil actuel.

Dû au fait que ce nouvel outil de coupe utilise des pastilles qui peuvent être remplacés facilement lorsqu'il sont usés, les coûts d'outillage en sont aussi grandement réduits. Les fraises actuelles sont spécialement conçues pour pouvoir s'insérer dans l'espace restreint entre les pales consécutives du rotor. À cause de leur forme spéciale, les outils actuels sont fabriqués chez Pratt & Whitney Canada avec des coûts de production élevés. De plus, ils sont entièrement faits en carbure, qui est un matériel dispendieux. Dans le cas du *Cup Mill,* seulement les pastilles sont faits en carbure est peuvent être facilement remplacés par de nouveaux pastilles fournis par un fabricant spécialisé.

En 2002 fut produit le premier prototype du *Cup Mill* chez Pratt & Whitney Canada. Cet outil a 10,125 pouces de diamètre et 8,875 pouces pour le diamètre intérieur (largeur de coupe de 0,625 pouces). Il comporte 20 pastilles standard en carbure, produits par Kennametal. Le potentiel de cet outil pour enlever efficacement de grandes quantités de matériel a été démontré par des tests en usine. Lors de ces tests, le *Cup Mill* plonge à répétition dans des blocs en alliage de titane (le même matériel utilisé pour les rotors aubages monobloc). Ces tests ont été concluants pour mettre en évidence l'usinage des rotors monobloc d'une manière efficace et à moindres coûts.

Les objectifs de la présente recherche peuvent être groupés en deux volets:

1. Développer, en partant du prototype réalisé en 2002 et en collaboration avec les grands fabricants d'outils, une forme améliorée du cup mill, utilisable dans les conditions concrètes de production chez PWC. Une attention particulière sera accordée à la durée de vie des pastilles et à la productivité. De plus, le coût des pastilles, ainsi que le corps de l'outil seront aussi pris en considération.
2. Développer un algorithme capable de générer le trajet de l'outil, algorithme qui permettra d'usiner avec une relative précision les pales de rotors aubages monobloc. La précédente version de l'algorithme produite par M. Petean en 2004 [10] sera analysée, ainsi que les méthodes de positionnement de l'outil développées pour l'usinage par points et pour le *flank milling.*

Développement du Cup Mill

Le développement de l'outil est fait en collaboration avec les trois grands fournisseurs d'outillage, soit Sandvik, Kennametal et Iscar. Le *Cup Mill* proposé par chacun des fournisseurs ne doit pas être plus gros que 9 pouces en diamètre et son poids ne doit pas dépasser 50 livres, ce qui correspond aux limites imposées par les machine-outils présentement utilisées chez Pratt & Whitney Canada. Les pastilles utilisés doivent être standards (coûts réduits) et posséder une bonne durée de vie ainsi qu'une productivité accrue.

Les trois fournisseurs nous ont présenté leurs designs pour le *Cup Mill.* Kennametal propose un outil qui comporte 20 pastilles standard en carbure. Les pastilles sont positionnées sur des cassettes, mais leur position radiale ne peut être contrôlée, car les cassettes sont fixées sur le corps de l'outil avec des visses. Le bon alignement des pastilles dépend donc seulement de la précision de fabrication du corps du *Cup Mill* (augmentation des coûts d'achat de l'outil). Les tests d'usinage dans des blocs d'alliage de titane effectués à l'usine de Kennametal ont démontré une usure des pastilles extrêmement rapide: les pastilles sont usés seulement après avoir plongé une seule fois dans le matériel. Au moment de l'écriture de cette thèse, le développement de ce prototype est interrompu.

Iscar propose un design de Cup Mill qui possède un diamètre extérieur de 8,50 pouces et un diamètre intérieur de 7,40 pouces. Il comporte 16 pastilles qui sont spécialement conçus pour cet outil, donc leur coût est plus élevé et se situe à 20 $ par pastille. Les pastilles sont fixés directement au corps de l'outil à l'aide d'une vis. Ce design (comme celui de Kennametal) ne permet pas de contrôler leur portion radiale et leur bon alignement dépend de la précision de fabrication du corps de l'outil. Le coût de celui-ci est relativement raisonnable (approx. 2000 $). Au moment d'écrire cette thèse, aucun test d'usinage n'a été effectué en alliage de titane, donc la productivité et la durée de vie des pastilles sont inconnues.

Le Cup Mill proposé par Sandvik est le plus intéressant. Il comporte 16 pastilles qui sont fixés sur le corps de l'outil en utilisant des cassettes, donc leur position radiale peut être ajustée. Les pastilles sont standard et coûtent 11 $ chacun. Le corps de l'outil est plus dispendieux que celui de Iscar et coûte 5000 $. Sandvik a effectué des tests d'usinage dans des blocs d'alliage de titane qui ont démontré la grande productivité et la bonne durée de vie de leur design. Avec un set de pastilles, Sandvik a réussi à plonger dans les pièces de tests 131 fois, à une profondeur de 45 mm. Chaque fente a été générée en 50 secondes.

En utilisant les résultats de ces tests, nous pouvons évaluer les performances du processus actuel de dégrossissage des pales de rotors monobloc en comparaison avec le nouveau processus en utilisant un outil *Cup Mill.* Pour ce faire, le rotor deuxième stage du moteur de la série 500 (numéro 30J2429) est utilisé comme pièce de comparaison. En premier, les coûts de production ainsi que la productivité du processus actuel sont évalués pour le rotor choisi. Ensuite, en utilisant les résultats des tests effectués par Sandvik, nous pouvons, pour la même pièce, évaluer les coûts de production et la productivité du nouveau processus de dégrossissage. Il en ressort une réduction potentielle de 70% des coûts et du temps d'usinage. Par conséquent, en remplaçant les fraises utilisées actuellement par l'outil Cup Mill, une économie de 300 $ par rotor produit est possible.

Développement de l'algorithme

Dû à la spécificité de cet outil, présentement il n'existe aucun programme capable de générer le trajet de l'outil pour usiner les pales de rotors. La deuxième partie de cette recherche consiste à développer un algorithme permettant d'enlever le plus de matériel possible entre deux pales consécutives du rotor aubage monobloc.

Une première version de l'algorithme a été développée en 2004 par M. Marius Petean, au cours de sa recherche de maîtrise. L'objectif premier de cet algorithme était de démontrer qu'il est possible d'usiner avec le *Cup Mill* des surfaces aussi complexes que celles qui composent les pales des rotors. Par contre, cet algorithme n'est pas suffisamment performant pour être utilisé en production:

- la précision d'usinage est insuffisante ;
- faible productivité, à cause du long trajet de l'outil, généré par cet algorithme.

Cette première version de l'algorithme génère un trajet de l'outil qui est à 3-axes, donc l'axe de l'outil reste fixe durant l'usinage. Ceci limite sérieusement la précision des positions de l'outil sur la surface des pales et laisse beaucoup d'excès de matériel. L'espace entre deux pales consécutives est usiné en générant les positions de l'outil à différents niveaux (hauteurs). La suite des positions à différents niveaux constitue le trajet de l'outil. La grande longueur du trajet de l'outil vient du fait que la surface d'aspiration et la surface de pression sont usinées à tour de rôle, ce qui crée un mouvement en zigzag de l'outil.

Les objectifs du nouvel algorithme sont:

- Générique (applicable à tous les différents numéros de pièces)
- Trajet de l'outil sans interférences
- Optimal (maximum volume de matériel enlevé)
- Productif

Cet algorithme est conçu en utilisant l'interface CATIA V5 et le langage de programmation Visual Basic 6.0, qui permet d'accéder aux fonctions API (Application Programming Interface) de CATIA. Cette combinaison CATIA V5 - Visual Basic simplifie la création de l'algorithme, car un bon nombre de fonctions nécessaires sont déjà intégrées dans CATIA et ne doivent pas être crés.

À chaque niveau, l'algorithme doit positionner l'outil le plus près possible de la surface de la pale (surface de pression ou d'aspiration). La combinaison des différentes positions de l'outil permet l'enlèvement d'un maximum de matière entre les deux pales. Pour ce faire, 5 lignes équidistantes recouvrent la surface de la pale dans la direction de la longueur de celle-ci (direction radiale du rotor). L'algorithme place l'outil tangent à la surface de la pale à un niveau donné et tente de minimiser les distances entre chacune des lignes et la surface coupante (surface engendrée par la rotation des pastilles) de l'outil. Ce problème de minimisation est résolu par la méthode de Newton. La fonction qui doit être optimisée est la somme des distances entre les lignes à la surface de la pale et la surface coupante de l'outil. Ces distances dépendent des angles de rotation (les 3 axes possibles) autour du point de tangence, qui sont les variables indépendantes de la fonction. Le calcul s'arrête quand les distances entre le *Cup Mill* et les cinq lignes, pour ce niveau d'usinage, sont au minimum. L'algorithme vérifie, à chaque itération, si l'outil entre en collision avec les autres surfaces du rotor (la surface de la pale opposée et la surface du disc du rotor). En cas de collision, pour cette direction de rotation, l'angle de rotation est réduit de moitié et une autre direction de rotation est recherchée (jusqu'à l'obtention de la solution). L'outil est translaté vers le haut ou vers le bas (dépendamment de la direction de calcul des positions) pour le niveau d'usinage suivant et le processus d'optimisation recommence pour cette nouvelle position. Le trajet de l'outil ainsi obtenu demande de l'usinage 5-axes et permet d'usiner la surface de la pale le plus précisément possible avec le Cup Mill.

Contrairement à la version précédente, cet algorithme usine chaque coté de la pale séparément (passes d'usinage successives). Les positions de l'outil constituant le trajet permettant l'usinage de la face de pression sont générées en premier lieu et ensuite les positions pour la face de aspiration. La combinaison de ces deux trajets de l'outil permet d'usiner l'espace entre deux pales successives du rotor.

L'algorithme comporte les sous-programmes suivants:

- *Gradient evaluation*: détermine la rotation (autour du point de tangence) qui permet la plus forte minimisation de la fonction.
- *Rotation axes*: crée les axes de rotation à partir du point de tangence, autour duquel l'outil tourne.
- *Tangency finding*: détermine le point de tangence et repositionne l'outil tangent à la surface de la pale après chaque rotation.
- *Collision prevention*: assure que les positions de l'outil obtenues ne sont pas en collision avec aucune des surfaces du rotor.

Afin de valider le présent algorithme, des tests d'usinage ont été effectués sur une machine 5 axes Taurus chez Pratt & Whitney Canada. La pièce brute utilisée est en aluminium (non en titane, pour des raisons d'économie) et sa forme est identique à celle du rotor aubage monobloc qui a pour numéro de pièce 30J2429. La vitesse d'avance utilisée pour ce test est de 3 pouces par minute, avec une rotation de l'outil de 70 tours par minute. Les résultats obtenus démontrent la grande précision du trajet de l'outil obtenu et la grande productivité de ce processus d'usinage.

Les directions futures de recherches proposées sur ce sujet peuvent être énumérées ainsi :

- Temps de calcul de l'algorithme très long.

  Chaque changement effectué par l'algorithme est affiché à l'écran. Dû au grand nombre de changements effectués, le temps de calcul est très long (2-3 jours de calcul). La solution consiste à développer un algorithme qui n'utilise plus cette interface entre CATIA et Visual Basic, mais plutôt un seul langage de programmation (C++, Fortran MatLab,...). Cet algorithme est plus compliqué à développer, mais le temps de calcul sera réduit.
- Collisions mineures avec la pièce brute observées lors des tests..

Des frottements entre le corps de l'outil et la pièce de test sont dûs à des lacunes dans le sous programme *Collision Prevention.* Présentement, des vérifications sont faites par ce sous programme entre la pièce (finie) et l'outil, mais pas entre le corps de l'outil et la pièce brute (nécessaire pour prévenir les frottements observés). Cette modification devra être faite pour la version suivante de l'algorithme.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

A

ASFM – Arbitrary surface flank milling

C

CAM – Computer aided manufacturing

CAD – Computer aided design

CATGEO – CATIA geometric interface Fortran routines

CC – Cutter contact

CL – Cutter location

CN – Control numérique

E

ECM – Electrochemical machining

F

FLAMINGO – Flank milling optimization

I

IBR – Integrally bladed rotor

IPM – Feed rate [in/min]

IPT – Feed rate per tooth [in/tooth]

IUA – Interactive user access

M

MED – Manufacturing engineering development

N

NC – Numerical control

P

P&W – Pratt & Whitney

P&WC – Pratt & Whitney Canada

PS – Pressure side surface

R

R&D – Research & development

S

SFC – Specific fuel consumption

SS – Suction side surface

V

VB6 – Visual Basic 6.0

LIST OF APPENDIXES

CHAPTER 1: Introduction

The past decade has witnessed a radical approach toward product manufacture, especially in the developed countries, in order to remain competitive. Efforts were particularly focused toward reducing non value added activities and achieving step increase in product manufacture. Aircraft engine manufactures are competing to develop new manufacturing process technologies that would help them to produce their parts faster and at lower costs. Particularly, axial compressors and impellers are composed of complex sculptured surfaces, and the machining of these components is characterized by low material removal rates.

1.1 Machining of sculptured surfaces

Sculptured surfaces are used in a wide variety of applications: in the aerospace, automotive, shipbuilding, dies and moulds industries. These surfaces are usually mathematically defined by parametric forms such as Bezier, B-splines or NURBS. As seen in Figure 1.1, a sculptured surface is formed by a series of isoparametric curves in "u" and in "v" directions.

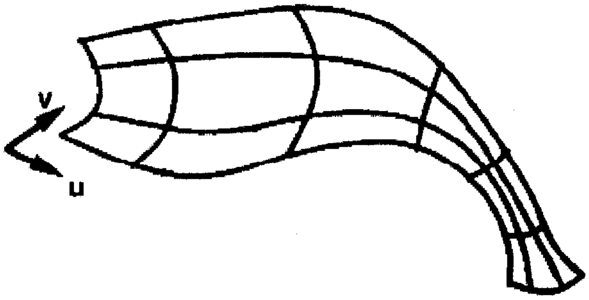


Fig. 1.1 Sculptured Surfaces.

Machining of a sculptured surface can be very expensive and time consuming. A great extent of energy is employed by researcher to develop strategies to machine such surfaces that are gauge-free, faster and produce a good surface finish. Machining processes of sculptured surfaces can be grouped into two distinct milling methods. The first method, point-milling, employs cutting edges near the end of the tool to remove the material. In flank milling, on the other hand, the side of the tool does most of the cutting.

1.2 Manufacturing of integrally bladed rotors (IBRs)

Compressors, fans and impellers are generally machined with material removal up to 90% of the bulk material. In the scope of reducing the specific fuel consumption of the engines, as well as the reduction of the gas emissions, engines components are more and more designed using temperature resistant alloys. These high temperature alloys are typically classified as difficult-to-machine material. Consequently, machining these components is generally characterized by low material removal rates. An increase in productivity, as well as the surface quality of the parts, was achieved by applying high speed machining technology with a point milling process. More recently, at Pratt & Whitney Canada, further productivity improvements were obtained by the introduction of the flank milling technique for machining airfoil components. Compared to the point milling technique, the introduction of the flank milling method in production represents a 14 times reduction in the numbers of cuts (correlating directly with machining time), 75% cost reduction and an improved surface quality.

1.3 Motivation

In order to increase the productivity of manufacturing compressor blades, most research was oriented so far to develop more efficient methods for finishing operations, since the point milling method used is time consuming. The introduction of the flank milling method greatly accelerated the finishing process, and consequently, the research has been reoriented toward the roughing operations.

The current endmills used at Pratt and Whitney Canada for roughing operations have poor material removal capacities. Their slender and elongated shape, specially designed to fit in the confined space of successive compressor blades, makes these tools more suited for finishing operations. Used for roughing, however, these fragile tools cannot withstand the large cutting forces implied. In addition, the chip evacuation potential is very limited. The cutting speeds are reduced to accommodate the limitations of these tools. As a result, the roughing operations are among the most time consuming ones.

The main goal of the present thesis is to develop a new more productive IBR roughing tool, in collaboration with majors tooling suppliers, and to elaborate the algorithm that generates the tool path to machine IBRs blades using this tool.

1.4 Objectives and approach

In 1999, Mr. Ioan Sasu, Ph. D., and his associates, proposed a new tool concept used to rough compressor blades, called cup mill cutter, to replace the current endmills tools. A picture of a cup mill is presented in Figure 1.2.

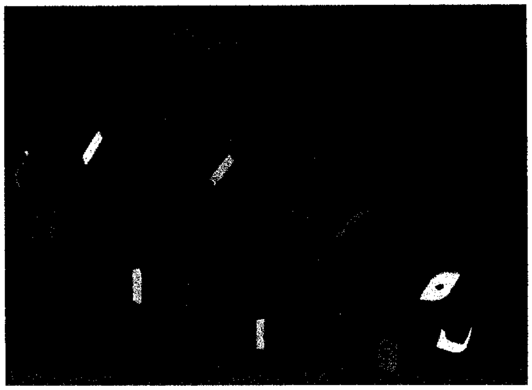


Fig. 1.2 Cup Mill cutter.

The objectives of this thesis can be summarized as follows:

- Develop new cup mill tool
- Evaluate the potentials of the new tool in terms of productivity, tool life and costs;
- Develop an algorithm capable to generate the optimal tool path of the cup mill (removal of a maximum volume of material).

The tool development is performed in collaboration with major tool manufacturers to optimize the performances of the cup mill. The proposed tool designs should accommodate standard inserts and cassettes and respect the size and weight limitations of the machine tools at Pratt & Whitney Canada. In parallel, the tools developed must be very productive and possess a reasonable tool life.

A first version of the algorithm was developed in 2004 by M. Petean [10] in his master project. We will analyze the tool positioning strategy, along with the tool kinematics of this algorithm and underline its limitations. By referring also to others tool positioning strategies developed for point and flank milling, we will develop an improved tool path generation algorithm in which a maximum volume of part material is efficiently removed.

1.5 Thesis structure

Chapter 2: Actual IBR roughing method

The machining difficulties encountered during the manufacturing process of IBRs are discussed in this chapter. It is also presented the current roughing method employed at PWC. Finally, we estimate the productivity, along with the related manufacturing cost, of the current process, for a selected part number.

Chapter 3: Bibliographic review

The methods used to machine compressor blades are discussed: point milling and flank milling. For both methods, various tool positioning strategies are presented. In particular, the "Triple tangent flank milling method" is discussed, which could be used as a starting point in the development of a tool positioning strategy used for the cup mill cutter. The cup mill cutter developed at PWC and the tool path generating program developed by M. Petean is also discussed.

Chapter 4: Tool development for IBRs machining with cup mill

This chapter presents the development of the "production" cup mill cutter in collaborations with major tool suppliers. The performances of the prototypes provided are evaluated and discussed. The potential improving of the productivity of the roughing process, as well as the manufacturing cost reduction is estimated for this new process for the same part number as in Chapter 2. The results are compared to the performance of the actual process.

Chapter 5: Program development for IBR machining with cup mill

A general overview of the new tool positioning strategy, as well as a new tool kinematics, is discussed here. This new algorithm tries to address the limitations of the first algorithm created by M. Petean, discussed in Chapter 2.

In addition, this chapter gives a detailed description of the algorithm. First, the new tool positioning strategy is described and the important subroutines of the algorithm are explained. Then, is presented the collision prevention strategy and, finally, the method used to smooth out the tool path.

Chapter 6: Tests & Results for IBR machining with cup mill

Finally, in this chapter are presented the results of the tests performed using the algorithm described in Chapter 5. For two selected part numbers, the algorithm was virtually tested in CATIA V5 and then, real machining tests in production were performed.

Chapter 7: Conclusions & Future Work

The potential costs reduction and productivity increase associated with the implementation of this technology in the production process is discussed. We also present some observation regarding the cup mill cutter and the algorithm developed. Some limitations of the algorithm are presented as well as the proposed direction for future work on this project.

CHAPTER 2: Actual IBR roughing method

2.1 Definition of Integrally Bladed Rotors (IBRs)

2.1.1 IBRs - part of a gas turbine engine

Modern aircrafts are powered by engines using the *open* Gas Turbine Systems. A gas turbine is a rotary engine that converts internal energy of combustion gases into mechanical energy (see Figure 2.1).

This system comprises three important parts: the compressor, the combustion section and the turbine. Air is admitted into the compressor where it is compressed, it then goes into the combustion chamber where the internal energy is increased and finally this energy is transformed into work by the turbine section. Part of the work generated is used to drive the compressor (through a shaft) and the rest is used for other needs of the aircraft (work out).

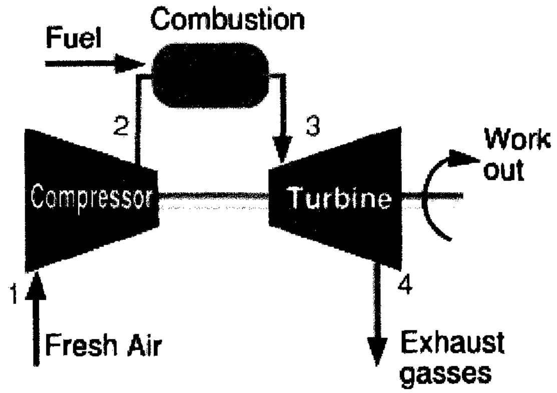


Fig. 2.1 Open Gas Turbine System

A simplified representation of an aircraft engine is presented in Figure 2.2. The compressor group is composed of a series of axial compressor (multi-stage compressor). At each stage, the air is further compressed until it reaches its maximum pressure prior to enter the combustion chamber. Each of these stage compressors are composed of airfoils in the form of rotor blades, extending radially outward from the periphery of the disk.

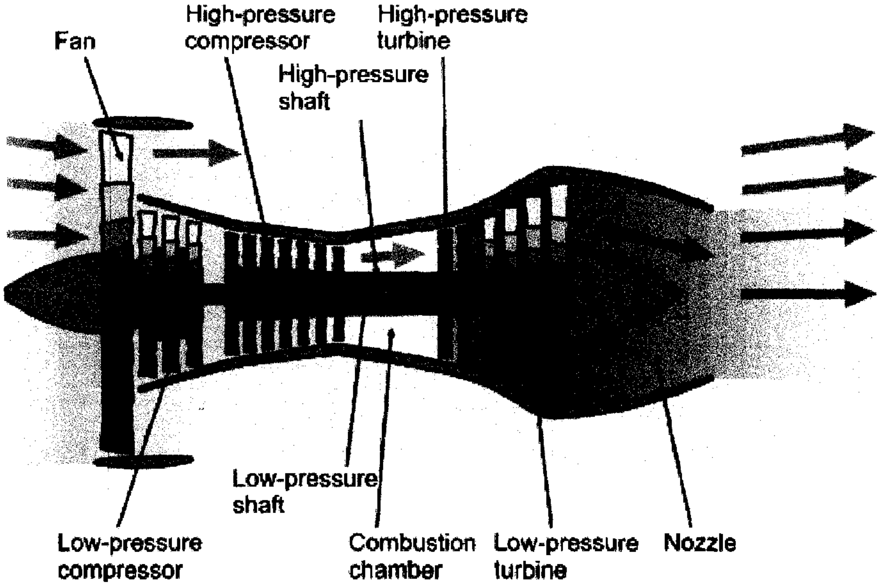


Fig 2.2 Twin Spool Turbofan Engine

The blades of conventional aircraft axial compressors are mechanically attached to the disks by the use of a precision-machined dovetail slots arrangement. Blisks (bladed disks), on the other hand, are produced by machining from a large one-piece forging. Compressors made by this method are also called integrally bladed rotors or IBRs. Figure 2.3 presents an example of an IBR.

Fig. 2.3 Integrally Bladed Rotor

Making a one-piece blisks with blades integrated in the disk offers substantial performance advantages by significantly reducing the rotating mass of a conventional turbine rotor assembly. Performance advantages also include:

- increased thrust-to-weight ratio;
- reduced specific fuel consumption;
- longer service life;
- manufacturing costs reduction of the part, compared to the dovetail-slot arrangement.

2.1.2 Aero-engine materials

The driving force for continual development of many materials over the years is the need for harder, stronger, tougher and corrosion/oxidation resistant materials. More specifically, for the case of aero-engine alloys, a high strength-to-weight ratio is required. Demand for hotter, more powerful and more efficient engines led to the development of superalloys (see Figure 2.4), which can withstand ever-increasing internal engines temperatures.

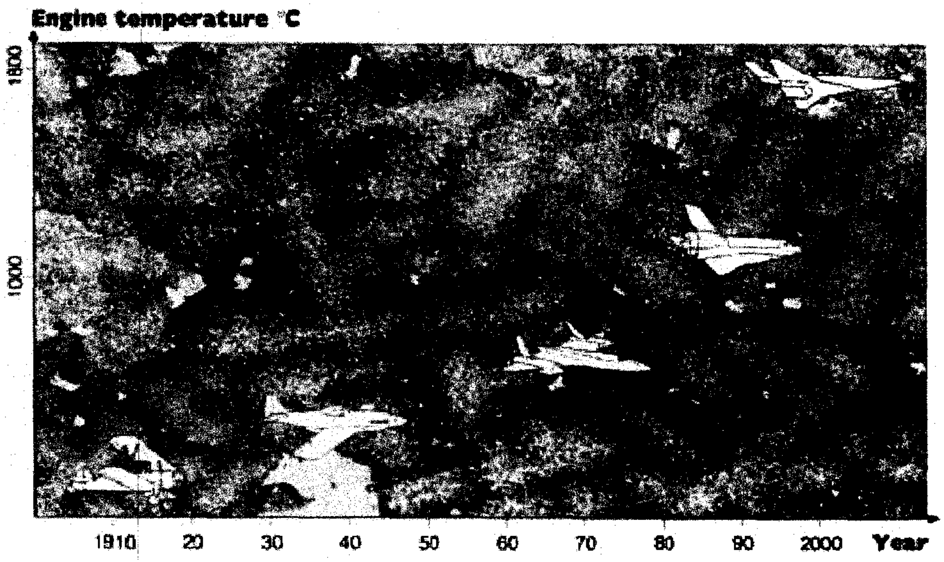


Fig. 2.4 Evolution of the internal engine temperature of aircrafts [4]

The machinability of aero-engines alloys is declining in order to satisfy the demand for higher temperature capability for structural engine alloys. From 1990s till date we have witnessed almost a two-fold increase in engine temperature. IBRs, in particular, are made of advanced titanium alloys (titanium aluminides). Titanium alloys exhibits an outstanding strength to density ratio. They exhibit exceptional resistance to corrosion, which provides savings on protective coatings like paints, otherwise necessary in the case of steel.

Titanium aluminides are one of the most exciting new alloys introduced for aerospace applications, because of their outstanding resistance to high temperatures. The most popular form of titanium alloy used for producing IBRs is the Ti-6Al-4V configuration [4]. In contrast, like most new materials, titanium aluminides are difficult to machine. The high temperature strength, toughness and ductility of these super alloys seriously impair chip segmentation. The presence of abrasive carbide in their microstructure accelerates tool wear at the cutting edge.

It is recommended to use a positive rake angle (α +ve) for semi-finishing and finishing operations [4]. This minimizes work hardening of the machined surface by shearing the chip away from the work piece in an efficient way and, also, minimizes the formation of a buildup edge. Cutting tools with sharp edges are useful in preventing material build-up and improves surface finish. Deflections of the work material should also be prevented, by using special fixturing.

The engine internal temperature could be further increased by manufacturing the IBRs in inconel 718, since it possess a higher melting temperature than titanium. It is, however, far more difficult to cut than titanium. Producing inconel IBRs using the current ball endmills is difficult and very expensive due to the extremely short tool life. A new machining method and cutting tools are required prior to starting production of inconel IBRs.

2.2 Current IBR roughing tools

2.2.1 Conical vs. cylindrical tools

Flank milling of ruled surfaces is commonly performed by using cylindrical cutting tools. The characteristics of a cylindrical tool make it suitable to solve most of the problems encountered in the industry. However, when the surface to be machined is within a confined space, a conical tool could prove to be a better choice, due to its smaller tip and a stronger shank in comparison to cylindrical tool. In addition, the deviations between the flank milled surface and the design surface, as well as the interferences between the cutting tool and the work piece must also be considered when selecting the type of the tool. A tool having a large diameter leads to large deviations and a higher probability of collisions. This suggests that conical tools, like taper ball endmills, could be a better choice in compressor airfoil machining.

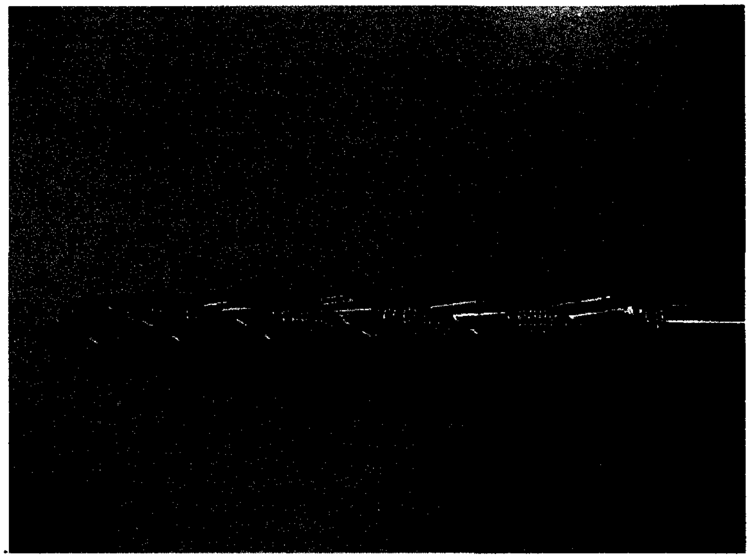


Fig. 2.5 Hog Taper Ball Endmill (Rougher)

A photography of a hog taper ball endmills, used at PWC to machine IBRs, is presented in Figure 2.5. These tools are ideal for finishing the airfoils of an IBR due to their thin and elongated geometry. In this case, the amount of material removed is small, the tool robustness and a good chip evacuation potential are not the most important tool characteristics. On the other hand, used for roughing operations, these tools are unproductive and expensive.

2.2.2 Productivity aspects

As mentioned above, it is very important to efficiently rough integrally bladed rotors, since most of the material of the initial blank part has to be removed during this operation. Currently, at PWC, hog taper ball endmills are used to rough IBRs, but the process is slow because the material removal potential of these tools is limited.

The design of the taper ball endmills is basically a compromise between access and rigidity, compromise which limits the productivity of the cutting tools. These limitations become evident during the roughing operations, when the cutting forces applied on the tool are at a maximum. During the open pass operation (opening the space between two blades) the cutter is fully engaged into the material (see Fig. 2.6). Due to the large machining forces present, the cutting feed used for this operation is low. The open pass operation is usually the most time consuming one, especially for IBRs with longer blades where a longer portion of the cutter is engaged in the material.

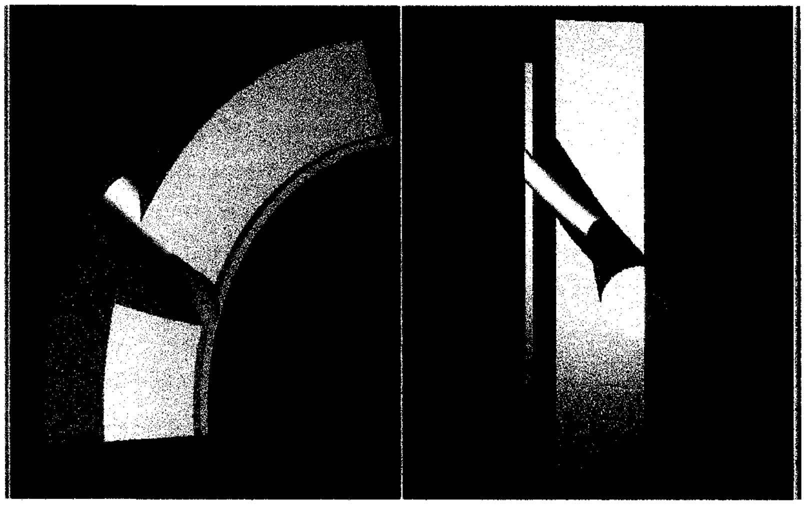


Fig. 2.6 Representation in VERICUT of the open pass operation

The long and slender shape of conical cutters tools limits the chip evacuation capacity. To increase the rigidity, the ball endmills have shallower flutes and thicker webs than conventional tools. In counter part, this impedes the chip evacuation, which is particular important during roughing. A significant increase in the cutting feeds used for the roughing process is impossible, because the tools cannot evacuate the quantity of chips generated. The chips accumulate in the space between the teeth and the tool breaks. This tool brakeage is more likely to occur at the junction between the shank and the cutting position, where the stress is the highest. An example of a broken tool (rougher) is illustrated in Figure 2.7.

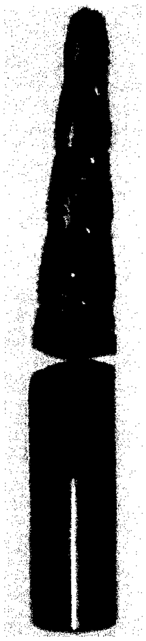


Fig. 2.7 Broken Ball endmills during roughing operation

2.2.3 Cost aspects

Besides being unproductive, these tools are also very expensive. Due to their special geometry, particularly adapted to the intended operation they are designed for, these tools are not standard and are not included in supplier's catalogs. Consequently, there are two options for their supply: in-house grinding or to buy them from a specialized grinding shop. Both solutions are very expensive.

Presently, PWC is producing in-house, in their cutter grind department, all the ball endmills used for IBRs flank milling. This involves a big investment in machinery and personnel. Figures 2.8 and 2.9 present Gemini cutter grinding machine used at Pratt & Whitney Canada.

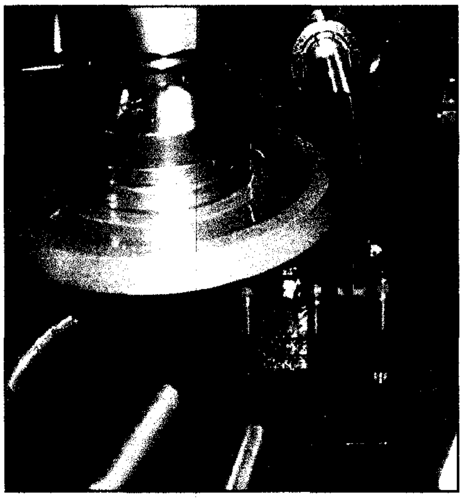


Fig. 2.8 Manufacturing of a ball endmills

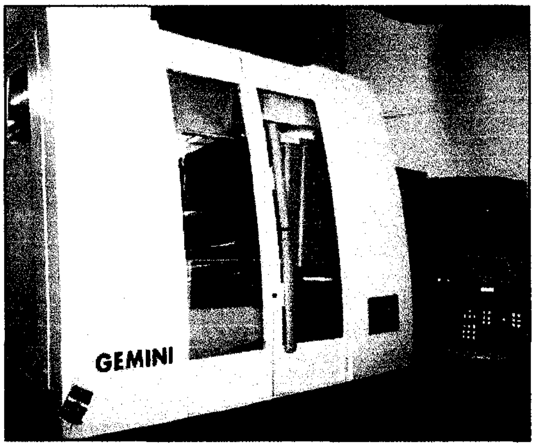


Fig. 2.9 Cutter grind machine tool

There are also some other aspects which contribute to the high production costs of taper ball endmills:

- The raw material they are made from – carbide – is expensive, relatively rare, and the shipping delay is long;

- The carbide is irrationally used, both part of the tool (the shank and cutting portion) are made from this material; after re-sharpening, the leftover carbide (more than 60%) is discarded.

2.3 Productivity & cost of actual method

To evaluate the productivity, and associated manufacturing cost reduction of the new cup mill method, we need to first determine the performances of the actual process. In that scope, the productivity and manufacturing cost of the current process are determined with respect to a selected IBR part number. The part number is 30J2429 of the PW500 engine series and a picture is presented in Figure 2.3. This same part will be used during the machining test with the cup mill cutter.

Table 2.1 Actual roughing process (P/N 30J2429)

Process	Tool	[redacted]	[redacted]	[redacted]	[redacted]	Total tool cost* [$]	Machining time/blade [min/blade]
Open Pass	PC319082	[redacted]	[redacted]	[redacted]	[redacted]	1100	2.42
R Hub							1.25
R Suct.	PC319279	[redacted]	[redacted]	[redacted]	[redacted]	1080	1.25
R Press.	PC311293	[redacted]	[redacted]	[redacted]	[redacted]	1000	1.75
						Total	6.67

* [Total tool cost] = [cost of blank] + [cost initial grind] + [cost/re-sharp] X [number of re-sharp]

The current roughing process of this part number comprises 4 roughing operations (in order):

1 **Open pass:** open the space between two consecutive blades;

2 **Roughing pressure side;**

3 **Roughing suction side;**

4 **Roughening of the hub:** removes excess material left of the hub by the previous roughing operations.

Three different taper ball endmill used for the roughing process for this part number and their "PC" number are identified in Table 2.1. For each of these tools, it is also presented the tool life between re-sharp and the number of possible re-sharps.

The total machining time of the cutters is calculated in the following manner:

[Tool life] x {[Number of re-sharps] + 1}

The total cost of each tool comprises: the initial cost of the blank carbide, the cost of the initial grinding operation and the cost of the successive re-grinding operations. Using the total tool life and machining time per blade (19 blades) of each roughing operations, we can calculate the numbers of parts that can be produced per tool.

In Table 2.2 is presented the tool cost per part, as well as the machining costs per part for the selected IBR. The hourly machining cost at PWC is estimated at 200 $/hrs.

Table 2.2 Roughing part cost of actual process (P/N 30J2429)

Process	[redacted]	Machining		Total Cost
		Time	Cost	
	[redacted]	[hrs/part]	[$/part]	[$/part]
Open Pass	[redacted]	2.11	422.43	464.73
R Hub				
R Suction				
R Pressure				

Analyzing the distributions of the actual manufacturing costs of this IBR, presented in Table 2.2, it stands out that more than 90% of the total cost is related to the machining costs, not to the cost of tooling (only 10%). In conclusion, a reduction of the machining time (productivity increase) has a greater impact on the total manufacturing cost of IBRs that the reduction of tool costs.

It is apparent from Table 2.2 that the roughing time per part is very long. This situation is responsible for the high manufacturing cost of the roughing process and it is evident that by using the current ball endmills the productivity of the roughing process cannot be greatly improved, thus the need for a new roughing method. The process of searching for a new method/tool for roughing IBRs has already started with the development of the cup mill cutter, discussed in the next chapter.

CHAPTER 3: Bibliographic review

3.1 Point Milling

Point milling was the first method of machining sculptured surfaces and it is still very broadly used even today. The strategy consists on using the ball end of the tool to generate the surface as it makes many closely spaced tool passes over the design surface. These passes follow the isoparametric curves of the surface (ex. in the "u" direction, see Fig. 1.1). Whenever a curved surface is machined using a ball nose or a radiused-corner endmill, cusps are left on the surface, as shown in Figure 3.1. These cusps are related to the cross-feed, the radius of the tool, and the curvature of the surface.

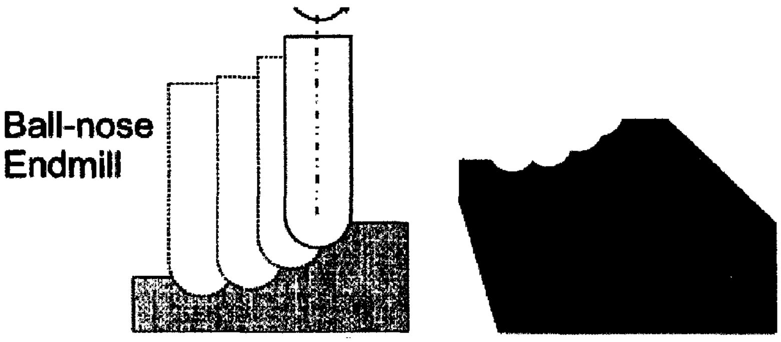


Fig. 3.1 Cusps formation [5]

Great efforts are invested on firstly, reduce the height of the scallops, secondly, reduce the length of the required path of the tool (machining time) and finally, avoid tool gouging into the surface. Faster machining time and a better surface finish can be achieved by using a 5-axis machine rather than a 3-axis one. The extra two axes (axis of rotation) permits a better matching between the design surface and the end of tool, see Figure 3.2.

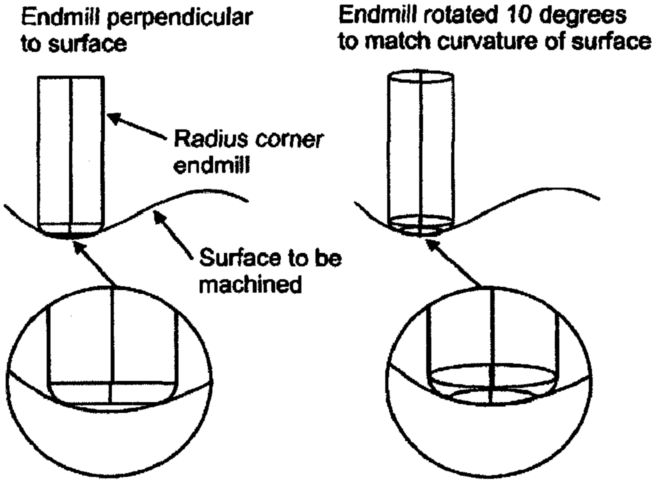

Endmill perpendicular to surface
Endmill rotated 10 degrees to match curvature of surface
Radius corner endmill
Surface to be machined

Fig. 3.2 Curvature matching with 5-axis [5]

If a ball nose endmill is more appropriate when using a 3-axis machine, S. Bedi et al. state in their paper entitled *Comparison of 5-Axis and 3-Axis Finishing Machining of Hydroforming Die Inserts* [5], that flat and radiused-corner endmills reduce machining time and improve surface finish in 5-axis machining. By tilting the tool, the curvature of the design surface at the point being machined can be matched more closely, thus reducing the number of passes required to machine the surface and reduce the cusp height.

3.1.1 Tool positioning strategies for point milling

3.1.1.1 Point milling 3-axis

Three-axis point milling using a ball nosed cutter was the first method developed for machining sculptured surfaces and is still very used today (Warkentin et al.,[13]). The strategy for positioning the cutter on the surface can be explained using Figure 3.3.

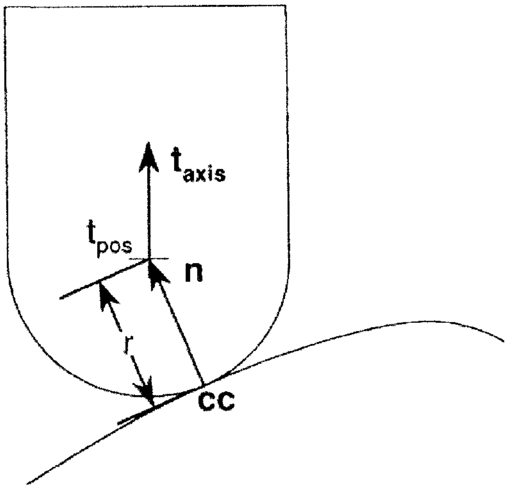


Fig. 3.3 Positioning the tool using the 3-axis tool strategy [13]

The ball nosed tool is in tangential contact with the surface at the contact point, *CC*. The tool position is offset from the contact point by the tool radius *r*, along the normal direction ***n***. The tool axis direction is constant (3-axis machining) and can be considered to be in the vertical direction, namely, $\boldsymbol{t_{axis}}$=$[0,0,1]^T$. Thus, the position of the tool can be expressed by the following equation:

$$\boldsymbol{t_{pos}} = CC + r\boldsymbol{n} \qquad (3.1)$$

3.1.1.2 Point milling 5-axis

A common 5-axis tool positioning strategy is the Inclined Tool method (also known as *Sturz milling*), proposed by Vickers et al. in 1985 [13]. This is usually a superior strategy than the 3-axis one because the orientation of the tool follows the shape of the machined surface. The tool axis, $\boldsymbol{t_{axis}}$, is inclined in the feed direction, $\boldsymbol{f}$, by a constant angle ϕ. Figure 3.4 shows the Inclined Tool method for a toroidal cutter. To easily define the position and orientation of the tool in space, a fixed-on-tool coordinate system is created. One of the axes of the coordinate system is in the direction on the normal, **n**, at the contact point *CC*. The second axis, **e**, is perpendicular to **n** and is on the plane generated by the feed and normal (*f* and **n**) vectors, as seen in Figure 3.5. Namely:

$\mathbf{e} = \mathbf{n} \times (\mathbf{n} \times \mathbf{f})$ using the triple vector product

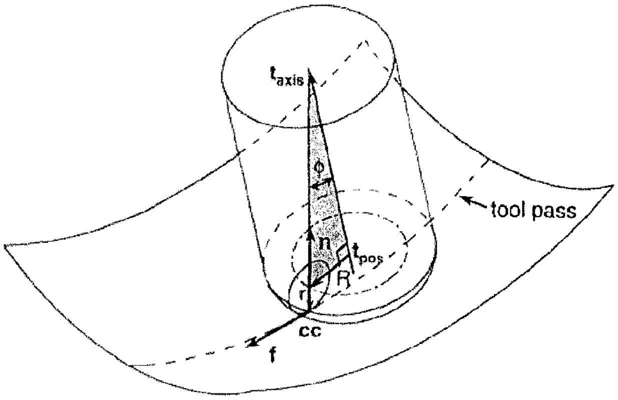
The tool axis is given by:

$$\boldsymbol{t_{axis}} = \cos(\phi)\mathbf{n} + \sin(\phi)\mathbf{e} \qquad (3.2)$$

and the tool position is given by:

$$\boldsymbol{t_{pos}} = \mathbf{c} + \mathrm{R}\sin(\phi)\mathbf{n} - \mathrm{R}\cos(\phi)\mathbf{e} \qquad (3.3)$$

where $\boldsymbol{c = CC + rn}$



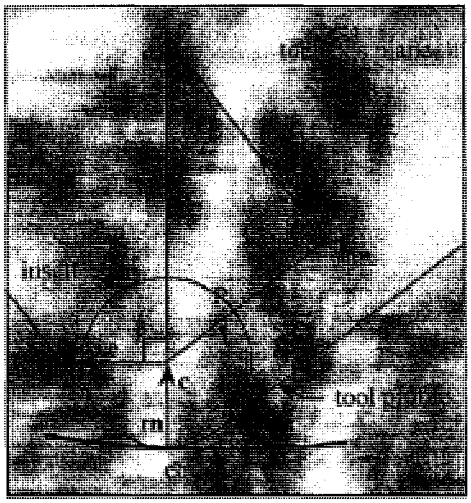


Fig. 3.4 Inclined positioning method [13] **Fig. 3.5** Tool axis plane [13]

Further improvement of this method was proposed by Rao et al. [13] who proposed the Principal Axis Method. That is one of the first methods that actually takes into account the *surface curvature* of the design surface and is presented in Figure 3.6. The tool inclination angle ϕ is no more constant and in the direction of the feed, but rather the value of ϕ depends of the curvature of the surface and is always in the direction of minimum surface curvature λ_2.

The tool position and orientation (*t_{axis}* and *t_{pos}*) are the same as the ones for the Inclined Tool method (eq. 3.2 & 3.3). The only difference will be the second coordinate frame direction **e** will now be in the minimum curvature direction.

$\mathbf{e} = \lambda_2$

The value of tilting angle ϕ will now be a function of the maximum curvature λ_1 at the point of contact *CC*.

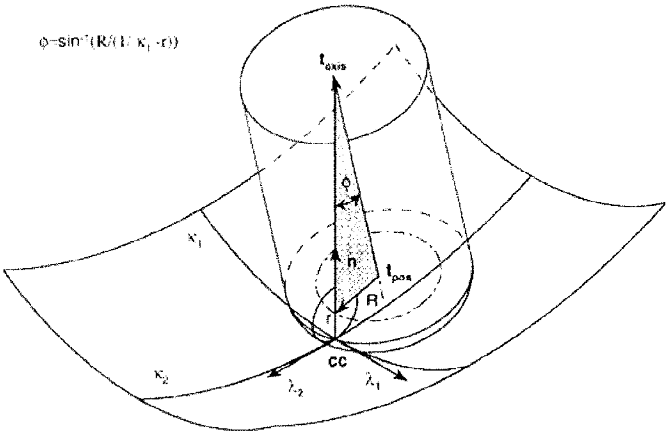


Fig. 3.6 Principal axis method [13]

A final method for 5-axis point milling of sculptured surfaces is the Multi-point machining. This method is a further improvement of the Principal Axis Method. MP machining tries to orient the tool in such a manner that there will have more that one contact point at the same time on the surface. As illustrated in Figure 3.7, besides having the first contact point CC_1, this method tries to find a second contact point on the surface, CC_2. This second point is located at a distance equal to *w* away from CC_1 in the direction of maximum curvature λ_1.

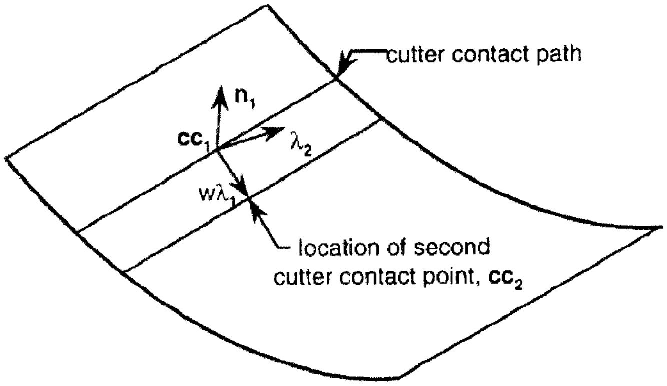


Fig. 3.7 Path of cutter contact points [13]

The major advantage of the point milling method is that it permits to machine virtually any 3 dimensional shapes, without any design modifications. From the manufacturer point of view, however, point milling is a very time consuming process since only a small amount of material is removed after each passage of the tool. Another disadvantage is that point milling produces a scalloped surface finish, which usually requires manual polishing operations. Using this method, compressor blades of a turbofan engine, for example, will require several hours to be machined (if not days) and another several hours for hand polishing.

The evolution of point milling from the simple 3-axis machining to 5-axis Multi-Point machining greatly increased the productivity and the quality of the surface finish produced. However, point milling remains a time consuming method for producing complex part such as compressors and turbines blades. It is clear that another more productive method is required.

3.2 Flank Milling

To solve this problem, a more productive approach of machining complex surfaces was developed: flank milling. In contrast to point milling, in flank milling the taper part of the cutter does the main cutting. Material removal rate is much higher since the whole length of the cutter is involved in the cutting. The surface finish is also greatly improved since no scallops are left behind.

This method is particularly well suited for machining impellers and turbine blades. Since the performance of these parts is directly related on the surface design, it is important to develop a tool positioning strategy that minimizes the error between the designed surface and machined surface.

3.2.1 Tool positioning strategies for Flank Milling

It is generally conceived that a surface is flank millable if it can be approximated by a ruled surface. A ruled surface is formed of two boundary curves B(u) and T(u) (see Figure 3.8) and an infinity of straight lines between corresponding point (same u parameter) on these curves. It has however been demonstrated by C.Y. Wu in 1995 that the ruled surface criterion is not sufficient to guarantee that a surface is flank millable in the conventional *single-pass* manner [15]. Fans blades for example, have highly twisted blades than can never-the-less be approximated by a ruled surface. However, owing to the sever twist of the surface along a straight line element, the flanked milled surface (generated in a single pass) may deviate quite significantly from the ruled surface. Wu et al. proposed for such components the *multi-pass flank milling* method that is discussed in the next section.

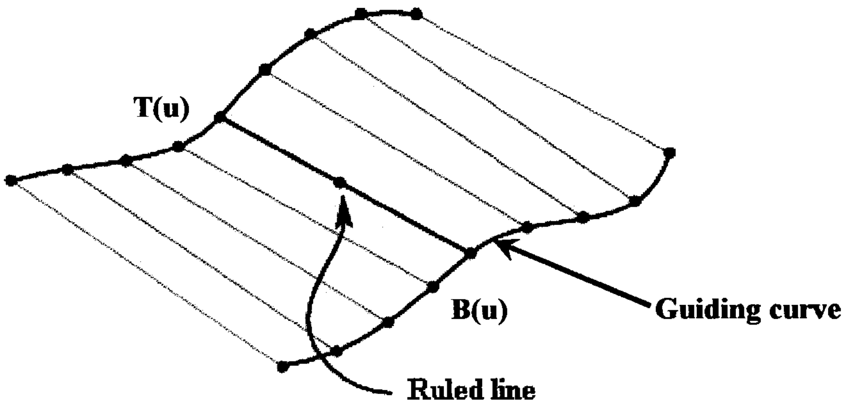


Fig. 3.8 Ruled surface

Many researchers have proposed different flank milling approaches for machining ruled surfaces in the past 10 years. Some examples of authors that present different algorithms of positioning the tool on such a ruled surface are: Bohez et al., Liu, Tonshoff and Rackow, Redonnet et al., Tsay and Her [8]. Each of their proposed solutions will not be discussed in this paper. However, it can be mentioned that each of these methods tries to minimize the discrepancies between the ruled surface and the tool positions through a minimization process. More recently, Bedi et al. proposed the "Rolling Cylinder method" [8] that deserves a closer attention.

This is a three step method that minimizes the deviation distribution between the ruled lines and the machined surface, as shown in Figure 3.9.

Step 1. The tool is initially placed, along a ruled line, tangentially to the two boundary curves (**B(u) & T(u)**). Referring to Figure 3.10, the maximum deviation between the machined surface and the ruled line occurs at constant parametric value w =0.5. Only the boundary curves of the surface are machined exactly.

Step 2. A first optimization takes place in the direction of the ruled line (constant parametric u values). This will ensure that the deviations distribution along the ruled line and the machined surface are minimized in the w direction, as portrayed in Figure 3.10. The tangency points will be shifted inward on the ruled line, thus introducing regions of overcut. This makes the tool tangent to two new isoparametric curves Surf(u,$\hat{w}_1$) and Surf(u,$\hat{w}_2$) at some values $\hat{w}_1$ and $\hat{w}_2$.

Step 3. The second optimization is in the direction of the feed (constant parametric w values). The tool is then "rolled" on the two isoparametric curves in opposite u-parameter directions, until the deviation switches from undercut to overcut (see Figure 3.10). At this instant, the tool will simultaneously be tangent to the surface in 3 points: to the two curves Surf(u,$\hat{w}_1$) and Surf(u,$\hat{w}_2$) and to the ruled line at some $w \in [0,1]$. This is considered to be the optimum position of the *Triple tangent Flank Milling Method.*

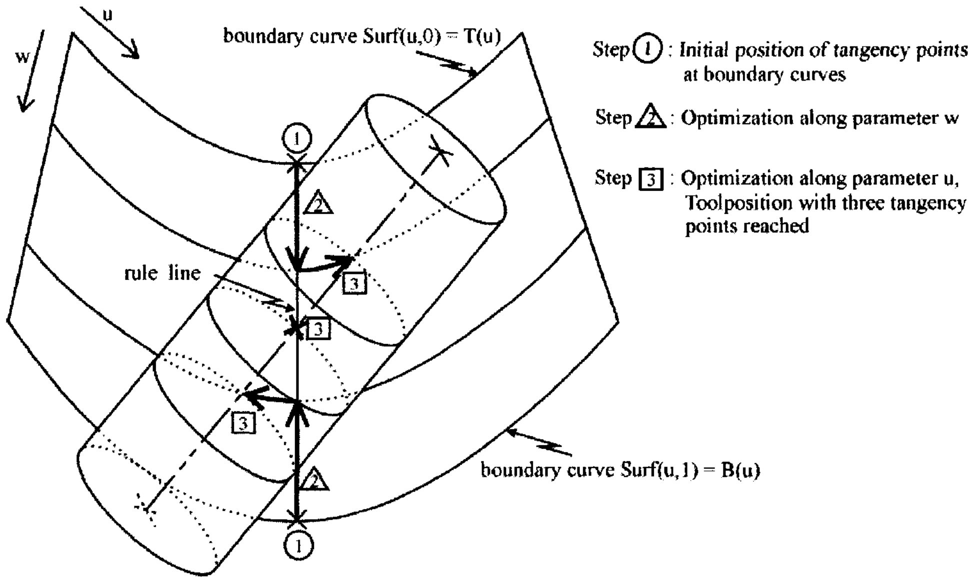

u
w
boundary curve Surf(u,0) = T(u)
Step ①: Initial position of tangency points at boundary curves
Step ⚠2: Optimization along parameter w
Step 3: Optimization along parameter u, Toolposition with three tangency points reached
1
2
rule line
3
3
3
2
1
boundary curve Surf(u,1) = B(u)

Fig. 3.9 Schematic drawing of search algorithm [8]

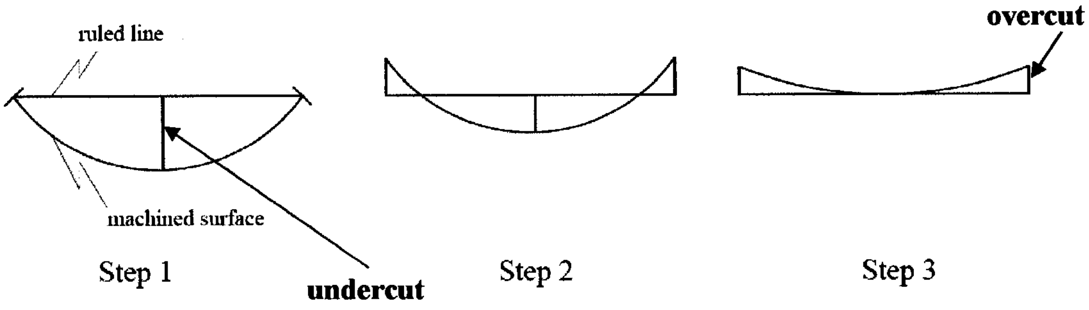


Fig. 3.10 Deviations distribution along the ruled line at each step [8]

3.2.2 Flank milling using CAD/CAM software

CATIA V5 has the possibility to generate a flank milling tool path to machine complex surface using endmills tools. The flank contouring code is provided to Dassault Systems by NCCS, a California-based NC software firm. According to a product review of CATIA V5 (R16) published by CIMdata, Dassault Systems "does not plan to have multiple granular cycles [for flank milling], but instead offer more generic cycles" [3]. The flank milling possibilities offered by CATIA are very limited, and insufficient to correctly machine complex aeronautical components like turbine blades, impellers and integrally bladed rotors. Major engine manufacturers have developed specific cycles for 5-axis machining of airfoils components. According to this product review, among the short-term priorities of IBM/Dassault for further product development of CATIA V5 is included an "increased investment in 5-axis machining" [3]. It can be supposed the future versions of CATIA V5 would possess more complete machining modules and could be suitable to machine complex aeronautical components, however, in the mean time, major aircraft manufacturers rely on home-made software.

3.2.3 Arbitrary surface flank milling program (ASFM)

In order to be more productive and reduce the manufacturing costs, Pratt & Whitney Canada has invested considerable efforts to expand the applicability of flank milling to cover all types of axial compressors.

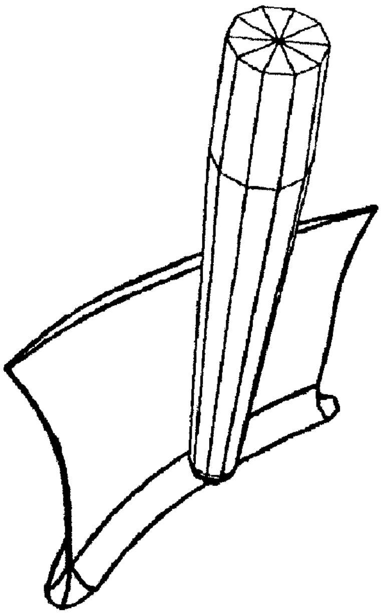


Fig 3.11 Single pass flank milling of an IBR blade [15]

In the flank milling method, the entire blade surface is generated by a single passage of the cutter through the blank material. This engages all the points along the conical cutting portion of the tool as well as the ball-nose surface, as illustrated in Figure 3.11. Some blade surfaces are very twisted and cannot be flanked milled by a single pass of the cutter due to the large deviations between the tool and the desired surface (even if this surface can be approximated by a ruled surface).

However, by imposing three design curves to lie on a ruled but highly twisted surface (instead of two, at the bottom and at the top of the blade), the surface can be flanked milled by the technique of multiple passes. Referring to Figure 3.12, this technique separates a twisted surface into two (or more) smaller and less twisted surfaces, and each one can be flanked milled. There is, obviously, a tangency condition at the joint of the two surfaces. Using this principle, C. Y. Wu and his team [15] were able to flank mill moderately complex rotors that can not be machined in a single flank milling pass.

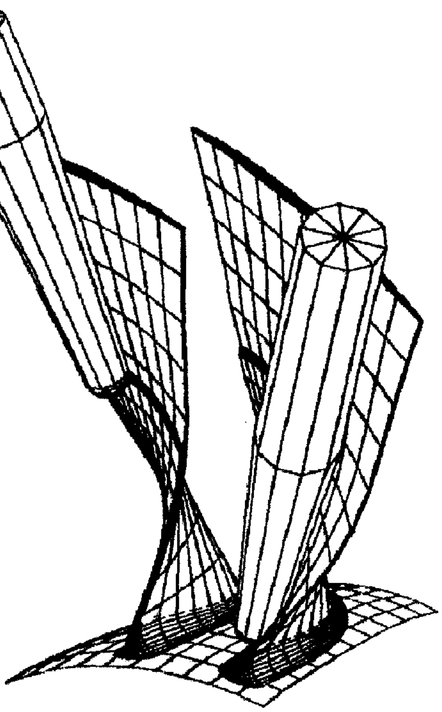


Fig. 3.12 Two pass flank milling for more complex blades [15]

Following the successes of this experiment, C. Y. Wu et al. started the development of the Arbitrary Surface Flank Milling (ASFM) system at PWC. This system's primary role is to transform an actual arbitrary blade design into a similar flank millable blade. By an iterative process, the designer can identify a blade design that is flank millable and possesses similar (or superior) performances and structural integrity as the original blade design.

The ASFM system comprises about 30 batch programs, each one performing one special function belonging to one or more of the following three logical phases of the system:

- Test for Flank Millability;
- Detailed Machining;
- Manufacturing Concerns.

Phase 1 - Test for Flank Millability

In this phase, a blade design is given as input into the system, in the form of curves on the pressure side and suction side. The system generates a *probable* flank millable blade design with the associated tool path. Based on the results obtained, the user can rapidly decide if the blade has a good chance to be flank millable or not. In the case of an affirmative answer, the second phase follows.

Phase II - Detailed Machining

Phase I determined the cutter trajectories that give a probable flank milled surface. In phase II each enveloping surfaces, associated with the cutter trajectories, are fine tuned to yield a flank milled blade possessing the same aerodynamic performances and structure integrity as the original blade design. Depending of the complexity of the blade, this iterative process can take a few days and a dozen of iterations until a correct flank millable blade is obtained.

Phase III – Manufacturing Concerns

Phases I and II only concern the finishing cut. Phase III determines all the other operations of the actual material removal process, like the roughing and the semi-finish cuts. These passes are simply generated by offsetting the cutter positions from the finishing cut away from the blade. This phase of the process permits also the user to have a way to cope with eventual blade and cutter deflections, by making adjustments in the orientation and positions of the cutter. The adjusted tool orientations, composing the flank milling tool path, compensates for the blade deflections and no overstock is generated.

Introducing the ASFM program at PWC represented a major step forward, in terms of productivity and cost reduction. In addition, it eliminated the need for rework or hand polishing of the blade surface after machining.

3.3 New IBRs roughing method

3.3.1 Tool development

3.3.1.1 Cup mill approach

The development of a new tool concept for roughing IBRs began at PWC ten years ago with the work performed by Mr. Ioan Sasu, Ph. D. at the manufacturing engineering department at Pratt and Whitney Canada [18]. The new tool, called **cup mill**, was designed to respond to the following technical and economical criteria:

- more rigid,
- more productive,
- good chip evacuation,
- cost effective,
- rational use of carbide

A simplified form of a cup mill is presented in Figure 3.13. This tool is composed of a hollow cylindrical body that is rotating about its axis, to which inserts are attached at the base of the tool. The rotation of the inserts is generating a surface identified as the cutting surface and it is the active portion of the tool. The main characteristic of this tool is its rigidity and it is expected that this tool would improve the productivity of machining compressor blades and, thus, significantly reduce their manufacturing cost.

The cylindrical body of the tool is made of steel, which imparts, as was mentioned, a high rigidity of the system, with the following consequences:

- higher productivity;
- reduced blade deflections due to the tool kinematics (see chapter 5);
- more aggressive cutting parameters.

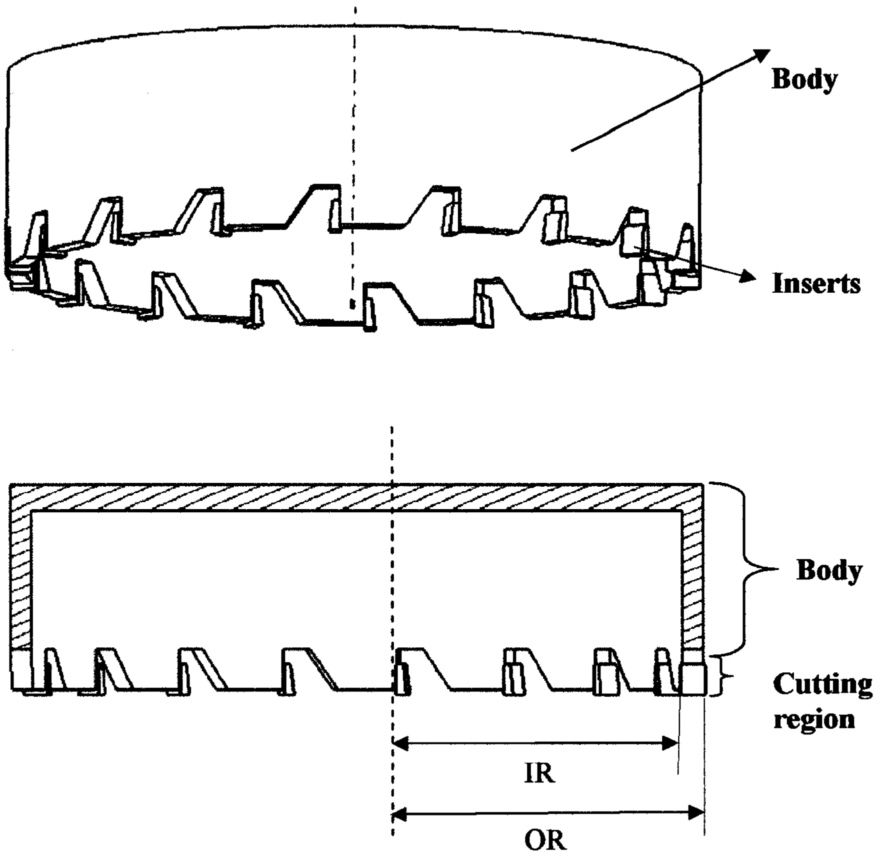


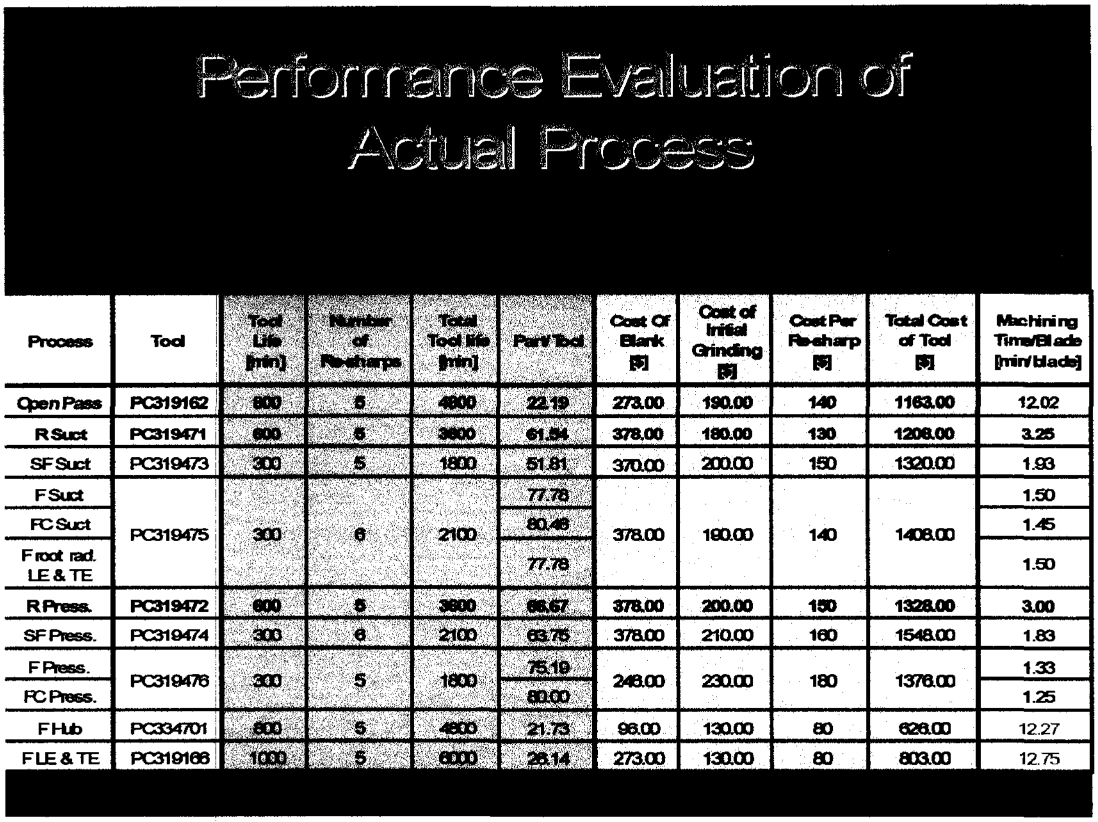
Fig. 3.13 New tool concept for roughing IBRs.

The cup mill has a diameter that can ranges between 7 to 10 inches. This allows mounting up to 20 inserts on the tool. The use of inserts procures important advantages when compared to full carbide ball endmills:

- more economic,
- easy to change,
- easy to get supplies.

Due to the high number inserts (20 vs. 3 or 4 teeth for ball endmills), this tool possesses a longer tool life. Using inserts, instead of manufacturing special ball endmills, also reduce the dependency upon the Cutter Grind department.

Table 3.1 Evaluation of the productivity and costs of current tools (PN3070531-01)



Performance Evaluation of Actual Process

Process	Tool	Tool Life [min]	Number of Re-sharps	Total Tool life [min]	Part/Tool	Cost Of Blank [$]	Cost of Initial Grinding [$]	Cost Per Re-sharp [$]	Total Cost of Tool [$]	Machining Time/Blade [min/blade]
Open Pass	PC319162	800	5	4800	22.19	273.00	190.00	140	1163.00	12.02
R Suct	PC319471	600	5	3600	61.54	378.00	180.00	130	1208.00	3.25
SF Suct	PC319473	300	5	1800	51.81	370.00	200.00	150	1320.00	1.93
F Suct	PC319475	300	6	2100	77.78	378.00	190.00	140	1408.00	1.50
FC Suct					80.46					1.45
F root rad. LE & TE					77.78					1.50
R Press.	PC319472	600	5	3600	66.67	378.00	200.00	150	1328.00	3.00
SF Press.	PC319474	300	6	2100	63.75	378.00	210.00	160	1548.00	1.83
F Press.	PC319476	300	5	1800	75.19	248.00	230.00	180	1376.00	1.33
FC Press.					80.00					1.25
F Hub	PC334701	800	5	4800	21.73	96.00	130.00	80	626.00	12.27
F LE & TE	PC319166	1000	5	6000	26.14	273.00	130.00	80	803.00	12.75

In Table 3.1 is presented, for a selected IBR (P/N 3070531), the tool life and the manufacturing costs of currently used taper ball endmills, as well as the machining time per blade of the different operations. It can be noticed that the most time consuming operations are the open pass, the finishing of the hub and the finishing of the trailing and leading edge. The unproductively of the hub, leading and trailing edge finishing operations can be explained by the fact that these portions of the blade are still machined using the point milling method. However, among the operations involved in the machining of the blade surface (flank milling operations), the open pass and the roughing of the blades have the longest machining time (more than 18 minutes per blade). The scope of this project is to replace all these roughing operations using taper ball endmill cutters, with a single roughing operation using the cup mill cutter that is 3 to 4 times more productive.

3.3.1.2 Cup mill first prototype

In 1999 started the production of a first prototype of the cup mill cutter, base on the work performed by Mr. Ioan Sasu and collaborators. This tool was made from 4140 steel and has 20 inserts. Referring to the CATIA model presented in Figure 3.14, holes were drilled into the body of the tool to permit the delivering of the cooling fluid from the spindle to each insert individually. The cup mill has an outside diameter of 10.125 inches and an inner diameter of 8.875 inches, giving an effective cutting with of 0.625 inches. This tool is designed to be used with a BT50 holder.

The inserts are alternatively positioned in the inside and outside of the tool so, ten of the inserts were left end (LFEW3534FLLN) and ten were right end (LFEW3534FRLN). They are carbide (uncoated) inserts of grade KC510M and were manufactured by Kennametal. The inserts are directly mounted on to the body of the cup mill without the use of cassettes. This prototype was finished by the end of 2002 and a picture of it is shown in Figure 3.15.

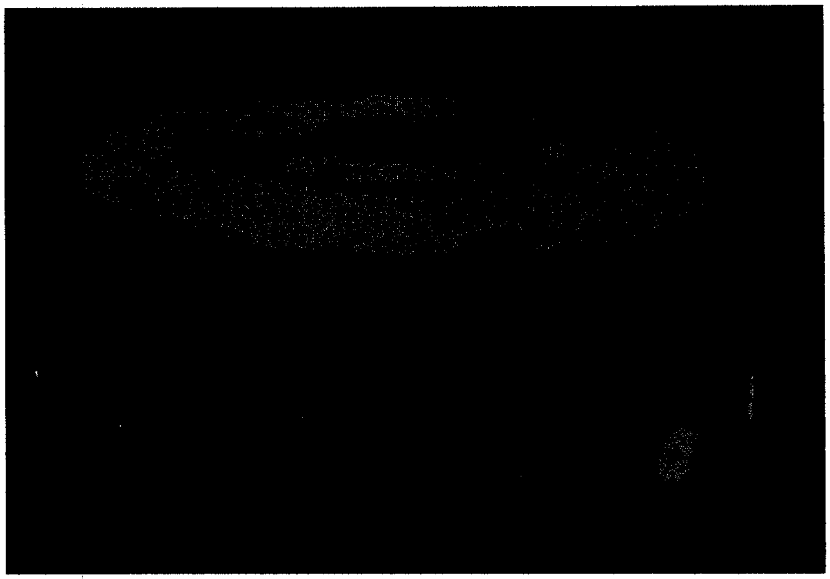


Fig 3.14 CATIA Model of the Cup Mill cutter developed at PWC

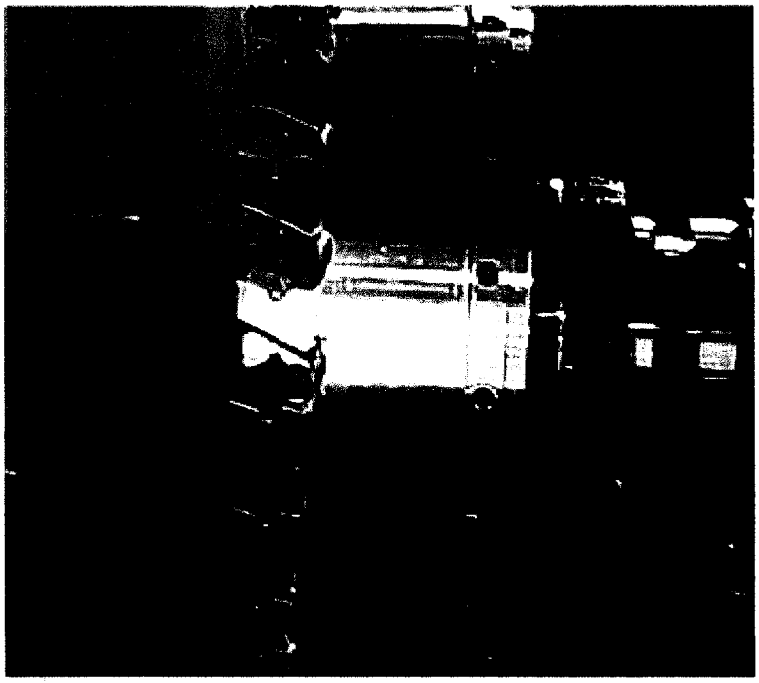


Fig 3.15 Cup Mill produced at PWC

3.3.1.3 Titanium machining test

Some concerns were expressed on the capability of such a large tool to sustain the large forces generated during the machining of hard titanium parts. To dissipate these concerns, the cup mill was tested under the most extreme conditions possible: plunging directly into a titanium test piece. This is to simulate the conditions of the tool during the open pass phase of the roughing, where the cutter is fully engaged into the workpiece and the machining forces are at a maximum. The titanium work piece used for the test was of cylindrical shape (half of cylinder), because it is a close approximation of the shape of a blank IBR part. The experiment consisted in simply plunging into the test piece with the tool to a predefined depth (thus creating a slot), indexing the part and plunging again. The results of this test (shown in Figure 3.16) were very encouraging: the initial expectations concerning the robustness and the productivity of this tool were demonstrated.

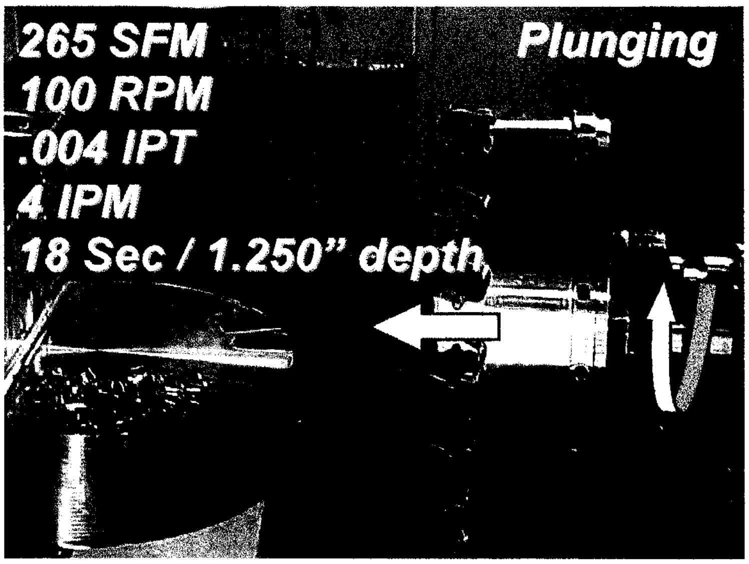


Fig 3.16 Evaluating the performances of the cup mill for machining titanium

The cup mill cutter was patented by PWC in 2005.

3.3.2 Program development

A machining test was conducted on an actual IBR part using the cup mill cutter, to demonstrate that this tool is capable to generate complex surface composing the blades. Since this tool concept is new, and very different from the actual tool used in production, no suitable CAD/CAM software exists that can program the cup mill cutter for such a complicated task. An algorithm was required that could generate the tool path to rough an IBR using this tool.

3.3.2.1 Software used

A first version of the algorithm was developed in 2004 by Marius Petean [10]. This is the first software capable to generate a tool path for roughing IBRs using a cup mill cutter.

The elaboration of the algorithm using only a computer language like C++ or FORTRAN is very complicated and requires extensive programming and mathematical skills. The drawings of both the IBR part and the tool have to be transformed into mathematical formats understandable by the programming language. On the other hand, by using simultaneously the CATIA V4 software and a programming language, the problem is greatly simplified. This combination is called *interfacing*, which is a mean of interaction between two systems.

More particularly, in CATIA V4, Application Programming Interface (API) is a set of sub-routines used programmatically to manipulate CAD data. Standard programming languages can be used to access these API functions. CATIA V4 allows the usage of API for specific data extraction, software customization and automation purposes.

For the first version of the algorithm, the FORTRAN programming language was used in combination with CATIA V4. CATGEO, which stands for CATIA Geometric Interface Fortran routine is the application that permits this indirect communication between the user and CATIA V4 by using a computer program (FORTRAN in this case).

3.3.2.2 Tool Kinematics

The tool positioning strategy of this fist version of the program [10] is fairly simple, since the tool orientation of the cup mill is kept constant throughout the machining of the blade (3-axis machining). For a constant tool orientation, the algorithm finds the corresponding tool positions coordinates, following a defined strategy, which will be described below. Also, when the direction of the tool axis is not changing, collisions between the part and the tool can easily be avoided. This implies that the algorithm does not require any collision detection procedures (no corrective measures are necessary), since the tool kinematics are designed in that scope.

Before the tool path computation, the algorithm has to determine the tool orientation. The program creates some geometric constructions on the part that will be described here; see Figure 3.17 below.

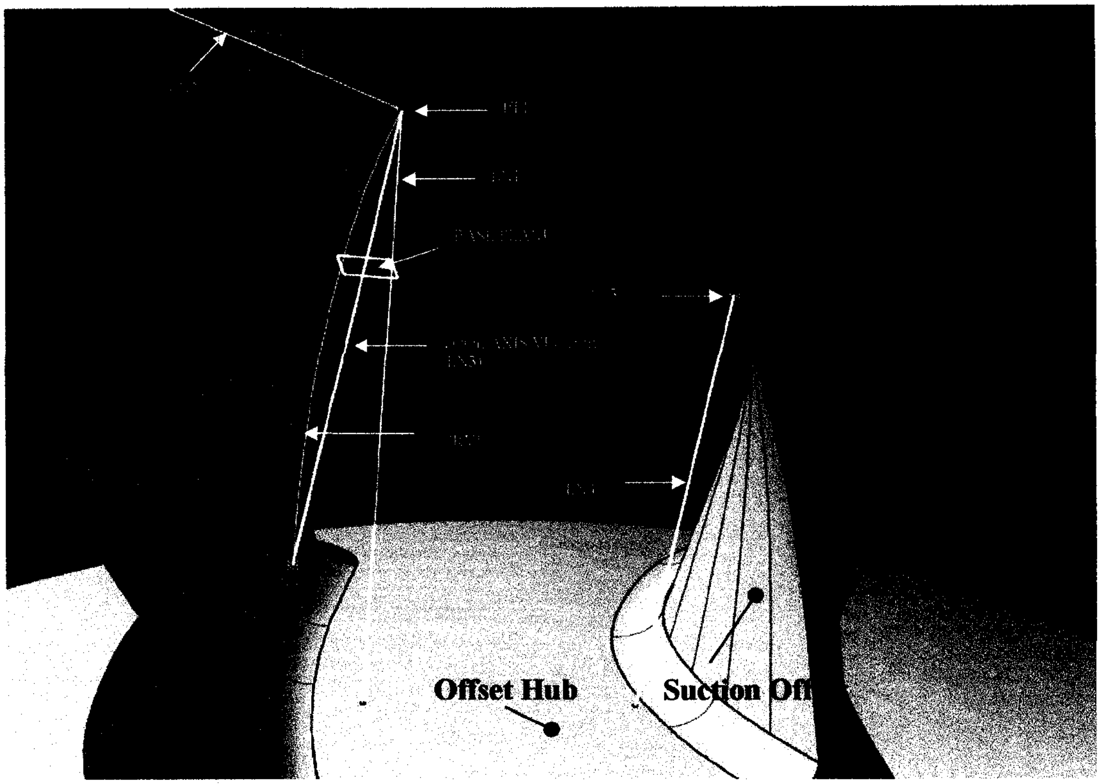


Fig. 3.17 Tool axis finding strategy

The program starts by offsetting the pressure and suction side faces, as well as the hub (or gas path) surface, toward the interior of the space between blades, by a small increment (0.035 inch in this case). These offset faces are the limiting surfaces for the tool and ensure that no gauging in the blade will occur. From the point PT1 (see Figure 3.17) on the offset pressure side, the lines LN1 and LN2 are created, which are two lines passing by PT1 and are normal to the hub and to the pressure side surfaces respectively. The plane defined by these two lines intersects the pressure side surface (offset) along the curve CRV1. The line LN3, connecting the end point of the curve CRV1 is defined as the tool axis direction. A parallel to this line, passing by point PT3, is defined on the suction side – LN4.

Once the tool axis orientation is determined, the path generation process begins. The tool kinematics selected in this version of the program excludes any possibility of collision between the tool's body and the part. In this scope, the space between the blades is machined level-by-level. Referring again to Figure 3.17, the tool is sliding on the "Base Plane" (normal to the tool axis); this plane is translated downward by a small increment, starting the next level. At each level both the pressure and the suction side of the blade are machined one after the other. According to M. Petean, by keeping the tool axis constant and machining the blades level-by-level permits to have a good control over the cup mill and reduces the risk of collisions.

At each stage, the tool is first positioned on the pressure side of the blade and then is translated, on the "Base Plane", to the suction side of the blade (or vice versa). However, this translation from one side of the blade to the other one is interrupted by a third intermediate position, called the "Base Position". The *Base Position* is the position of the tool, for the already determined tool axis direction, limited at the outer circumference by the line LN1 and on the inner circumference by the line LN4. The role of the base position is to ensure that the material is completely removed from the blade surfaces around the regions that are the most probable of entering in contact with the cup mill body. The corner PT1 on the pressure side and the corner PT3 on the suction side are the most advanced edges of the blade, and thus, it is around these points that a collision with the tool is likely to occur.

3.3.2.3 Tool path generation strategy

Figure 3.18 represents the cross section of two blades machined by the cup mill cutter at a particular level of the tool path. On this figure, the cup mill is in the base position. The outside diameter of the tool is tangent to the line LN3 and the inside diameter of the tool is tangent to the line LN4 (both are parallel to the tool axis). Note that in Figure 3.18 the view portrayed is in the direction of the tool axis, and, as a result, the line LN3 and LN4 are observed as two points: PT1 and PT3. It can be also observed that the positions of the tool at the base position, at the different levels of machining, are following a line that is parallel to the tool axis.

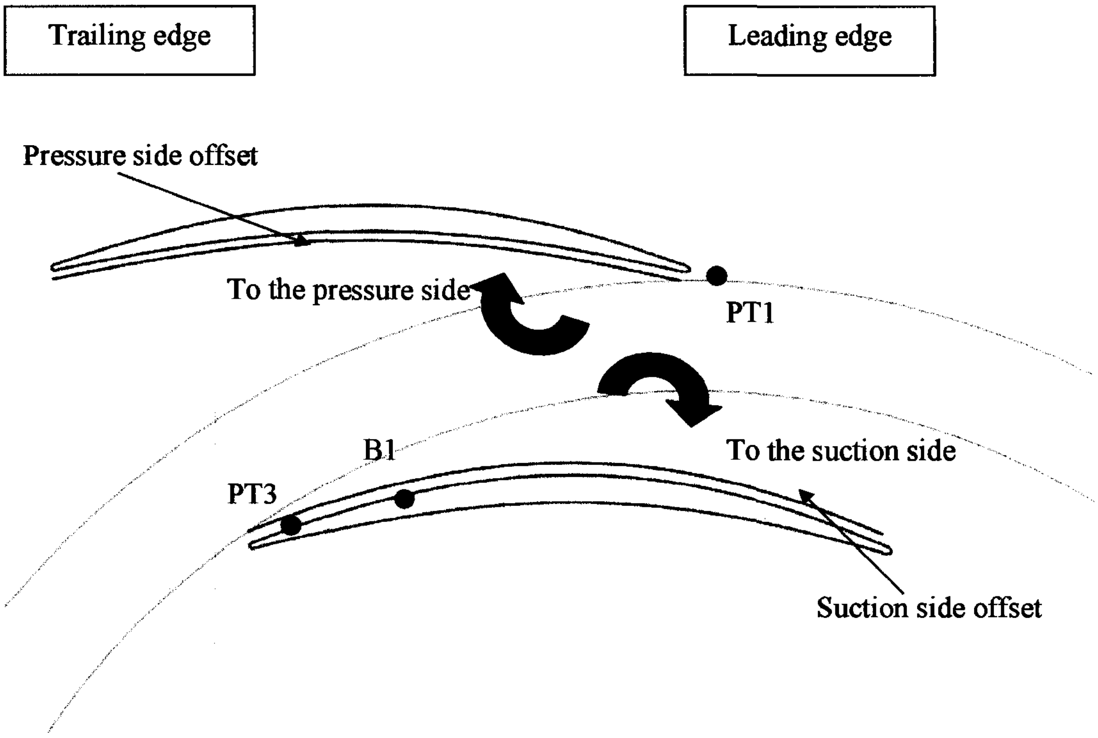


Fig. 3.18 Tool path finding strategy – Base position

A clockwise rotation of the cup mill about PT1 (i.e. line LN3) will bring the cup mill closer to the pressure side of the blade. This is the second position of the tool at a level of the tool path: the *Pressure Side position*. Due to the more pronounced curvature of the suction side, rotating the cup mill about point PT3 (line LN4) will not bring the cup mill close enough to the leading edge portion of the blade. Therefore, a new point B1 is defined that is on the inner circumference of the cup mill at the base position, see Figure 3.18. This point is at a certain distance from the suction blade side. M. Petean's master thesis [10] does not specifies clearly how this point was determined; it simply mentions that it is at a "sufficient distance from the blade" so, a rotation about this point, brings the cup mill close to the leading edge of the blade. This is the third position of the tool path: the *Suction Side position*.

At each level, the cup mill is positioned on the pressure side of the blade, then rotates about PT1 to the base position and rotates again about B1 to the suction side. The cup mill is lowered by a defined increment to the next level (toward the hub), and this cycle is to be repeated in the reverse direction. The algorithm stops when the bottom of the blade is reached by the tool (a collision with the hub is detected).

A portion of the tool path is presented in Figure 3.19 for the levels "i" and "i+1". As mentioned above, at each level, the tool does not go directly from the pressure side to the suction side (or vice versa), but will pass trough the "Base Position" fist, generating a "zigzag" motion. Also, from this figure it can be noted that the Base Positions are collinear. The transition between the cup mill positions is executed by a circular interpolation.

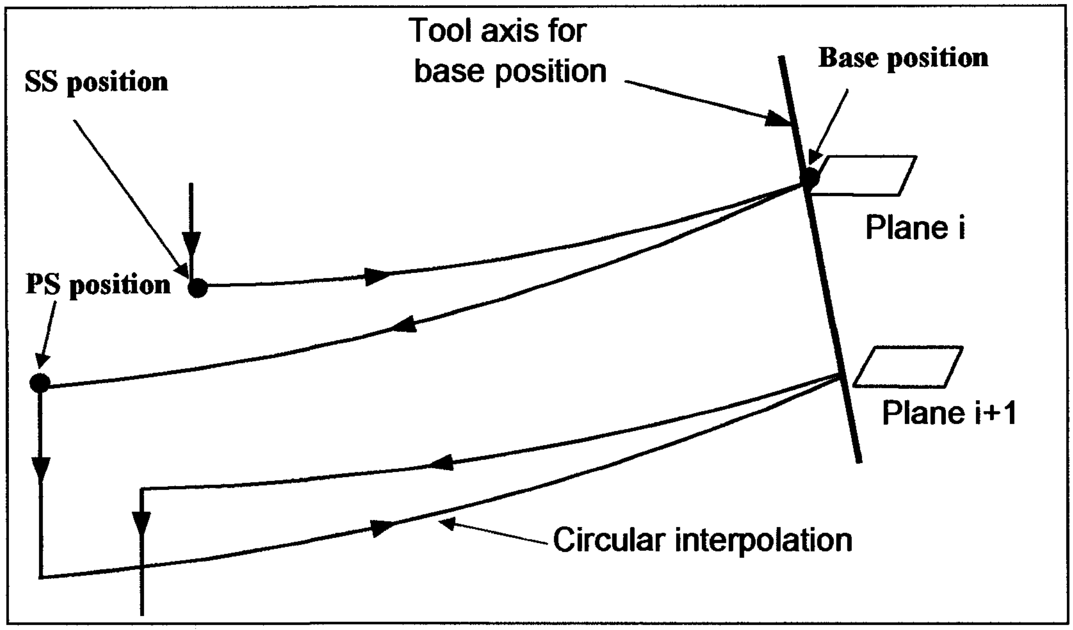


Fig. 3.19 Tool kinematics

3.3.2.4 Testing the algorithm developed by M. Petean

Machining tests were carried out using the tool path generated by this first version of the program, with the cup mill build at PWC. The IBR selected for the test was a second stage compressor from the PW 500 series engines and a picture is presented in chapter 2 (see Figure 2.3). For safety and economic reasons, the blank material for this part was not made of titanium but rather from aluminum. This test was performed on a 5-axis Hu80 machine tool.

A picture of this test is presented in Figure 3.20 and the results obtained are presented in Figure 3.21.

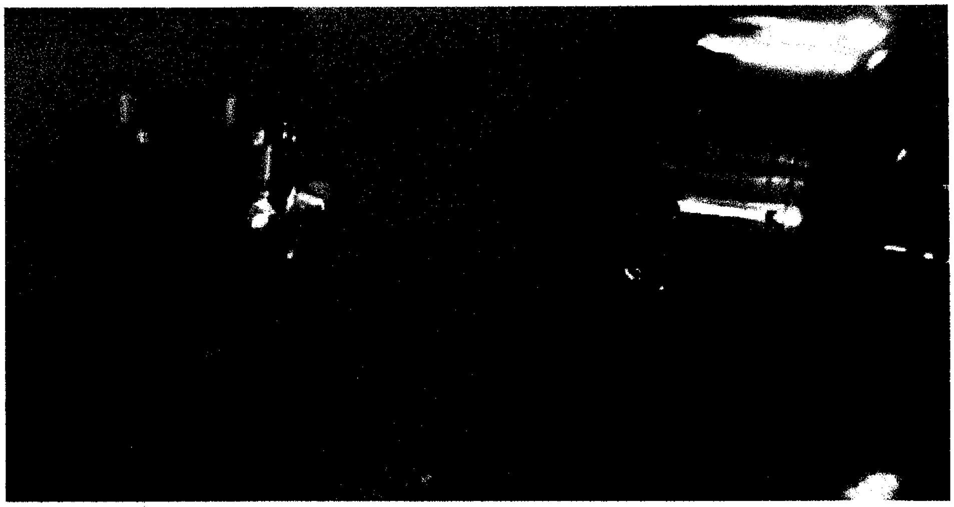


Fig 3.20 Machining test

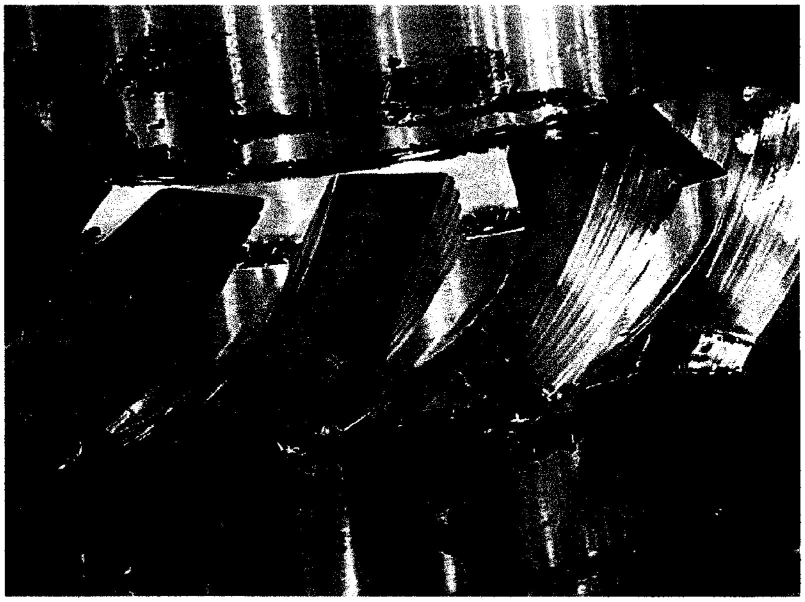


Fig. 3.21 Results of cutting test using the algorithm [10]

3.3.2.5 Limitations of the actual program

A close analysis of the presented algorithm reveals major drawbacks which limit the productivity and the accuracy of the machining process using a cup mill:

- The tool positions are determined on the base of an arbitrary and intuitive geometrical construction, and not on the base of any logical criteria;
- The kinematics of the tool is inefficient (long tool path).

First, the determination of tool axis orientation is questionable. The axis is obtained by performing a geometric construction on the part, but the logical steps that lead to the development of this construction are not explained. There are no reasons to consider that this tool axis is optimal (as mentioned by Marius) because the volume of removed material can be further increased by finding another tool orientation. There exists a tool orientation that allows a maximum volume of material to be removed (for 3-axis machining). This tool axis is optimal (and unique) and could be found through an optimization process. We consider that this approach is the correct solution for a fixed axis machining strategy.

There is another important aspect to be considered: the machining precision obtained using a 3-axis tool strategy might not be sufficient to justify the replacement of the current cutters with the cup mill, even if the tool has the optimal axis (fixed throughout the blade machining). A better precision can be obtained by using a 5-axis machining strategy. Based on this strategy, the tool path permitting a maximum volume removal has to be determined. In this case, only a modification of the dimensions and geometry of the cup mill permits to further improve the precision of the roughing process. The tool dimensions and geometry can be "optimized" (compromise between access and rigidity) to remove a maximum volume of material.

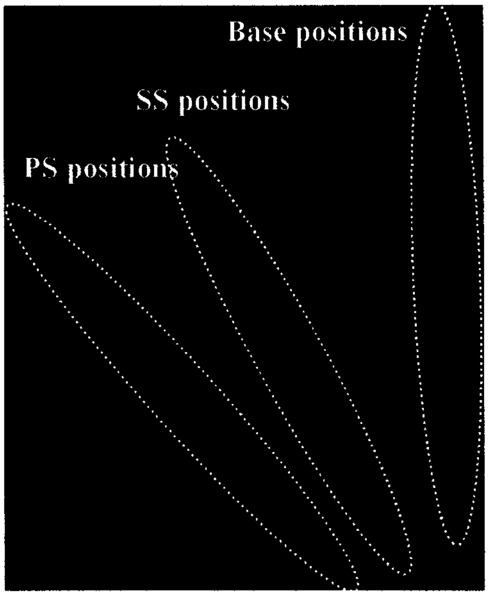


Fig. 3.22 Complete tool path generated

The second drawback of this first version of the algorithm is the proposed tool kinematics. The complete tool path generated by the algorithm is presented in Figure 3.22. The three different cup mill positions are identified in this figure: pressure side, suction side and base positions. It can be observed that the tool path is long due to the back-and-forth movement of the tool from one side of the blade to the other side. A major part of the manufacturing costs of IBR is related to the machining time of the parts. It is essential that the proposed method of roughing IBRs (cup mill) is productive, in order to justify its validity. A new tool motion that is more efficient has to be found.

In conclusion, the current algorithm fulfills only three of the five major requirements as presented in Table 3.2. The main objective of this paper is to determine a new algorithm which satisfies all the requirements.

Table 3.2 Requirements evaluation of the algorithm developed by M. Petean [10]

Requirement	**Alg. Version #1**
Generic	Yes
No interferences	Yes
Optimum (Max. Volume removed)	**No**
Productive	**No**
Smooth tool path	Yes

Even if this version of the algorithm is not suitable to introduce the cup mill tool in production at PWC, it nonetheless succeeded in its primary objective, namely:

➢ **Demonstrate the feasibility of roughing IBRs blades using cup mill cutters.**

3.4 Compressors machining using annular cutters

A new concept of roughing integrally bladed rotors blades is introduced by Hunt in 1994 [6]. The tool comprises two annular cutters, rotating about their axes, which are parallel but not coincident (Figure 3.23). One annular cutter surrounds the other one. The interior annular cutter cuts the workpiece to form the concave airfoil surface and the exterior cutter generates the convex part of the same blade. The annular cutters 20, 30 are moved relatively to each other by pair of rotating cams 56, 62, while their rotation axis remains parallel. Both annular cutters can be moved independently, which permits an adjustment of space between them so that the shape of the convex and concave aerofoil surfaces of the blade 16 may be varied.

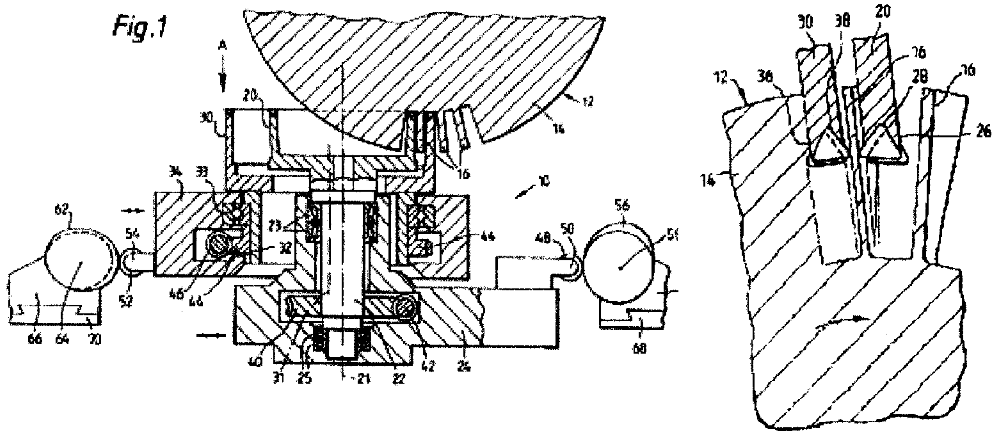


Fig. 3.23 Roughing IBRs using annular cutters [6]

Both the cup mill cutter method and the method proposed by Hunt are machining IBRs using annular cutters. In the twin annular cutter method, both side of the blade are machined in a single tool motion. On the other hand, using the cup mill cutter, two separate tool motions are required to machine both sides of the blade since only one annular cutter is used.

However, Hunt's proposed method imposes the cutters axes to remain parallel to each other which limit the machining precision on the blade. At a defined level of machining, the difference of blade geometry on the pressure side and suction side could require a different cutter orientation to be closely machined, which cannot be simultaneously accommodated if the cutters axes are parallel.

Using a 5-axis milling strategy, this tool could be very difficult to program to generate complex blades surfaces since two cutters are present and each cutter has to be controlled independently.

3.5 Conclusions of bibliographic review

In this project, we seek to develop a new tool, along with the path generation algorithm, that is to be used for a very specific application: roughing of IBRs. Unfortunately, not many papers, in the technical literature, could be found of significant relevance to this research since this proposed roughing process is very different than the currently employed methods. The point milling and, especially, the flank milling methods propose a tool positioning strategy that tries to minimize de deviation between the tool and the blade surface. Such an approach could be used to develop the cup mill algorithm, since it is required to remove a maximum of material from the blade surface in a single pass of the cutter.

To introduce the Cup Mill project in production, two important issues have to be resolved:

1. **New tool**

   A new cup mill tool developed in collaboration with major tool suppliers that could be used in production (production tool).

2. **New Program**

   - More productive
   - Better machining precision

CHAPTER 4: Tool development for IBRs machining with cup mill

4.1 Tool designs

4.1.1 Design requirements

In spite of the good results obtained during preliminary tests performed with the prototype of the cup mill, this tool presents some technical limitations which are, in the same time, subjects of future improvements. The testing first tool design at PWC permits to make the following conclusions:

- There is a potential for the project
- Better tool rigidity

At this step we are required a tool suitable to be introduced in production. The observed issues of the tool prototype developed at PWC were transmitted to major tool suppliers in order to develop a suitable production tool.

First of all, the diameter exceeds the maximum size permitted by the tool magazine of the machines currently in production at PWC. The outside diameter is 10.124 inches and the maximum tool diameter accepted by the machines is no more than 9 inches. The insertion of the tool into the spindle of the machine had to be done manually by the operator. The tool weight is about 50 lbs, which largely overpass the maximum weight imposed by workplace safety regulations, which limits the allowable weight that can manually be manipulated by an operator at 25 lbs.

It is to be noted that the cup mill, in its prototype form, respects weight and power limitations of the current machines at PWC, which can support heavier tool weights that

50 lbs and the power consumption of the tool during machining is relatively small, due to the low spindle rotational speed.

Past the physical limitation, the cup mill design must justify itself by an increase in productivity from the actual method and a reduction of the manufacturing costs. This tool is very productive, but the tool life of the inserts is short. The technical input of the tooling suppliers is essential here to maximize the tool life of the inserts. In addition, these inserts have to be standard. Standard inserts are less expensive than custom made inserts, the supply is guarantied and only small quantities are required to be kept in inventory.

The design requirements of the cup mill are summarized in Table 4.1:

Table 4.1 Design requirements of the cup mill

Requirements	Advantages
Standard inserts and cassettes	Reduce tool cost Reduce inventory Guarantied tool supply
Fit the cutter in machine-tool magazine Respect actual machine-tool weight and power limitations	Easy implementation
Increase productivity Reasonable tool life	Reduce manufacturing costs

4.1.2 Production tool proposals

Three major tooling manufactures were contacted, namely Sandvik, Iscar and Kennametal. Each supplier proposed a design for the cup mill, in accordance to the specified requirements. Each one of the potential suppliers presented its own design that will be discussed here.

4.1.2.1 Tool designed by Sandvik

The tool developed by Sandvik (see Figure 4.1) has an outside diameter of 8.66 inches and an inside diameter of 7.48 inches. It has 16 half effective square inserts. These are carbide inserts and have a double layer PVD coating of TiAlN-TiN. This tough, wear resistant coating, applied on a very fine grain carbide substrate, provides a sharp cutting edge to the insert. The inserts are not directly mounted on the body of the cup mill, but are fixed on cassettes. The reason for this is that in the case of an accident, the body of the tool (most expensive) would not be damaged and only the broken cassettes needs to be changed. Both inserts and cassettes are standard (Sandvik standard) so, their cost is relatively low.

The coolant delivering system is similar to the one of the tool made at PWC: channels drilled in the body of the tool deliver the coolant to each individual insert.

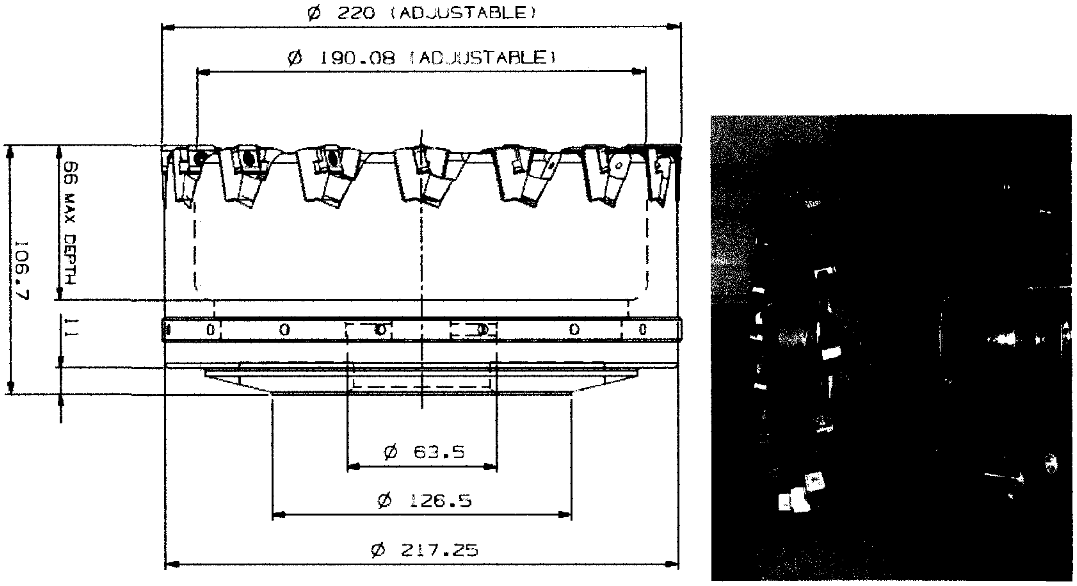


Fig. 4.1 Cup Mill design proposed by Sandvik

It is interesting to mention than the cassettes can be adjusted on the tool body by sliding them toward the interior or the exterior of the tool, as shown in Figure 4.2. This is an efficient and inexpensive method of controlling the runout of the inserts.

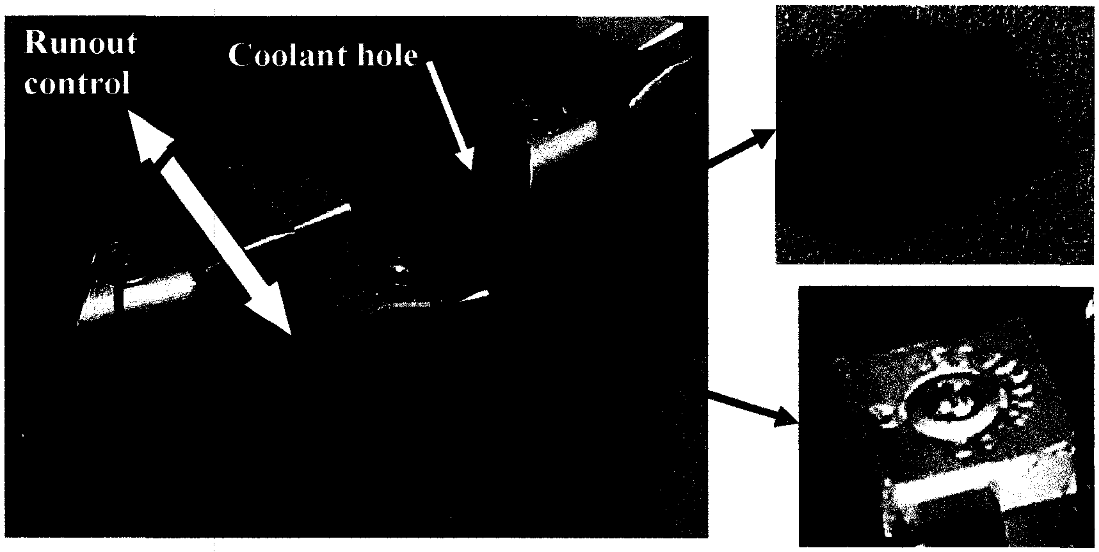


Fig. 4.2 Adjustable cassettes (runout control)

The inserts are well supported in the cassettes both in the tangential direction as in the radial directions, as portrayed in Figure 4.3.

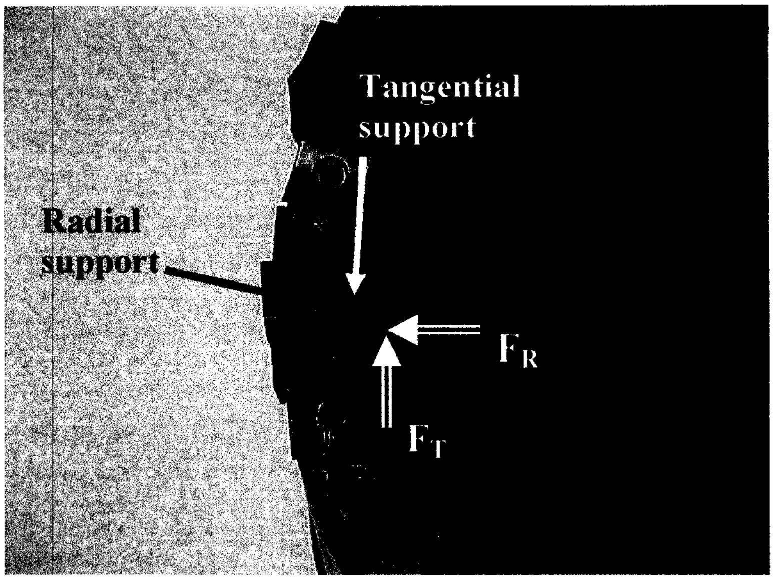


Fig. 4.3 Inserts support

The price of the tool body is 5000$ (first prototype).The price of the inserts is 11 $ each.

Conclusions

Table 4.2 Sandvik tool evaluation

Advantages	Disadvantages
Standard inserts Easy maintenance Use of cassettes Runout control	High tool body cost

The tool design proposed by Sandvik is excellent. The inserts proposed are standard Sandvik inserts so their cost is low (11 $/insert). Also, each inserts comprises 2 cutting edges (indexes). This design permits an easy runout control of the inserts. The only negative aspect of this proposal is the high cost of the body, which is about 5000 $. However, since the inserts are mounted on cassettes (not directly on the tool body), the tool body is expected to be used to produce a large quantity of parts before it is needed to be replaced.

4.1.2.2 Tool designed by Iscar

The tool developed by Iscar is presented in Figure 4.4. This tool has an outside diameter of 8.50 inches and an inside diameter of 7.40. It has alternating left and right end inserts that are directly mounted on the body of the cup mill (no cassettes). The carbide inserts are not coated and are specially made for this tool, so they are expected to be more expensive. The cooling system is again similar to the one presented before.

It is interesting to note that this cup mill is, however, slightly thinner then one presented by Sandvik by 0.040 inch, and 0.075 inch thinner than the cup mill developed at PWC. This could be an advantage over the tool proposed by Sandvik because it allows a greater freedom of movement of the tool between the compressor blades (but only if the productivity and rigidity are not affected).

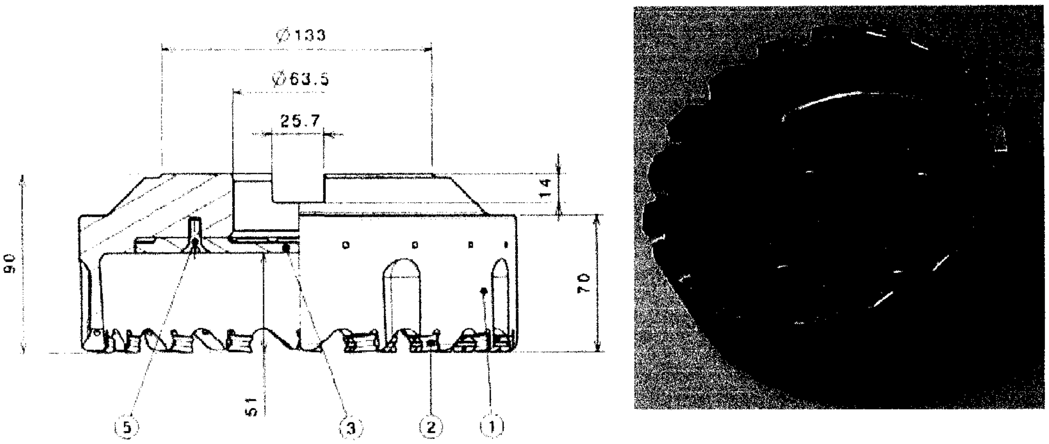


Fig 4.4 Cup Mill design proposed by Iscar

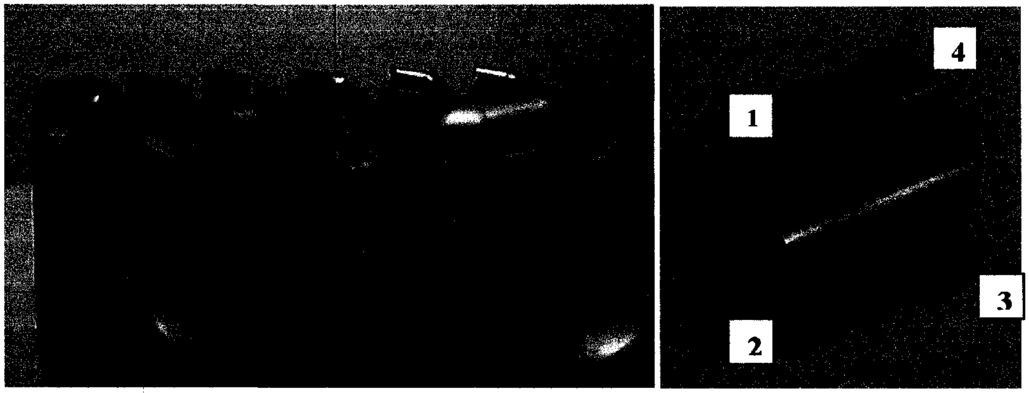


Fig. 4.5 Special Iscar inserts

The inserts have 4 possible cutting edges. To prevent the inactive cutting edge of the inserts to be damaged during machining, a gap (relief) is designed at the junction of the supporting faces on the tool body, as pictured in Figure 4.5.

Since the inserts are directly mounted on the body of the tool and fastened using a single screw, we have no means of controlling the runout of the tool (depends only on the manufacturing precision of the tool body). Issues related to the rapid tool wear could arise. In addition, the inserts are well supported in the tangential direction, but only the screw supports them in the radial direction, as portrayed in Figure 4.6.

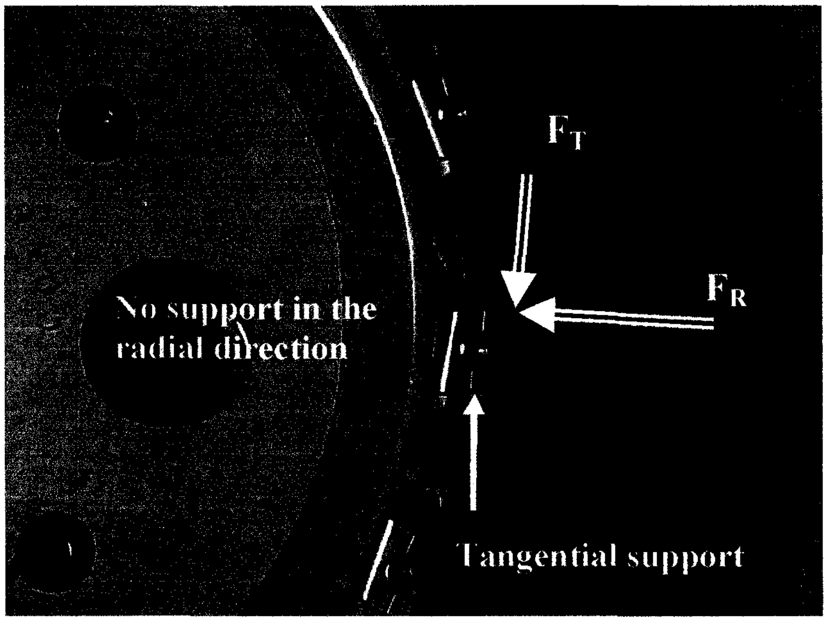


Fig 4.6 Insert support

The tool body costs about 2000 $ and, since the inserts are special, they are more expensive and cost 20 $ each.

Conclusions

Table 4.3 Iscar tool evaluation

Advantages	Disadvantages
Easy maintenance	Not standard inserts (high cost)
Inexpensive tool body	No cassettes
Small width of tool walls	No runout control
	Improper inserts support

The cutting width of the tool proposed by Iscar is less than the design proposal of Sandvik by 0.04". This permits a better machining precision due to a greater freedom of movement between the blades of the IBR. The inserts are special Iscar insert so they are more expensive (20 $/insert). However, each inserts have 4 cutting edges, so their price is comparable to the Sandvik inserts that are less expensive but have only have 2 cutting edges. The tool body is inexpensive (only 2000 $), but it does not permit any runout control or the inserts. At the current time, Iscar did not perform any machining test in titanium with this tool, so the performances are unknown. If they are comparable (or better) to the performances of the tool proposed by Sandvik, this design could be preferred due to the low tool body cost.

4.1.2.3 Tool designed by Kennametal

Based on the information available at this time, Kennametal produced a tool with 20 standard inserts mounted on cassettes. The inserts are made from coated carbide and are standard Kennametal inserts. The inserts are mounted on cassettes, but the position of the cassettes on the tool body is not adjustable (no runout control possible). The cooling system for this tool is similar to the previous ones. We have been also informed that preliminary test (plunging in titanium) were unsatisfactory and the development of this tool is, therefore, stopped for the moment.

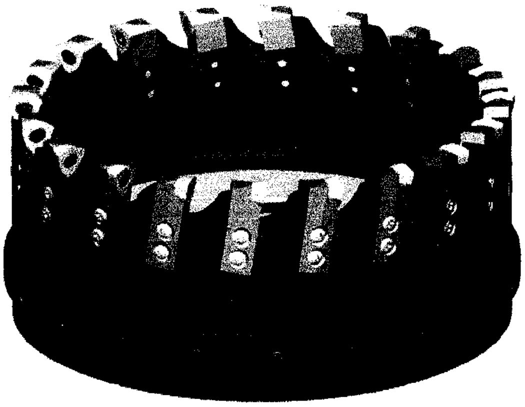


Fig 4.7 Cup Mill design proposed by Kennametal

Conclusions

Table 4.4 Kennametal tool evaluation

Advantages	Disadvantages
Standard inserts Maximum number of inserts (20) Easy maintenance Cassettes	No runout control

Like the Iscar tool, the prototype proposed by Kennametal has no means to control the runout of the inserts. The runout measured during the tests was 0.003", which exceeds the chip thickness of 0.002". This implies that some inserts are not cutting, while others remove more material than intended. The runout problem alone is not sufficient to explain the poor tool life experienced during the machining tests: only one slot could be machined. This suggests that the inserts geometry and the inserts coating is inappropriate. Until major modifications are made to the proposed design by Kennametal, this cup mill cutter cannot be considered as a future IBR roughing tool.

Table 4.5 Test results for Kennametal

Positive aspects	Negative aspects
• None	• Runout up to 0.003" (chip thickness 0.002") • Inappropriate insert coating • Inappropriate insert geometry

Of all 3 proposals (Sandvik, Iscar and Kennametal), Sandvik's tool design seems to be the most interesting.

4.2 Sandvik tool performance evaluation

It is essential for the viability of the project to demonstrate a significant increase in productivity and a reduction of manufacturing costs with the introduction of the new technology in production. In that sense, each potential supplier needs to evaluate the performances of their respective design. It was already mentioned that Kennametal failed the test (very short tool life) and Iscar has not performed any tests to evaluate the tool life and productivity of its tool at the current time. Consequently, only the results of the tests performed by Sandvik are discussed here. The results of this test will influence the decision concerning the selection of the supplier of the cup mill and of the inserts among the three tool manufacturers.

The testing methodology used is the same for each of the three potential suppliers: each cup mill successively plunges into blocks of titanium to evaluate the tool life of the inserts and tool's productivity. Figure 4.8 illustrates a cup mill plunging into a cylindrical block of titanium, thus generating a slot of a predefined length. The test piece can be indexed and a new plunging is performed.

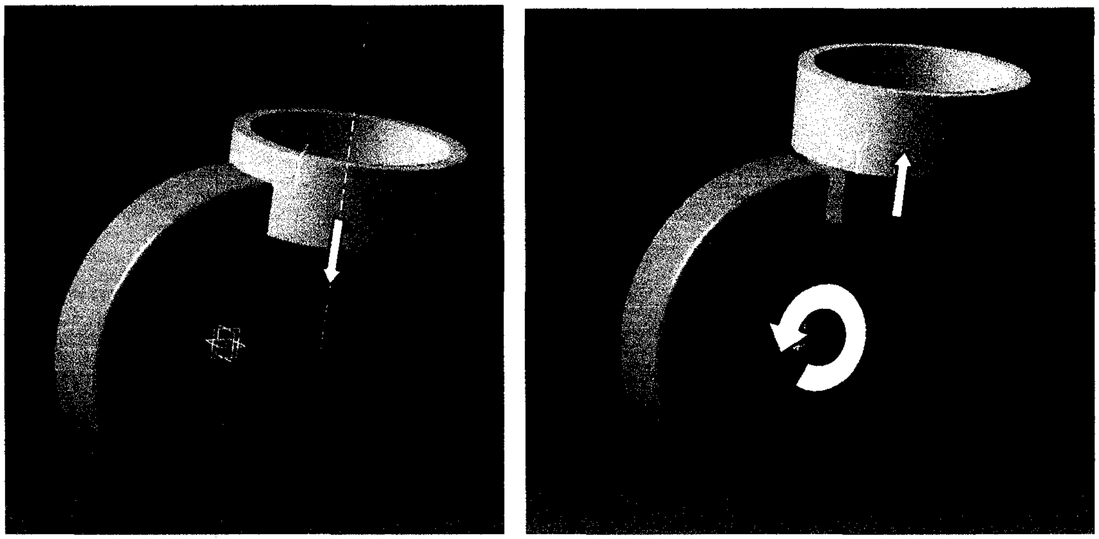


Fig 4.8 Testing the cup mill – plunging into a titanium part

This method of plunging into the material with the whole width of the cup mill is a fast and cheap way of evaluating the productivity and the life of the tool because it exposes the tool to the most extreme cutting conditions.

4.2.1 Sandvik

A 5-axis Morei Seiki NH-8000DCG machine was used to perform this test at Sandvik and the slots generated are 1.77 inches deep (45 mm). The cutting parameters selected for this test were selected by Sandvik and are based on their experience with this type of inserts to obtain a maximum productivity and tool life.

The machining parameters are presented in Table 4.6 and a picture of the test is presented in Figure 4.9:

Table 4.6 Sandvik test – Cutting parameters used

Speed	1772 in/min
	65 RPM
Feed	0.031 IPR (0.00394 IPT)
	4.05 IPM
Coolant	11 gal/min
Machine	Morei Seiki NH-8000DCG

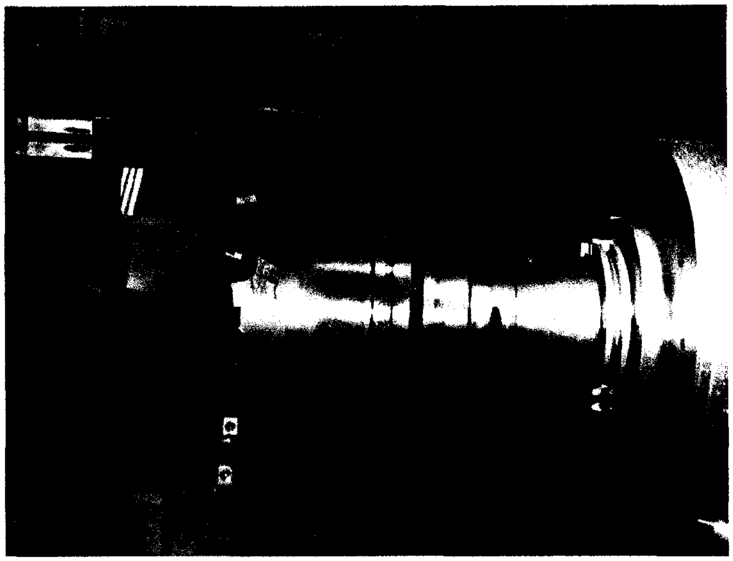


Fig 4.9 Sandvik test – Plunging into a titanium block

Sandvik's tool machined 131 slots, 45mm deep, in the blocks of titanium (see Figure 4.10) using the same index of the inserts. The wear of the inserts (a picture is presented in Figure 4.11) is still very small; it can be supposed that the inserts are still suited for machining. However, due to the high cost of titanium it was decided to stop the tests after 131 slots. Each slot of 4.5 cm in depth was generated in 50 seconds.

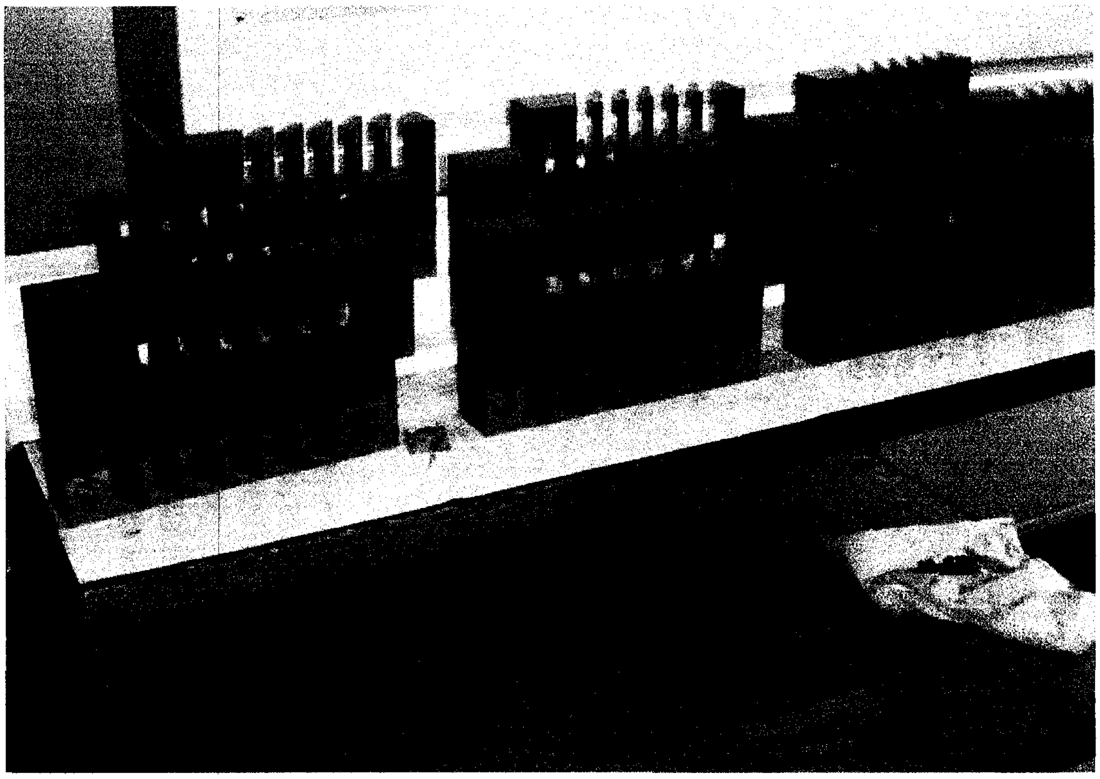


Fig 4.10 Sandvik test – Machined Parts

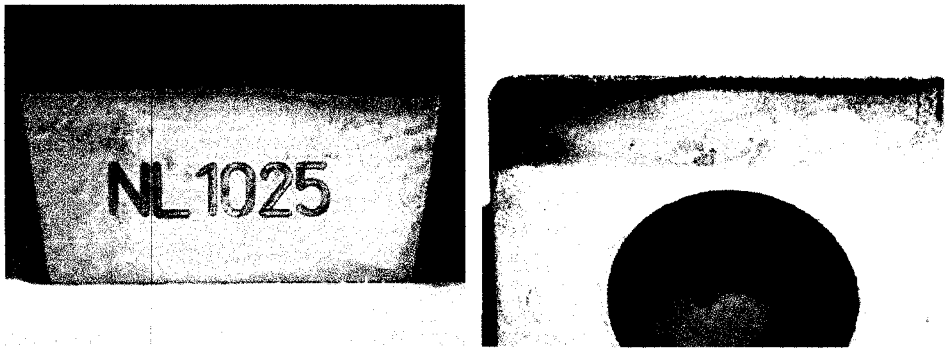


Fig 4.11 Sandvik test – Inserts wear

4.3 Productivity and cost evaluation

The results of this test can be used to evaluate the performance of this tool to rough IBRs. In section 2.3 we analyzed the performance of the actual roughing process for a selected part number (30J2429, see Fig. 2.3). For comparison purposes, the same part number is used here to evaluate the performance of this new technology. In particular we are interested to evaluate: the productivity increase and the cost reduction of the new process versus the actual method.

Table 4.7 Performance evaluation of Sandvik tool – IBR part # 30J2429

Process	Tool	Number of Inserts/Tool	Cost/insert [$/insert]	Number of Edges/insert	Inserts cost/Tool [$]	Part/Tool (2 edges)	**Machining Time/Blade [min/blade]**	Tool Life (2 edges) [min]
Roughing	Cup Mill SANDVIK	16	11	2	176.00	16	**2.00**	608

Table 4.7 presents an estimation of the productivity of this tool to manufacture a 30J2429 part number IBR. The next sections describe how the values in the table were calculated.

4.3.1 Tool life

Considering that 131 slots 1.8 inches deep (4.5 cm) could be generated with the same insert index, if each slot was instead 1.2 inches deep (length of blade), more than 200 could be generated in a similar manner. This corresponds to perform the open pass operation on more than 10 IBRs (19 blades/IBR). Including also the pressure side and the suction side machining passes (that remove far less material), we can realistically estimate that 8 IBRs can be roughed using the same index of inserts (or 16 IBRs / set of inserts). Note that the subsequent passes remove less material than the open pass (less wear).

Currently, for this part number 3 different tools are involved in the roughing process. The tool performances and their manufacturing costs are tabulated in Table 2.1.

The tool used for the open pass is re-grinded after 10 parts and the tools used for roughing the pressure and suction side, after 16 and 12 parts respectively. In other words, in average one cutter (rougher) needs to be re-grinded after 4 IBRs produced. Based on our estimation, 4 times more parts can be roughed using a set of inserts than using a new hog taper ball endmill (16 vs. 4).

4.3.2 Productivity

It is also interesting to note that these slots were generated in about 50 seconds each (depth of 4.5 cm). Using again the same reasoning as before:

- Open pass operation could be performed in 33 seconds (for a slot 1.2 inches deep);
- Estimated total roughing time (open pass + pressure side + suction side): maximum 2 minutes per blades.

When comparing the volume removal rate of the cup mill cutter to the current production tool, it is observed an increase in productivity by more than 3 times. This volume of material is removed in 7 minutes using the actual tool compared to 2 minutes using the cup mill cutter.

4.3.3 Tool and machining cost reduction

The inserts supplied by Sandvik cost 11 $/insert. Since there are 16 inserts per tool, the total tooling cost is evaluated at 176 $. It was estimated previously that 16 IBRs can be roughed using one set of inserts, so the tool cost per part for the new process is 11 $. The actual tool cost is calculated from Table 2.1, and is 42.3 $/part.

The machining costs reduction is calculated using the actual hourly machining cost at PWC, which is 200$/hour. From Table 2.1, the total roughing time of the actual tools is 2.11 hours (6.66 min/ blade) for the part number 30J2429. In comparison, the estimated total roughing time for the new process is only 0.63 hour (2 min/blade). This represents a 1.5 hours reduction of the roughing time and 296 $ of savings per part.

The comparison between the actual roughing process and the new proposed one is summarized in Table 4.8.

Table 4.8 Cost evaluation of Actual vs. New process

	Actual Process				[redacted]			
		Machining						
Process	Tool Cost/Part [$/part]	Time/Part [Hrs/part]	Cost/Part [$/part]	Total Cost/Part [$/part]	[redacted]	[redacted]	[redacted]	[redacted]
Roughing	42.3	2.11	422.43	464.73	[redacted]	[redacted]	[redacted]	[redacted]
				Savings	(32)	(1.5)	(296)	(327)
				Savings [%]	76%	70%	70%	70%

4.4 Inconel 718 tests

To improve the performances and efficiency of aircraft engines, manufacturers are constantly researching new material that permits an increase of the internal temperature of the engines. Producing IBRs from inconnel 718 using the current tool is very expensive due to the short tool life.

The cup mill tool could be a solution to this problem. Machining tests in inconel were performed at Sandvik, using their proposed tool. Similarly to the titanium tests, the cup mill cutter plunged into blocks of inconel 718.

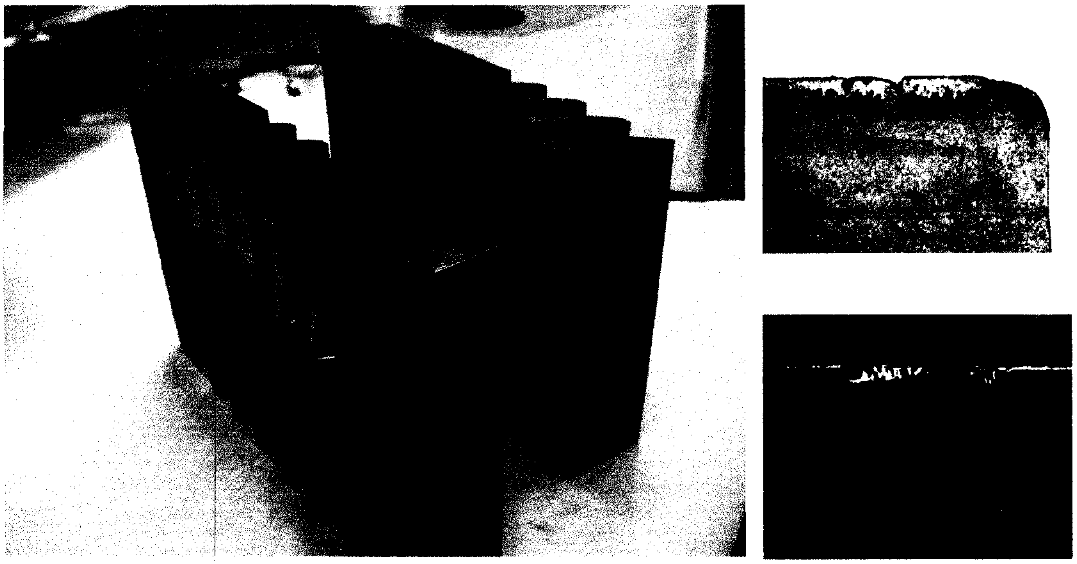


Fig. 4.12 Tests in Inco 718

Using the same inserts (same index), they performed 12 slots to a depth of 45 mm (1.8 inches) in 100 seconds per slot. The inserts were worn-out after 12 slots, as portrayed in Figure 4.12.

Table 4.9 Inconel test conclusion

Process	Tool	Number of Inserts/Tool	Cost/insert [$/insert]	Number of Edges/insert	Insert cost/Tool [$]	Part/Tool (2 edges)	**Machining Time/Blade [min/blade]**	Tool Life (2 edges) [min]
Roughing	Cup Mill SANDVIK	16	11	2	176.00	1	**4.50**	**86**

Again, referring to the part number 30J2429, we estimate that 12 slots to a depth of 1.8 inches is equivalent to perform the open pass operation on 18 blades of this part (1.2 inches deep). The time required for the open pass is 68 seconds. Considering the pressure and suction side machining passes, we obtain a maximum of 4.5 minutes of roughing time per blade (4 x 68 sec).

Based on these results, one set of inserts (both cutting edges) could be used to rough one IBR (all passes). At 11 $ per inserts, the tool cost for producing one IBR is 176$/part. The cup mill tool could be a solution of producing inconel IBRs in a relatively fast time and at a low cost.

Table 4.10 Cost evaluation of inconel IBR (P/N 30J2429) using cup mill cutters

Process	**Tool cost**	**Machining Time**	**Machining Cost**	**Total Cost**
	[$/part]	[hrs/part]	[$/part]	[$/part]
Open Pass	176	1.43	285	461
R Hub				
R Suction				
R Pressure				

Based on this estimation, roughing an IBR in inconel can be performed faster using the cup mill cutter that what it currently takes for a titanium part using the actual tools (1.42 hrs vs. 2.11 hrs, see table 4.8). The tooling cost is much higher (176 vs. 42.3), however, the total manufacturing cost is relatively similar (≈ 460$ / part).

In conclusion:

- Actual process:
- Impossible to produce inconel parts – unacceptable high price
- New process:
- Can produce IBRs in inconel at the same price as titanium parts using actual method

.CHAPTER 5: Program development for IBR machining with cup mill

5.1 Software used (CATIA V5 – VB6)

Most engineering software packages expose pre-built functions through which we can access database and other routines. Standard programming languages can be used to access these Application Programming Interface (API) functions. Some advantages and scope of API are:

1. API functions allow the user to extend the CAD system capability by programming additional features into it;
2. allows selective information to be extracted quickly and interactively;
3. user need not depend on any third party software tool for data extraction.

The first version of the program took advantage of the possible "interfacing" between CATGEO [10], using the FORTRAN programming language, and CATIA V4. A similar approach is used in this paper. In our case, the newer version of CATIA (V5) is used along the programming language Visual Basic version 6.0.

It is possible to use solely a programming language (like C++ or even MatLab), but the solution would be more complex. The difficulty comes from the fact that the CATIA parts have to be published into corresponding mathematical models usable by the program. There are 2 groups of 3-D publishing formats: polygon-based and NURBs 3-D representation.

Polygon-based 3-D uses polygons to approximate mathematical perfect curves. Thus, the accuracy of the 3-D data in a polygon-based format is dependant on the level of tessellation [17]. In general, the lower the tessellation, the lower the accuracy of the model, and vice versa with high tessellation. In Figure 5.1 is presented a polygon-based CGR part (CGR is the CATIA 3D publishing). For easier visualization, polygons constituting the part were selected (orange triangles).

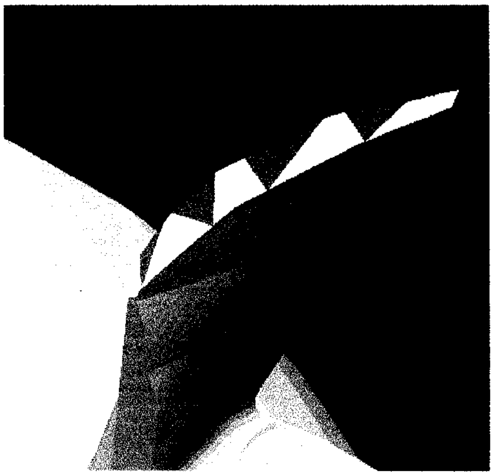


Fig 5.1 Polygon-based part

The second possible case of 3D publishing is the XVL (eXtensible Virtual world description Language) format, developed by Lattice3D. Instead of using a polygon-representation of 3D models, XVL "re-surfaces" 3D model with a form of NURBS called Gregory Patch. XVL provides very high fidelity, because it is based on pure mathematical model that follows the curves very exactly, thereby enabling good communication of the data at all levels of zoom. This is a relatively new technology, so its use is still very limited.

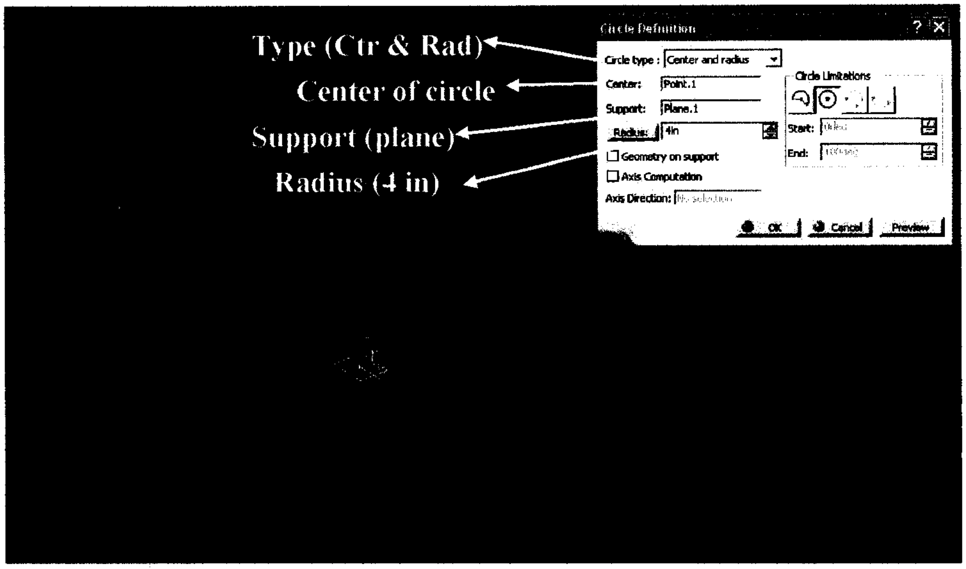
As mentioned, both the cup mill and the IBR CATIA parts would have to be first transformed into a 3D file format, to be usable by the computer language. The program would have to be written in such a manner as to be compatible with the mathematical representation of the parts, thus complicating the algorithm considerably. At this point of the project, it was decided, for simplicity, to use the CATIA environment since the functions required are built-in into the software. The possibility of using 3D published format of the part, rather than the CATIA part, will be discussed at the end of this paper.

Just as in the first version of the algorithm, both the IBR part and the tool are manipulated in CATIA (V5), using the programming language Visual Basic 6.0. Visual Basic enables the user to access the API function of CATIA V5 (note that VB can perform operations for Excel in a similar manner). Using the API function of CATIA, the user can program new functions into the CAD/CAM software, where repetitive operations and logical decisions are automatically executed. These operations are, otherwise, impossible to be executed manually.

To explain how this communication between CATIA V5 and VB6 is performed, an example is presented here: the creation of a circle directly in CATIA and through VB6. All operations are similar, so this simple example illustrates very well the general idea.

5.1.1 Example of communication between CATIA and Visual Basic

In CATIA the circle (defined by a center and a radius) would be created in the following manner: from Figure 5.2 the user specifies the type of circle, then center point, a support plane, and finally the radius of the circle.



Circle type : Center and radius

Center: Point.1

Support: Plane.1

Radius: 4in

□ Geometry on support

□ Axis Computation

Axis Direction: No selection

Start: 0deg

End: 360deg

Fig. 5.2 Creating a circle manually in CATIA V5

Performing the same operation through Visual Basic follows the same logic and steps, namely the code would be the following:

```
1 Dim hybridShapeCircleCtrRad1 As HybridShapeCircleCtrRad
2 Set hybridShapeCircleCtrRad1 = hybridShapeFactory1.AddNewCircleCtrRad(reference1, reference2, False, 101.6)

3 HybridShapeCircleCtrRad1.SetLimitation 1

4 HybridBody1.AppendHybridShape hybridShapeCircleCtrRad1

5 part1.InWorkObject = hybridShapeCircleCtrRad1

6 part1.Update
```

Center Support Radius

Step 1

Variable declaration phase. The variable "hybridShapeCircleCtrRad1" (name of variable) is of the type/dimension ("Dim") hybridShapeCircleCtrRad (circle defined by a center and a radius, same as before).

Step 2

Adding a the required **inputs** for the feature:

- to "reference1" is associated the center point (performed previously)
- to "reference2" is associated the support plane (performed previously)
- the circle has a 101.6 mm (4") radius (VB only works in mm)

Step 3

Options declaration phase (if any). The option in this case is to set the limits of the circle (see the dialog box in Fig. 5.2). A full 360° is desired, so the value of the limit is 1.

Step 4

The **geometrical set** in which this circle would be located ("hybridBody1" is associated to a geometrical set, defined previously)

Step 5

The **document** containing the project ("part1" is associated to the current CATIA document, defined previously).

Step 6

Updating the project after the operation (the circle appears on the screen after this step).

All other CATIA operations are performed in a similar manner, following the 6 intuitive steps presented in this example.

5.2 Algorithm

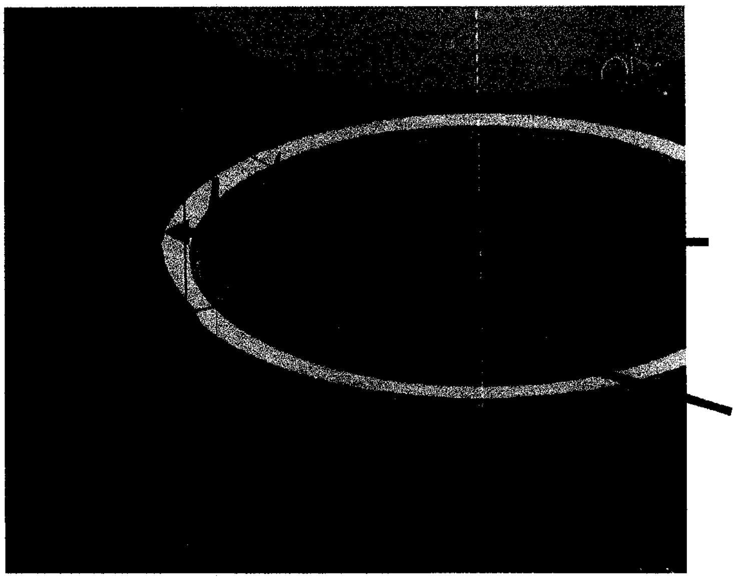
5.2.1 Algorithm particularities

Due to the complexity of the surface of IBR blades, current CAD/CAM software can only machine them using the point milling method. According to Allan Christman from CIMData [3], future versions of CATIA V5 will offer the possibility to machine compressor blades using also the flank milling method, but for now, major aircraft engines manufacturers use homemade software to flank mill their IBRs.

In flank milling, the positional control of the tool is only with respect to the desired target surface and all other surfaces comprising the part are often not even supplied to the program. When machining compressor blades using the flank milling method, it is considered that the endmills used provide sufficient clearance to avoid any collisions with the other surfaces of the part. The user is responsible to verify the tool path, and in the case of collisions, make the necessary corrections (select a different tool, for example).

The difficulty is even greater for programming the cup mill tool, when compared with flank milling, due to the unusual shape of this tool. The cup mill cutter is a very large tool which is not providing the same clearance as the endmills. Furthermore, in the case of the cup mill, both the exterior and the interior of the tool have to be controlled at the same time by the program with respect to the pressure and suction side of the blade as well as the hub surface. In order to machine one blade surface using the cup mill, the opposing blade and the hub surfaces must be considered by the algorithm to avoid any gouging (or collisions) into them. Due to the complexity of this problem, no suitable CATIA function exists.

As presented in Figure 5.3, due to the limited space between blades, both the interior and the exterior of the tool have to be controlled simultaneously.



Machining the pressure side with the outside circumference of the cup

Verify interior circumference for gouging into suction side

Fig 5.3 Simultaneously machining both sides of the blade

In addition to avoid gouging into the part while machining, the software must also ensure that there is no collision between the non-cutting portion of the cup mill (cup mill body) and the part itself. When machining with endmills, the tool cutting portion is long enough to practically eliminate the risk of collision between the part and the tool shank. A simple simulation software is sufficient to ensure that the machining program is collision free. For the cup mill, however, the risk of collision between the tool body and the part is very high because the tool body enters in the space between the blades. Eventual collisions between the part and the tool body must be prevented by the algorithm, which further increase its complexity.

In summary, the specificity of the process requires the elaboration of a new software, due to the following reasons:

- new tool;
- no software currently available;
- simultaneously machining with OD, ID and front of tool;
- 5-axis machining;
- avoid interferences.

5.2.2 Algorithm requirements

The algorithm has to fulfill some key requirement, in order to ensure a correct machining of an IBR blade: the program has to be generic; it has to provide the optimum tool path, which is collision free and smooth.

Generic:

The generality of the algorithm is particularly important. The algorithm strategy must be applicable to different IBRs without the need for any modifications or adjustments to the program. In addition, each of the different part numbers requires cup mills of different shape and size. The algorithm must give the machining programmer the opportunity to change easily the configuration of the cup mill used to machine the particular IBR.

Optimization:

The tool path generated by the algorithm must remove a maximum volume of material. Even if the cup mill is intended to be applicable only in roughing the IRBs, the excess material left on the finished surface of the blade should be minimum. The objective in our case is to have an overstock not exceeding 0.020" to 0.030" anywhere on the blade surface. In addition, this deviation has to be distributed as evenly as possible on the blade surface. The semi-finish, finish and free cut cutters have limited material removal capacities. Consequently, they would not be capable to remove excess material beyond the mentioned values. An uneven distribution of the excess material could result in undercuts, on some portions on the blade surface, and over cuts on others due to the same reasons.

Collision free:

The algorithm must generate a tool path that has:

1. No gauging in the part.

The algorithm has to control simultaneously the inside and outside of the tool with respect to the pressure and suctions side surfaces as well as the hub, and prevent that the tool gauges in these surfaces.

2. No collisions between the tool body and the part

The non-cutting portion of the tool (tool body) has to be compared with the part to detect any collisions, and, if needed, take corrective measures.

Smooth tool path:

It is also important that the tool path generated to be smooth. The successive positions of the tool must respect an order 0 and an order 1 continuity, i.e. point continuity and tangency continuity.

5.2.3 Algorithm strategy

The elaboration of the algorithm is in two steps: new tool positioning strategy on the blade surface and a new tool kinematics. The positions on the blade will be 5-axis positions, as close as possible to the blade surface to permit a maximum volume of material to be removed. The tool kinematics has to efficiently remove this volume of material.

5.2.3.1 Positioning strategy

As it was already mentioned, one of the drawbacks of the 3-axis algorithm is the poor machining precision. Compressor blades have very complex surfaces, that are composed of isoperimetric curves of varying radii of curvature, which can be simultaneously concave and convex, and thus, it is impossible to precisely machine them with a cup mill having a fixed orientation. Therefore, we focused our work on the elaboration of a 5-axis tool positioning strategy.

A parallel can be made between the positioning strategy developed for the flank milling method and our particular case. For this reason, a similar approach will be considered: positioning the tool on the blade surface to minimize the "mismatches" between the surface and the tool. This is required for all the positions of the tool path. If this condition is respected for all the surfaces to be machined, the resulting 5-axis tool path removes a maximum of material between successive blades.

In Figure 5.4 is illustrated, in a 3-D representation, the differences between positioning the cup mill using a fixed tool axis strategy (3-axis) and the optimum solution (5-axis). Cross sections at 3 different levels of machining are taken for the 3 tool positions shown to illustrate the mismatches between the blade surface and the tool, for the 3-axis and 5-axis positioning strategies. Due to the imposed fixed axis, in the first case, the tool cannot be positioned closer to the blade, without intersections. In the second case, the tool orientation is "adapting" to the geometry of the space available and, for each level of the tool, it is as close as possible to the target surface.

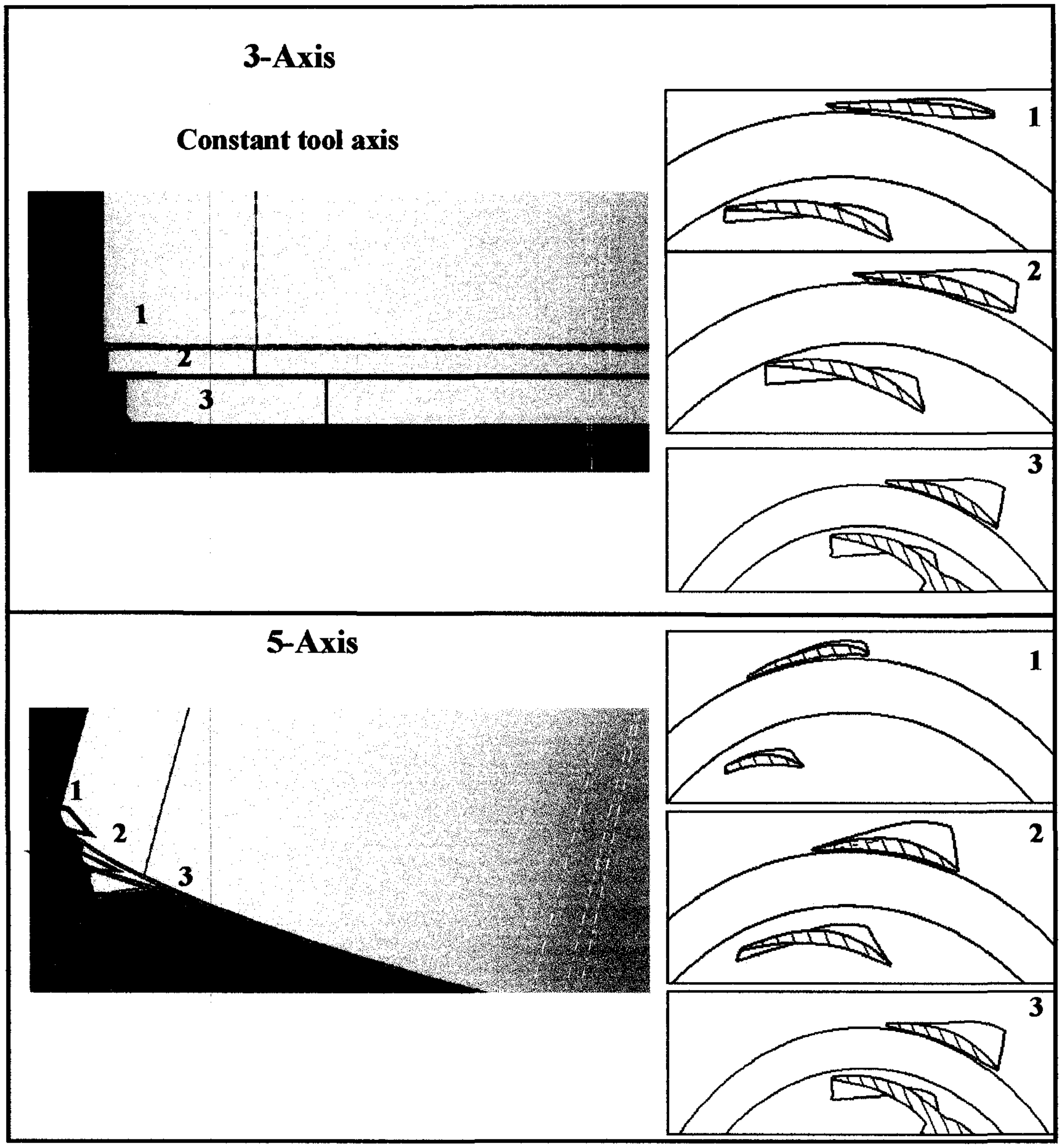


Fig. 5.4 Comparison between a 3-axis positioning strategy and the optimal, 5-axis one.

The cup mill can be separated into two portions: the cup mill body and the cup mill cutting surface (see Figure 5.5). The cup mill body is composed of the non cutting portions of the tool and the cutting surface is the surface swept by the rotating inserts. It is only the cutting surface of the cup mill that is required to be positioned as close as possible to the blade surface, since the rest of the tool is not cutting. The non-cutting portion is verified for eventual collisions with the part.

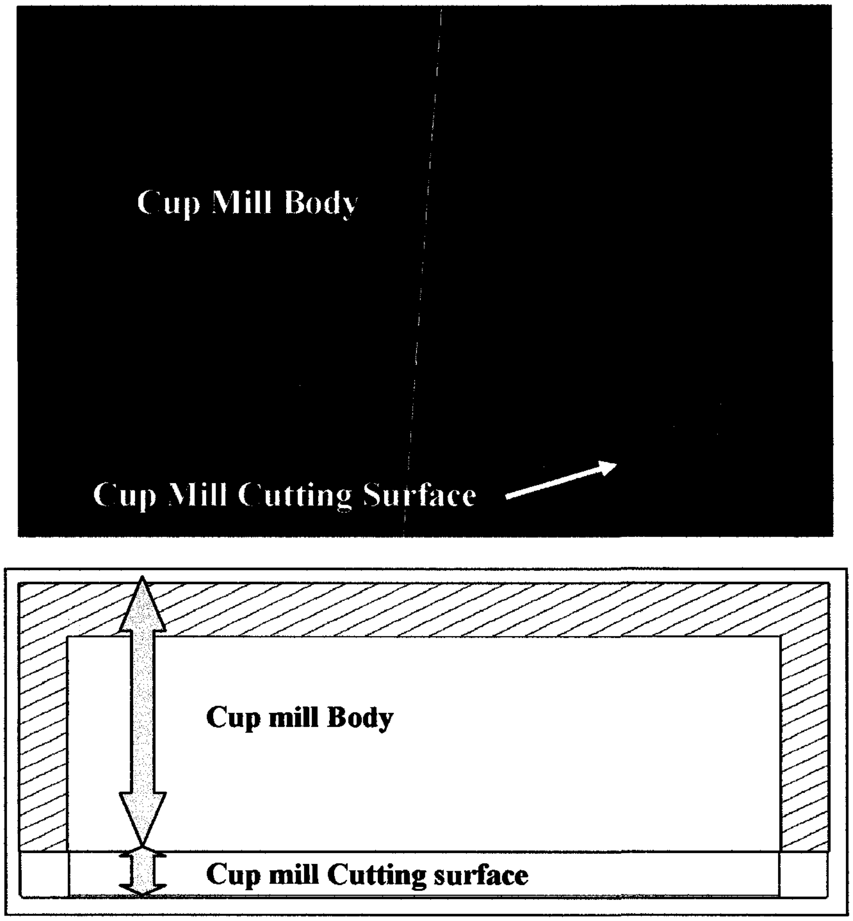


Fig. 5.5 Cutting and non-cutting surfaces of the cup mill

5.2.3.2 New kinematics strategy

The next step is to revise the kinematics of the tool.

In Figure 5.6 is presented the tool path generated by the first version of the algorithm. As can be seen, the back-and-forth movement of the tool between the suction and pressure side, at each level, generates a long tool path. However, as we will demonstrate, a more productive tool path can be found, that eliminates this back-and-forth motion and is, at the same time, collision free.

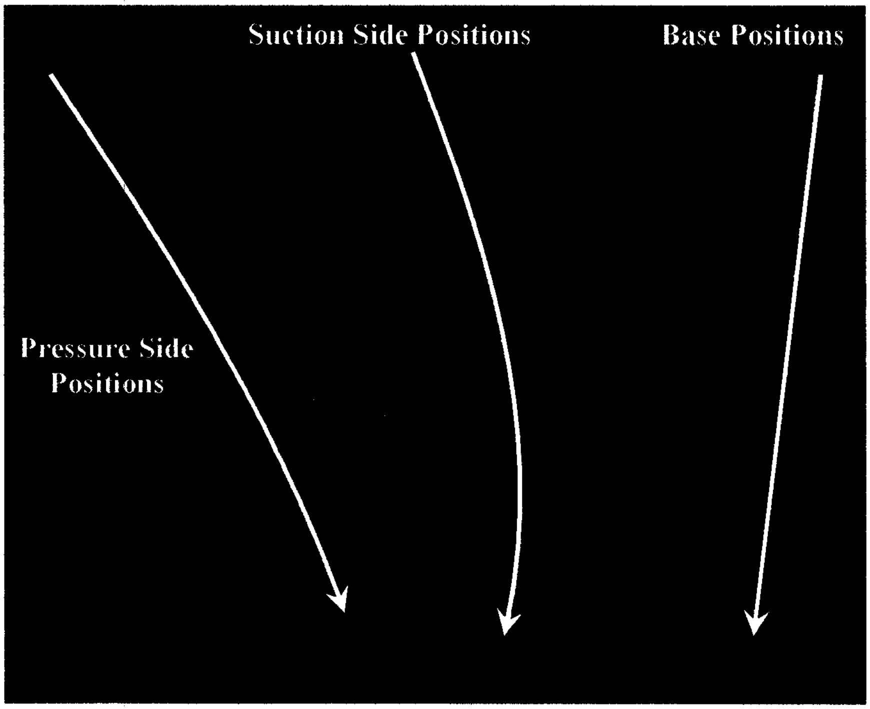


Fig. 5.6 Revised tool kinematics

The simplest and most direct way of improving the productivity of the machining process using a cup mill tool is to split up the tool path into three distinct tool motions. Referring again to Figure 5.6, the tool positions have been organized into 3 distinct groups (indicted by the arrows): pressure side, suction side and the base positions. If the cup mill passes by each one of these positions in separate tool motions, the length of the tool path is greatly reduced.

Instead of machining level-by-level both sides of the blade in a single pass, each side is machined separately in two different tool motions. Prior to this, the first tool operation is the open pass (having a fixed axis), which opens the space between the two blades (equivalent to the actual open pass). The surfaces of the blades are machined in a similar manner, by a plunging motion of the tool. Figure 5.7 presents the same tool path as in Figure 5.6, reorganized into the three distinct tool motions. The open pass is the first tool motion (1), and then comes either the pressure or the suction side machining (2 or 3). This new tool path is considerably shorter than the previous one.

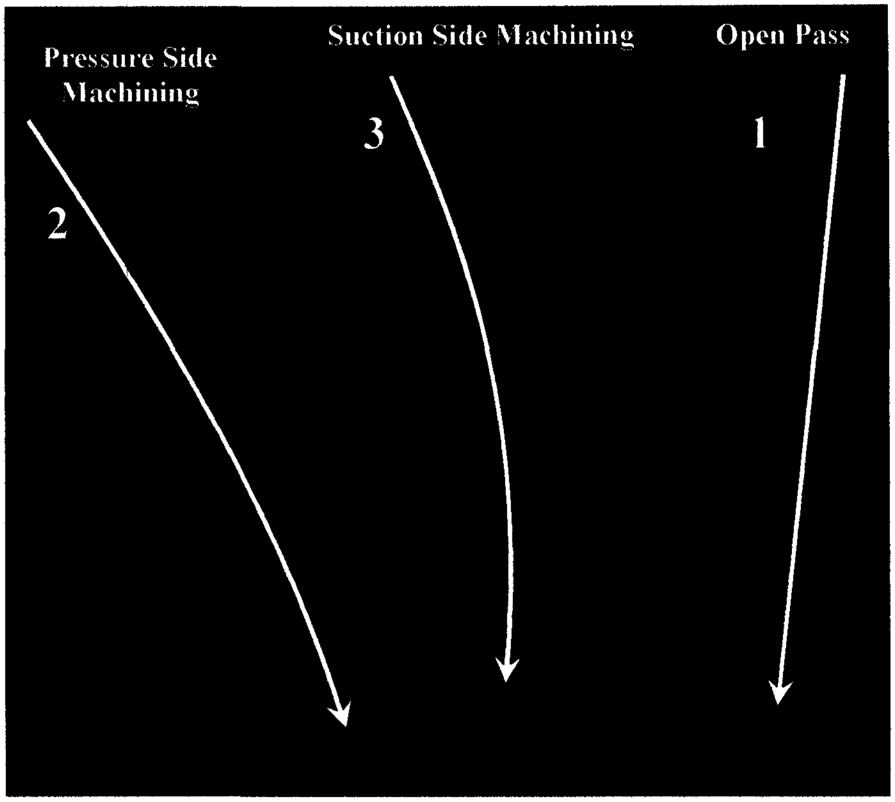


Fig. 5.7 New tool kinematics

Besides improving the efficiency of the machining process, the surface quality of the blade is also improved. Machining is performed on each side of the blade surface in a single continuous plunging motion, so it does not create any scallops on the blade surface like the previous tool motion (see Fig. 3.21 vs. Fig. 6.11).

In addition, the plunging motion of the tool reduces the vibrations and deflections of the blade. As pictured in Figure 5.8, the current ball endmill tools are machining with the flank of the tool and, thus, pushing on the blade, generating a deflection. On the other hand, when machining these surfaces with the cup mill in a plunging motion, mostly the bottom of the cup mill is engaged in the part (red portion). This motion of the tool generates a downward force that is not pushing on the blade. Also, there is always excess material present below the cup mill, which supports the blade at the cutting region, limiting even more the deflections of the blade.

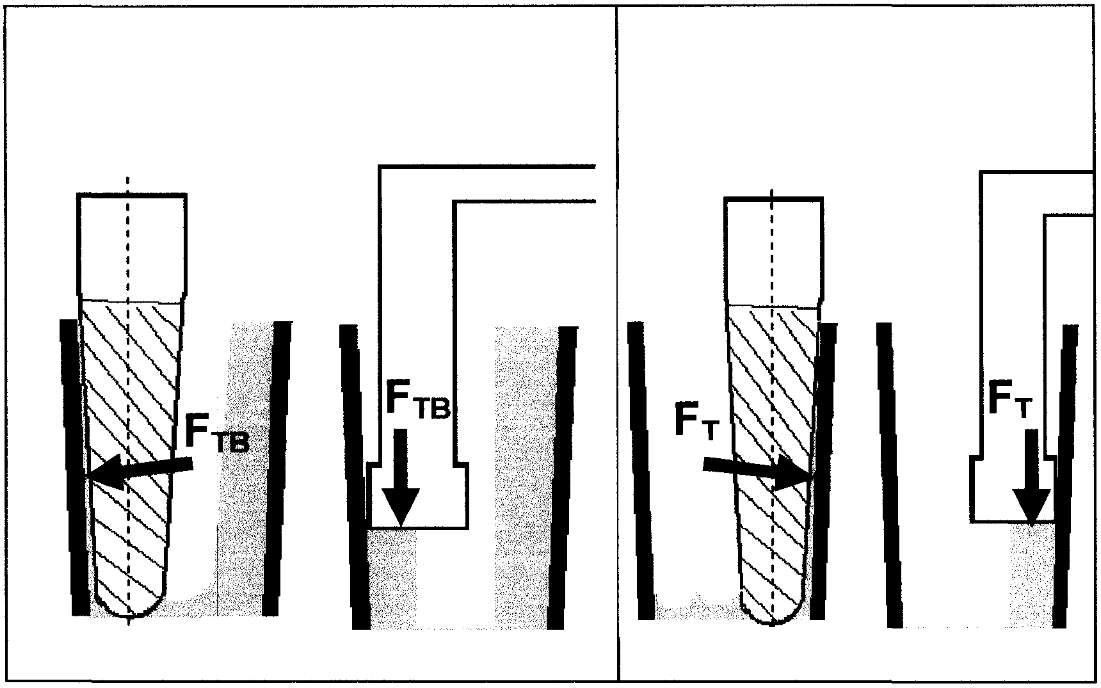


Fig. 5.8 Cutting forces present during flank milling and using a cup mill

5.2.4 Tool path optimization

The pressure side and suction side tool paths presented in Figure 5.7 are composed of tool positions at different levels of machining (the number of positions depends on the refinement of the computation). If at each tool position, the mismatches between the tool (cutting surface) and the blade are as small as possible, the resulting tool motion machines the blade as close as possible to the final surface. This concept is illustrated in Figure 5.4.

For that purpose, construction lines are created on the target surface. The general idea is to minimize the distances between these lines and the cutting surface of the tool. If these lines are spread across the whole surface, the mismatches between the surface and the tool are also minimized, see Figure 5.9.

These curves have to be evenly distributed on the whole surface of the blade. The number of curves is a compromise between the accuracy of solution and the computational time. In the example presented in the figure, five lines are created.

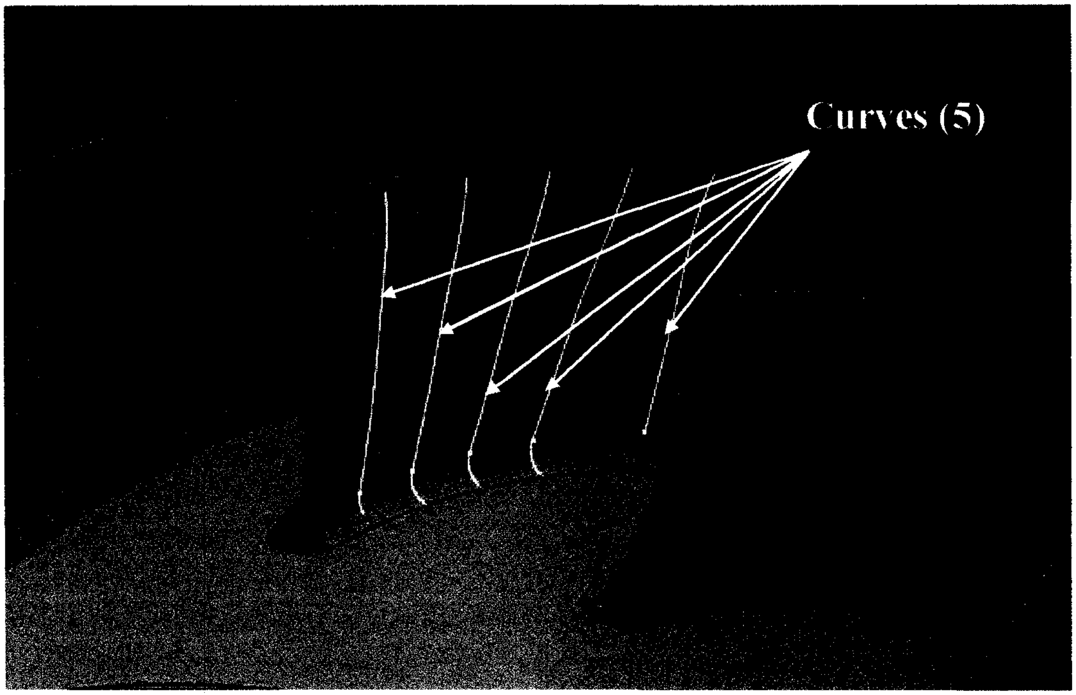


Fig. 5.9 Creation of the target curves

Referring to Figure 5.10, the distances between the cup mill (cutting surface) and the curves are noted by D_1 to D_5.

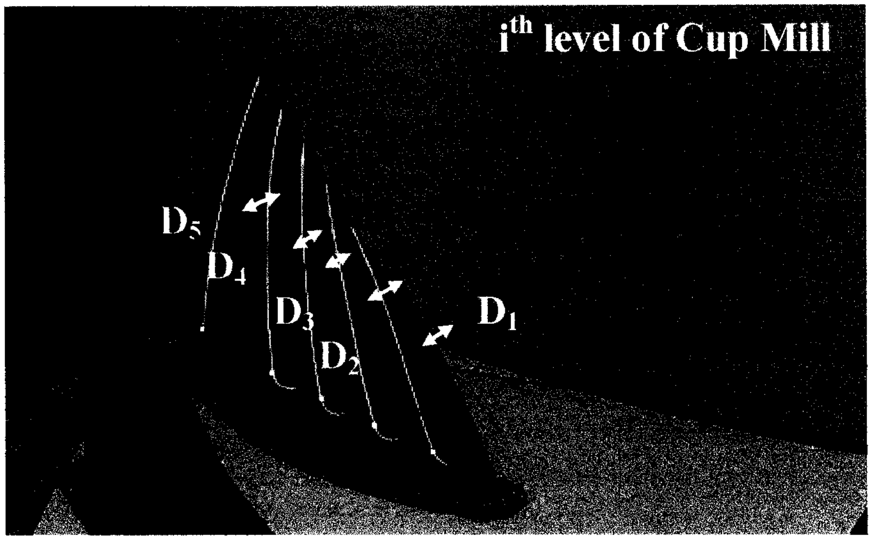


Fig. 5.10 Measure of the distances between the lines and the cutting surface

The minimization of the distances between the cup mill and the curves is performed by rotating the tool around the tangency point with the blade. This rotation about the tangency point is performed as illustrated in Figure 5.11.

In that scope, the first action of the program is to bring the tool in contact with the blade at a single point: bring out the cup mill if it is located in the blade or bring it toward the blade if the cup mill is away from the blade. After each movement (rotation) of the tool, an operation is performed with the purpose of repositioning the cup mill tangent to the blade surface, thus ready for the next rotation.

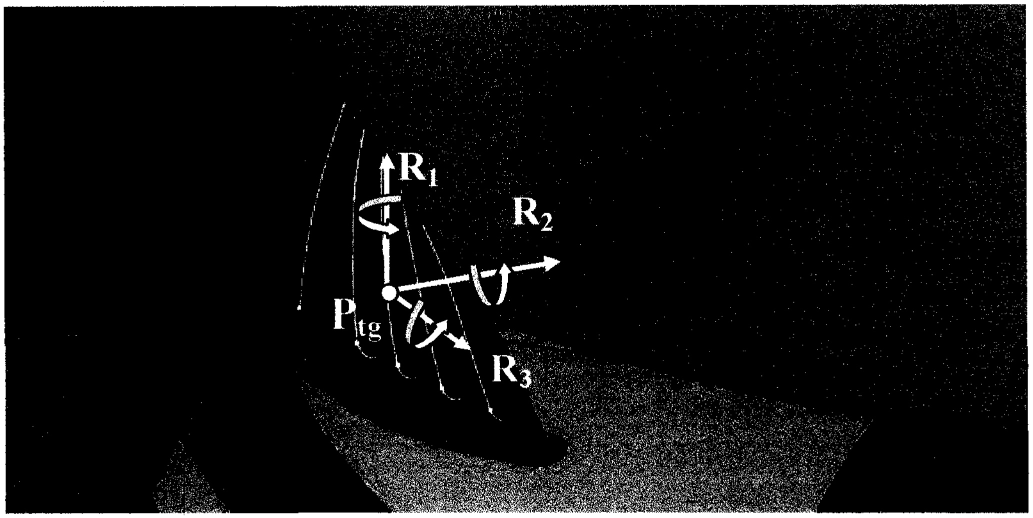


Fig. 5.11 Rotation of the cup mill about the tangency point

Once that the tangency point was identified, three possible axis of rotation are created about this point ($R_1 \perp R_2 \perp R3$, see Figure 5.11).

The optimal position of the cup mill on the blade is identified by successively rotating the cup mill about the tangency point, around one of the rotation axis, and then finding the new tangency point. The rotations are mutually independent to give the cup mill freedom of movement. This process is repeated until the distances to curves (D_1 to D_5) are the smallest possible.

When the optimization process stops (sum of distances are minimum), the cup mill is located at the first position of the tool path (position and orientation of the tool are kept in memory). From this position, the tool is translated upward (or downward, depending on the direction selected for the path generation) by a small increment. Then, the whole process is repeated for the new position (level) of the cup mill until the next solution is found. The process stops when all the positions of tool path are found which permits the machining of the whole surface.

The algorithm performs exactly the same operations on the opposing blade surface, generating the second tool path. The complete machining tool path is composed of three distinct tool motions:

- Open pass,
- Pressure side machining motion,
- Suction side machining motion.

5.2.5 Algorithm conclusions

Based on the tool positioning strategy and kinematics, the following sub-programs (controlled by the Main program) are required:

Tangency finding: After each movement (rotation), the tool is repositioned tangent to the blade (target surface) by this sub-program.

Rotation axes: Creates the rotation axes on the part about the tangency point.

Gradient evaluation: Determines the direction of strongest minimization of the objective function.

Collision prevention: Ensures that the tool positions are collisions-free.

Next section gives a detailed step-by-step description of the whole concept presented here.

5.3 Program

5.3.1 Optimum position (using Newton's method)

5.3.1.1 Function definition

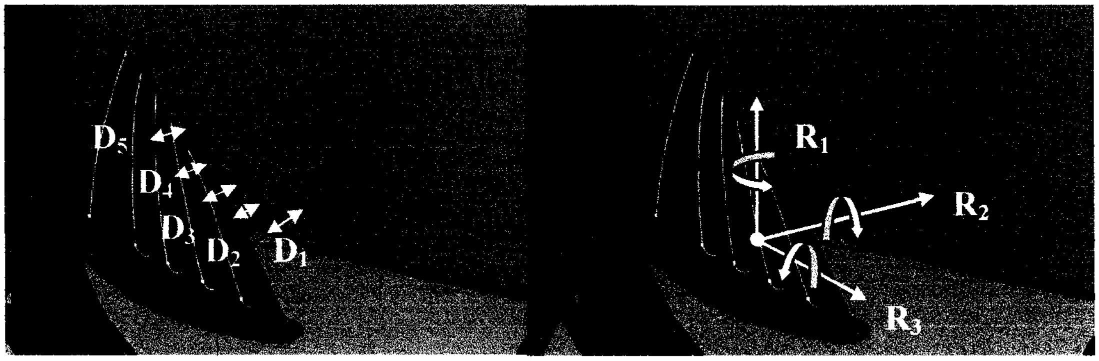


Fig. 5.12 Tool positioning strategy

As a reminder, the guide lines of the tool positioning strategy are summarized in the Figure 5.12, presented above. The optimum position of the cup mill is found by rotating the tool around the tangency point, until the mismatches with the blade are as small as possible. This problem can be transformed into a mathematical problem, to which an optimization algorithm is applied.

The first step to solve this problem is the creation of an appropriate mathematical model. Looking at the above picture, the distances D_1 to D_5 are in function of the rotation angles R_1 to R_3. Thus, this problem is described by the following function:

$$F(R_1, R_2, R_3) = \sum_{i=1}^{5} D_i \qquad (5.1)$$

Finding the minimum of this objective function is equivalent of finding the position of the cup mill that is as closely as possible to the target surface.

Mathematically speaking, optimization is the minimization or maximization of a function, subject to constraints on its variables. The following notation is used:

R$_i$ are the *variables*, also called *parameters.*
F is the *objective function*, a function of R$_i$ that is to be optimized.
C are the *constraints* that the variables must satisfy.

The optimization problem can then be written as:

$$\min_{x} f(x) = \min_{R_1, R_2, R_3} F(R_1, R_2, R_3) = \min \sum_{i=1}^{5} D_i \qquad (5.2)$$

The above formulation is an unconstrained optimization. The minimization of the objective function depends on real variables, with no restrictions at all on the values of these variables. This formulation is still incomplete, because the actual problem is constrained by physical limits on the positions of the cup mill. It is possible that the solution obtained by the unconstrained optimization presented above, gives a position for the cup mill which it is in collision with the part. Boundaries have to be set to the possible cup mill position (solution to the problem) as to prevent collisions with the part. This problem can be reformulated in the following manner:

$$\min_{R_1,R_2,R_3} F(R_1,R_2,R_3) = \min \sum_{i=1}^{5} D_i$$

$$s.t.$$

$$(R_1^*, R_2^*, R_3^*) \in feasible\ region \qquad (5.3)$$

The feasible region is the set of possible cup mill positions satisfying all the constraints (limits of the cup mill positions), and the optimal point (R_1^*, R_2^*, R_3^*) is the solution of the problem. In other words, the optimal position found for the cup mill is not the *absolute* minimum solution, but rather the “collision free” optimal solution. The only case where this collision free optimal solution is also the absolute solution is when the absolute minimum is inside the feasible region.

It is hard to identify what are the boundaries of the feasible region. Even if this region is impossible to be defined, it does however exist. The method used to ensure that the optimum solution found is included in the feasible region will be discussed later in this chapter.

Once the model has been formulated, an optimization algorithm can be used to find its solution. There are numerous different algorithms possible, each of which is tailored to a particular type of optimization. In this case, however, due to the total lack of information concerning the function and the constraints, the possible algorithm choices are limited.

5.3.1.2 Optimization problem

The algorithm used to find the solution of the defined problem is the Newton method presented in section 3.4.1.The gradient and the Hessian of the function cannot be exactly determined in this case since the function is unknown (we know only the values of the functions at each point); so the method cannot be directly used. To overpass this problem, approximation of them will be used:

$$\nabla f_k = \frac{f(x_k + \Delta) - f(x_k)}{\Delta} \text{, for a small } \Delta \qquad (5.4)$$

A good approximation for the Hessian matrix is obtained by simply applying the Sherman-Morrison-Woodbury formula (see section 3.4.2):

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad \text{(BFGS)} \qquad (5.5)$$

Where,

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f_{k+1} - \nabla f_k$$

Since the objective function is unknown, the values of the function have to be evaluated at every single step of the computation. This complicates the approximation of the Hessian matrix. For simplicity reasons, the Identity matrix replaces the Hessian matrix at all the iteration steps. This can be explained by the absence of information concerning the function: first, the evaluation of the function is not exact, since the values of the distances are truncated (by CATIA) and, secondly, these values are used to calculate the approximated gradient vector, amplifying even more the error. This is more apparent as the values of the function are getting smaller and smaller, the error becomes more dominant. As a result, the approximation of the Hessian is so poor that the identity matrix is just as good as an approximation.

In conclusion, in our case:

$$B_k = I, \ \forall\, k \tag{5.6}$$

Considering the changes made, our optimization algorithm becomes:

```
Given: Starting point $x_0$ , tolerance $\varepsilon>0$
$k \leftarrow 0$;

While $\|\nabla f_k\| > \varepsilon$

    $d_k = -I\ \nabla f_k = -\nabla f_k = -\left[\frac{f(x_k+\Delta)-f(x_k)}{\Delta}\right]$

    $x_{k+1} = x_k + \alpha_k d_k$

    $k \leftarrow k+1$

end (while)
```

Algorithm 5.1 Positions optimization algorithm

Note: α_k is obtained by using the line search algorithm presented in section 3.4.3 (Wolf condition).

5.3.2 Initial guess (starting position)

To start the computation of the algorithm, an initial position of the tool is required. This initial position is called the initial guess and corresponds to the “x_0” in the optimization algorithm. It is very important that this initial position is relatively close to the final solution. If this initial position is not carefully chosen, then the algorithm might converge to a local minimum of the objective function, or it might even fail to converge. The best way to explain the need for an initial guess close to the final solution is by referring to the example presented in Figure 5.13. If the initial condition is selected far from the absolute minimum, chances are that a local minimum exists between the solution and the initial guess. The algorithm will converge to that local minimum instead of the true minimum (global minimum).

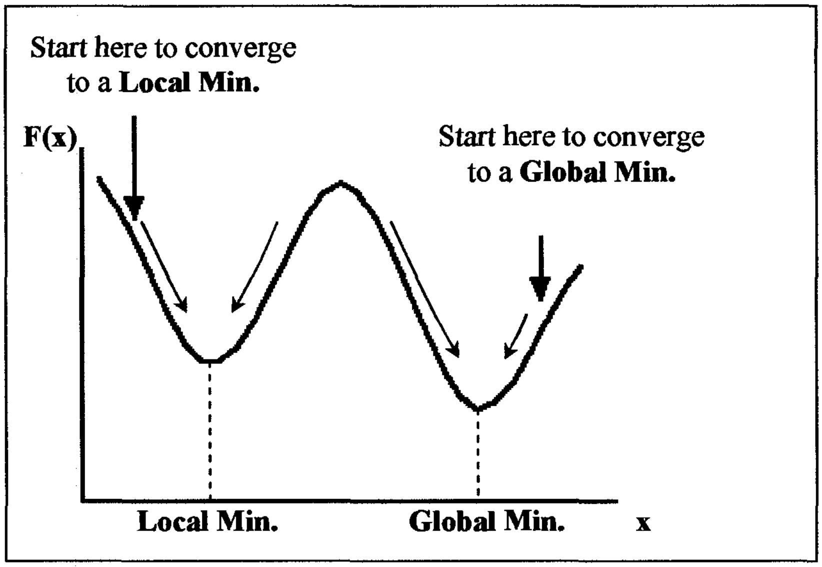


Fig 5.13 Convergence to a local minimum vs. convergence to a global minimum

The initial position of the cup mill is performed in two steps. First, an initial axis orientation is determined. Then, having this orientation, the tool is positioned on the blade using a strategy that tries to constrain the eventual deviations right from the start.

5.3.2.1 Initial orientation

Having a good cup mill orientation to start with is very important. It was observed that the convergence to the solution depends more on a good initial cup mill orientation than on a good initial position of the tool. However, no positioning strategy of the tool could be found that guarantees the convergence of the algorithm to the solution. The construction described here for determining a correct initial tool orientation was developed by a trial-and-error methodology. This construction was tested for several different part number IBRs, and in each case, the correct solution was found (results are presented in the next chapter).

For a given IBR part number (presented in Figure 5.14), a first line is created, identified by LTop. This line goes from the top corner of the leading edge of the pressure surface to the top corner of the trailing edge on the suction surface. A second line, LBottom in the figure, is created in a similar manner at the bottom of the blades. The line passing by the midpoints of "LTop" and "LBottom" is our initial cup mill orientation ("Direction _line" in the figure).

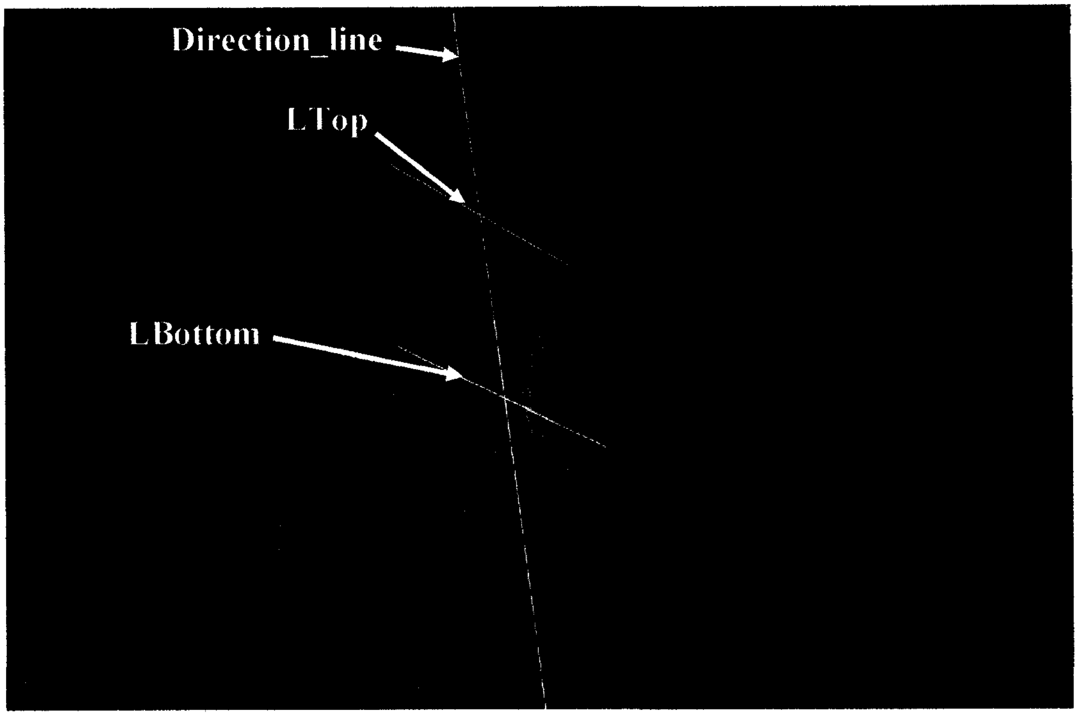


Fig. 5.14 Initial orientation

This simple construction for determining the initial cup mill orientation permits a convergence to the solution for all the part numbers on which the program was tested. Before coming up with this strategy, other different variations to this one where tried. None gave as good results for all the part numbers, so this construction was selected.

5.3.2.2 Initial position

Using this orientation, the tool is positioned on the blade to start the optimization. However, the convergence of the algorithm does not depend as much on the initial position (x, y, z), but it is ensured by a good cup mill initial orientation. On the other hand, a good cup mill initial position on the blade reduces the number of iteration steps required to reach the solution. The method described here places the cup mill relatively close to the blade, right from the start (for the defined orientation), but it is not based on any mathematical principles and other methods could give good results just as well.

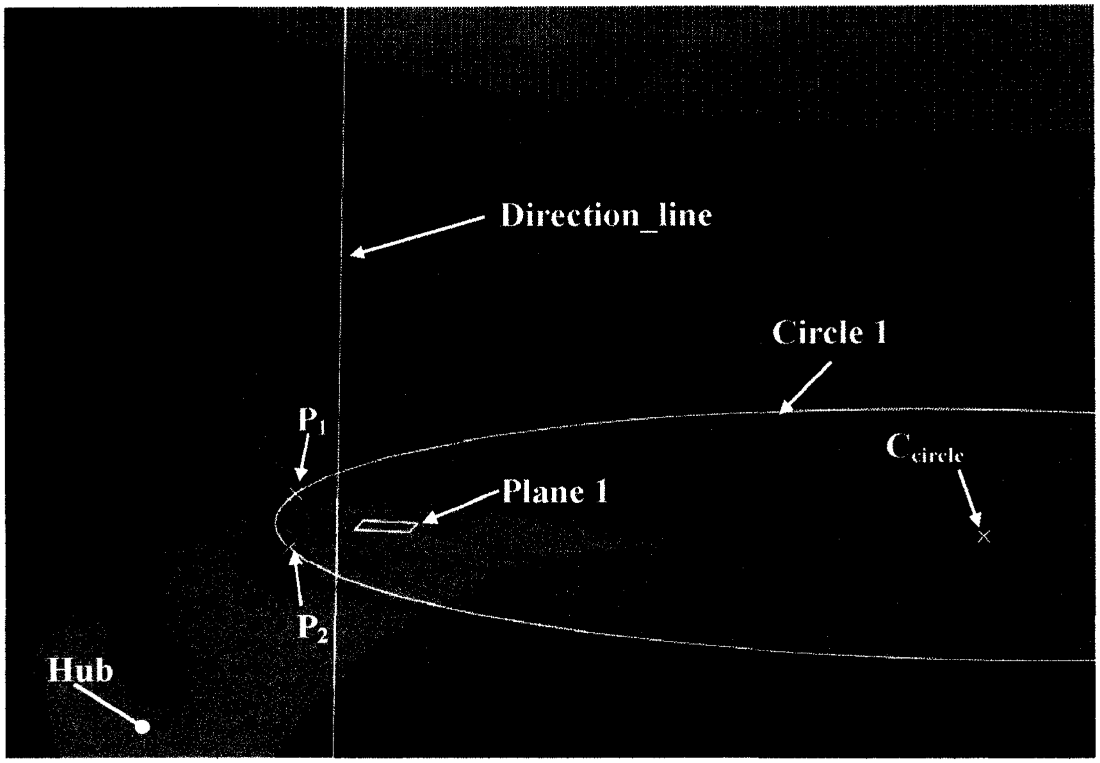


Fig 5.15 Initial Position

The tool positions are generated from the bottom of the blade and moving upward. Consequently, we are required to identify the tool level that is the closest to the hub from which starts the computation. At this level, a plane is created that is perpendicular to the "Direction_line", identified by Planel in the Figure 5.15. This plane is then intersected with the target blade surface (offset pressure side in this case). The end points of intersection are identified as P_1 and P_2.

A circle is then created, positioned on this plane and passing by the two points, P_1 and P_2 (Circle 1). The diameter of the circle is the average between the outside and inside diameter of the cup mill, i.e.:

$$Rad = \frac{OD + ID}{2}$$

The center of the cup mill is positioned on the center of this circle (C_{circle} in the Figure 5.15) and resting on Plane 1. This is the initial position of the tool.

5.3.3 Optimization program

5.3.3.1 Tangency repositioning subprogram

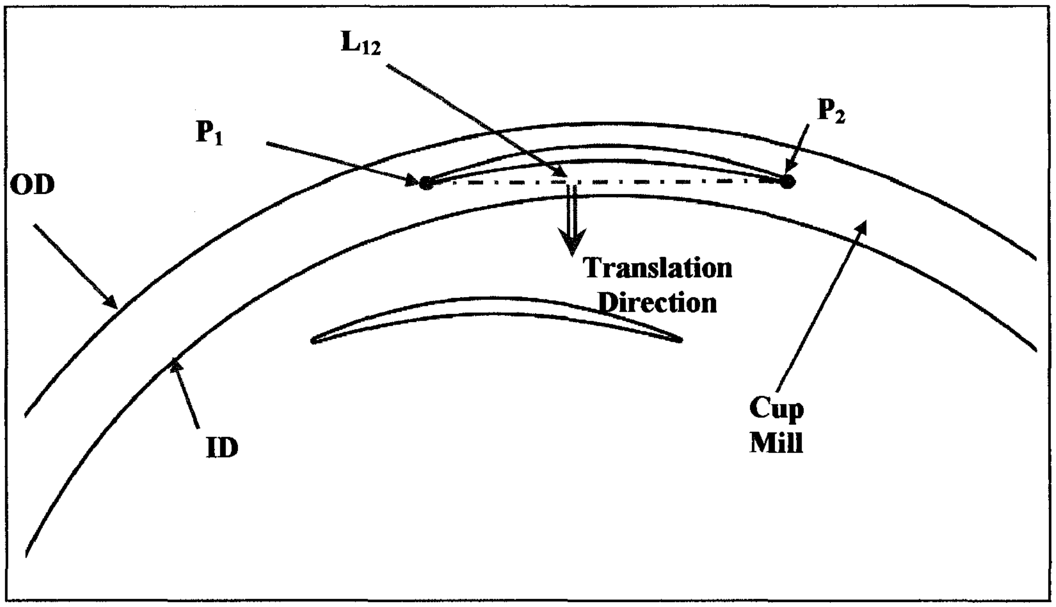


Fig. 5.16 Initial position of the cup mill (inside the blade)

At this position, the tool is inside of the blade as portrayed in Figure 5.15 and Figure 5.16. The cup mill must be translated outward until it is tangent to the blade surface. This operation is performed by a subprogram (called by the main algorithm) which is called the "Translate" subprogram. This program was necessary since no suitable CATIA function was found for this purpose.

Given a *target* surface (the surface that it is desired to bring the cup mill tangent to, i.e. suction, pressure or hub surfaces) and a direction of translation, the subprogram will bring the tool tangent to the target surface, whenever required by the main program. We present below the method used.

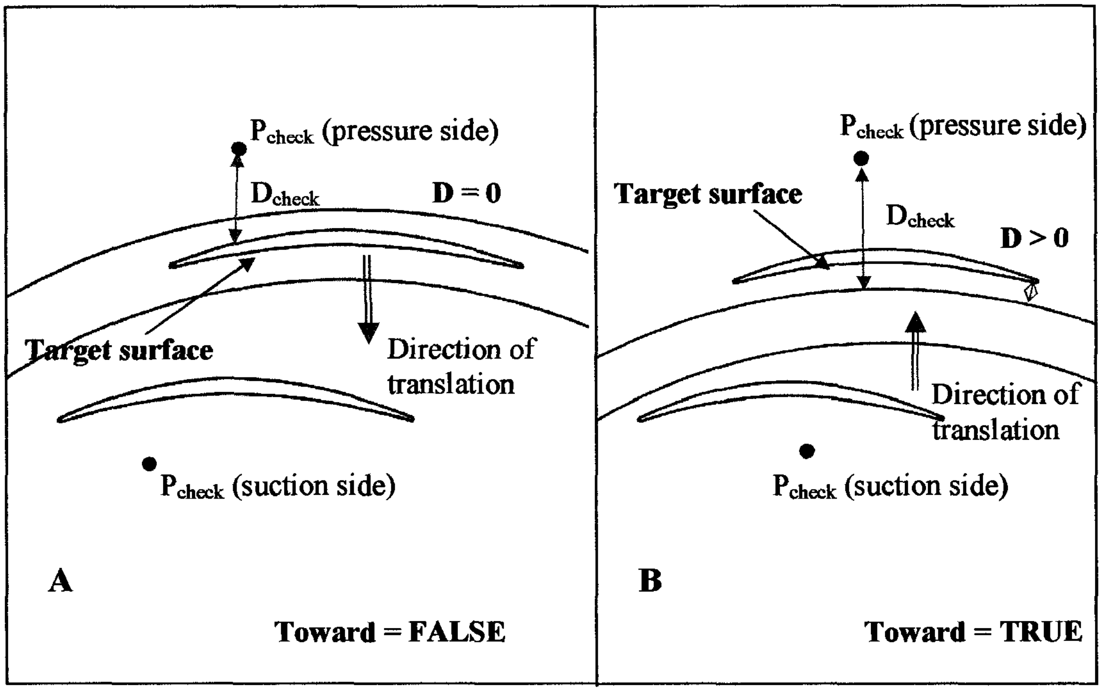


Fig. 5.17 Determining the sense of translation

The "Translate" subprogram requires three inputs from the main program: the target surface, the line of translation and the correct direction of translation along that line. The surface is simply given by the main program. The line of translation is determined by creating a simple construction: referring to Figure 5.16, this line is on the plane "Plane 1" and is perpendicular to the line passing through the points P_1 and P_2.

Finding the correct direction of translation along this line is more difficult. From Figure 5.17, two situations are possible: (a) the cup mill is inside of the blade or (b) the cup mill is at a certain distance from the blade. These two possible situations impose opposing directions of translation. Moreover, when the tool is inside the blade (Fig. 5.17 A), translating the tool upward or downward, both bring the tool tangent to the target surface (with the inner or outer circumferences). The program cannot differentiate between these situations, but only the "down" direction is desired. To complicate the problem even more, the direction is reversed if the target surface is the suction side instead of the pressure side.

Before starting the translation, the subprogram performs some tests to clarify the situation of the cup mill. Based on the results, the correct orientation is determined.

The first test performed determines if the tool is intersecting the target surface. For that purpose, a Boolean variable (true or false) named "Toward" is assigned. Referring to Figure 5.17, when the tool is not intersecting the blade (case B), the distance between the tool and the blade "D" is non-zero positive. Here, the tool has to be translated toward the blade, so the variable "Toward" becomes TRUE. In the case of intersection with the surface, "D" is zero and the variable "Toward" is FALSE. This variable clarifies the situation of the cup mill, but it is not sufficient to define the direction of translation.

The translation direction is determined using a *reference* point, P_{check}. This point is always positioned *outside* the space between the blades, sufficiently distant from the target surface as to prevent any possible contacts with the cup mill (~ 1 inch), as shown in Figure 5.17. A similar point is placed with respect to the opposing blade surface.

For the situation when the tool is positioned outside of the blade, a decreasing value of "D" (distance between the tool and the blade) indicated that the direction of translation is correct, if "D" is increasing then the direction is reversed. On the other hand, when the tool is inside of the blade, "D" is zero, and for a small translation of the tool in both directions "D" remains zero. In this case, the reference point P_{check} is very useful: if the distance D_{check}, between the tool and the point P_{check}, is increasing, the direction is correct, otherwise it is reversed. The tool is translated outside from the intersection using small increments until "D" becomes non-zero (positive). Then, the direction of translation is reversed, "Toward" becomes TRUE, and translations of step-size of value "D" are used until the cup mill is tangent to the blade (within a tolerance).

The next flowchart presents the "Translation" sub-program.

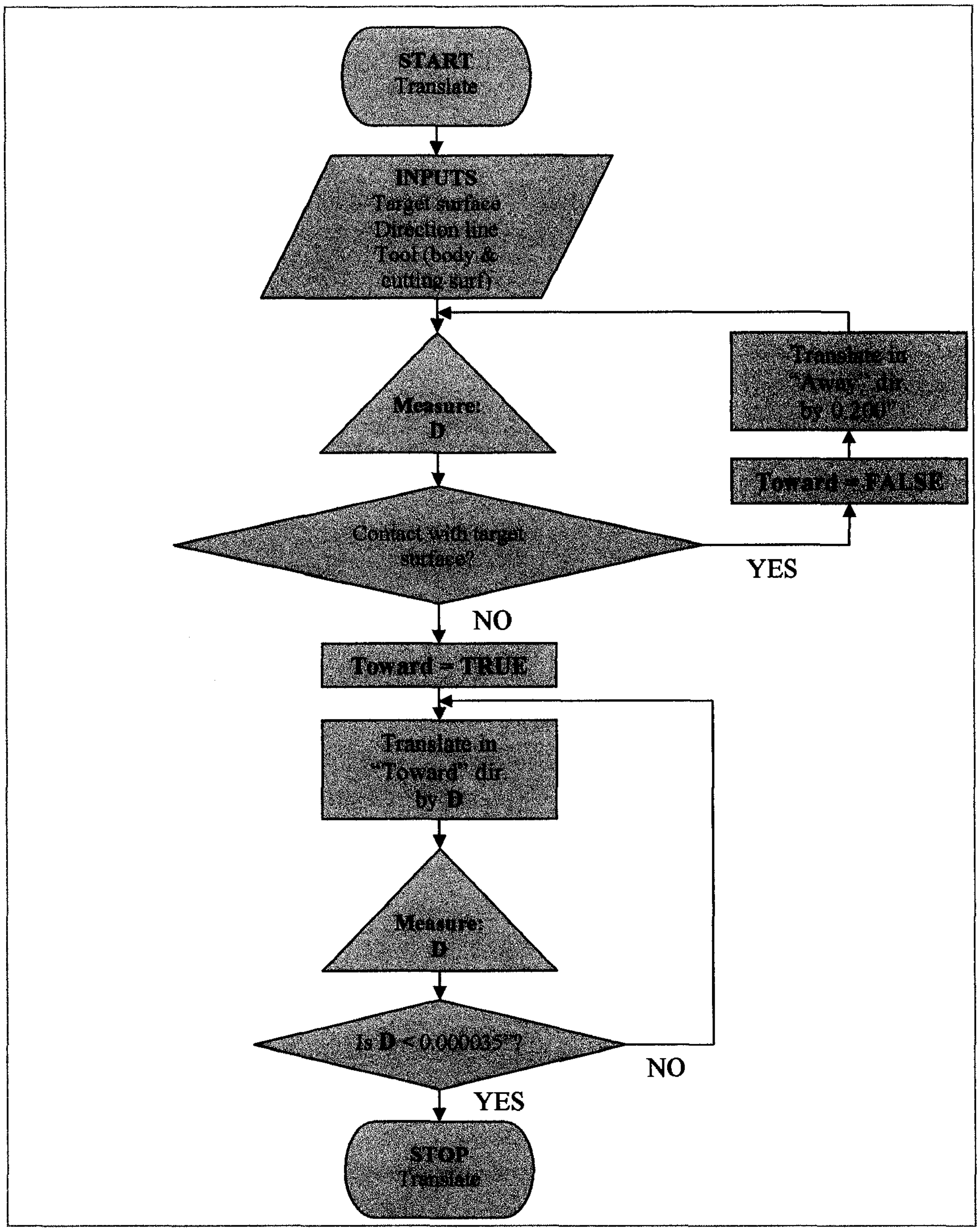


Fig. 5.18 Flowchart of the Translation sub-program

Referring to the flowchart presented in Figure 5.18, the Translation subprogram is composed by two distinct phases (loops). The first phase determines if the cup mill is intersecting the target surface. In the case of intersection, the program translates along the direction line (input) the cup mill in the "away" direction by steps of 0.200 inches. This operation is repeated until the cup mill is no more intersecting the target surface and passes to the next phase. At the point, the direction of translation is reversed "toward" the blade and, from here on, no more intersections with the target surface are possible. To ensure this, the fixes step (translation) length used in the previous phase is replaced by a variable step length. The minimum length between the target surface and the tool is measured (D) and the tool is translated "toward" the surface by that distance. This operation is repeated until the distance between the tool and the target surface is less than a desired tolerance.

As mentioned, once the tool is moved outside of blade, no further intersections with the blade are possible. By referring to Figure 5.19, a body B is translated toward body A by a translation length equal to the minimum distance between the two. If the direction of translation is parallel to the minimum distance vector, the two bodies will become tangent to each others. A translation in any other direction will result in a gap between the two bodies (impossible to have a collision).

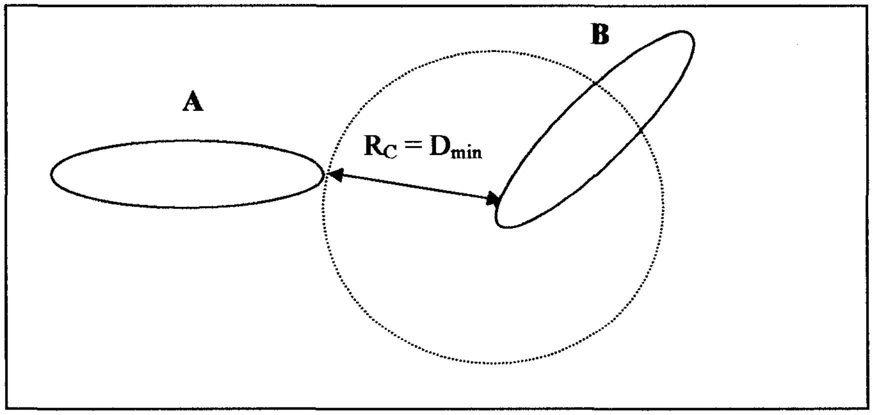


Fig. 5.19 Translation of variable length "D_{min}"

In our case, the tool is translated by a step-size of variable value "D", which is the minimum distance between the tool and the blade. Since this translation is performed along a predefined line, not corresponding to the minimum distance vector, the tool is translated closer to the blade but will never intersect it. The process stops when the value of "D" becomes very small (typically <0.001").

Note that for simplicity, the flowchart refers to "away" and "toward" directions. How these directions are identified was described previously, but is not mentioned in the flowchart.

5.3.3.2 Rotation axis subprogram

At this point the cup mill is tangent to the target surface and the rotation axis about the tangency point around which the tool is rotating need to be created. This operation is performed by another subprogram called "Rotation axis" subprogram. Whenever required by the Main program, this simple subprogram creates the three axis about which the tool rotates.

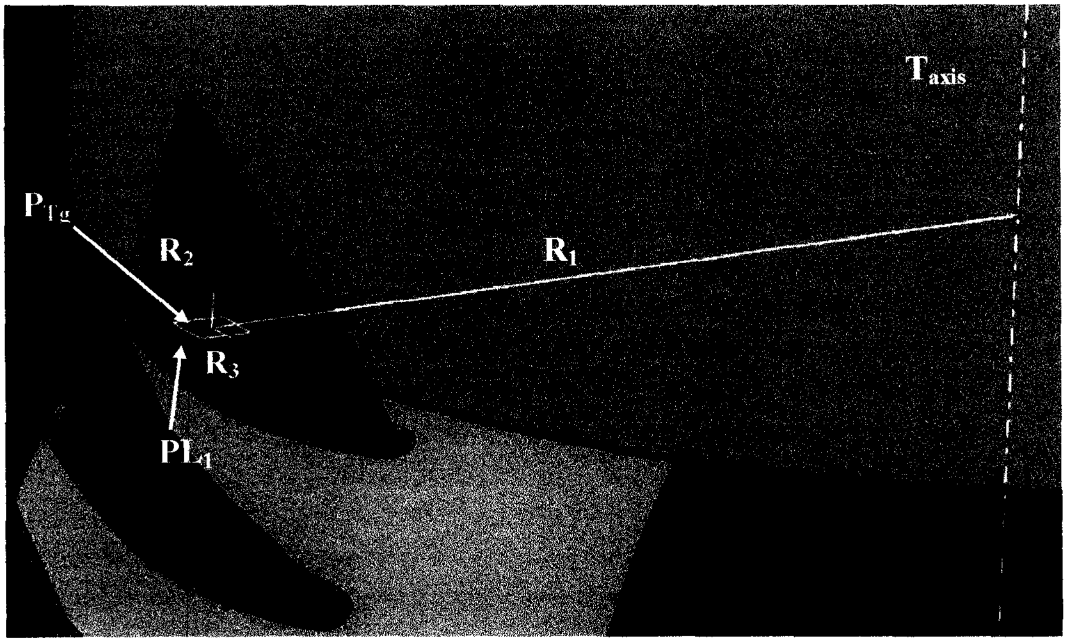


Fig. 5.20 Creation of the rotation axis

Looking at Figure 5.20, the cup mill is placed tangent to the blade at the previous operation, by using the Tangency sub-program; obtaining the tangency point P_{Tg},. The axis of the tool in the current position (T_{axis}) is created. The line perpendicular to the tool axis (T_{axis}) and passing by the tangency point P_{Tg} is the first rotation axis, R_1.

On the tangency point, a plane is created, PL_1, which is normal to T_{axis}. The second line of rotation is normal to the plane PL_1 and passes by the tangency point (note:

R_2 // T_{axis}). The third line of rotation R_3 is simply perpendicular to both R_1 and R_2. As a result, all three axis of rotation are independent from each other (i.e. $R_1 \perp R_2 \perp R_3$) passing by the tangency point, as shown in Figure 5.20.

5.3.3.3 Gradient evaluation subprogram

Referring to the general algorithm of the program presented in Algorithm 5.1, the gradient of the function ∇f_k need to be evaluated at each step of the iteration. As a reminder, the gradient evaluation (approximation) formula is presented below:

$$\nabla f_k = \left[\frac{f(x_k + \Delta) - f(x_k)}{\Delta} \right]$$

Once the tool is tangent to the blade, the strategy is to determine which of the three possible rotations (R_1, R_2 or R_3) gives the strongest minimization of the function. The minimization potential of each rotation is indicated by the value of the module of their gradient vector: the bigger the module of the gradient, the stronger the minimization in that direction.

By keeping the other two angles constant, the algorithm evaluates the gradient module of each possible rotation. Then, the one giving the strongest minimization in the function is selected and retuned to the Main program. The minimization at this step will be only performed about the selected rotation axis. For the next step, the minimization potential of each rotation axis is reevaluated again and the strongest one is selected.

The flowchart of the gradient subprogram is presented in Figure 5.21:

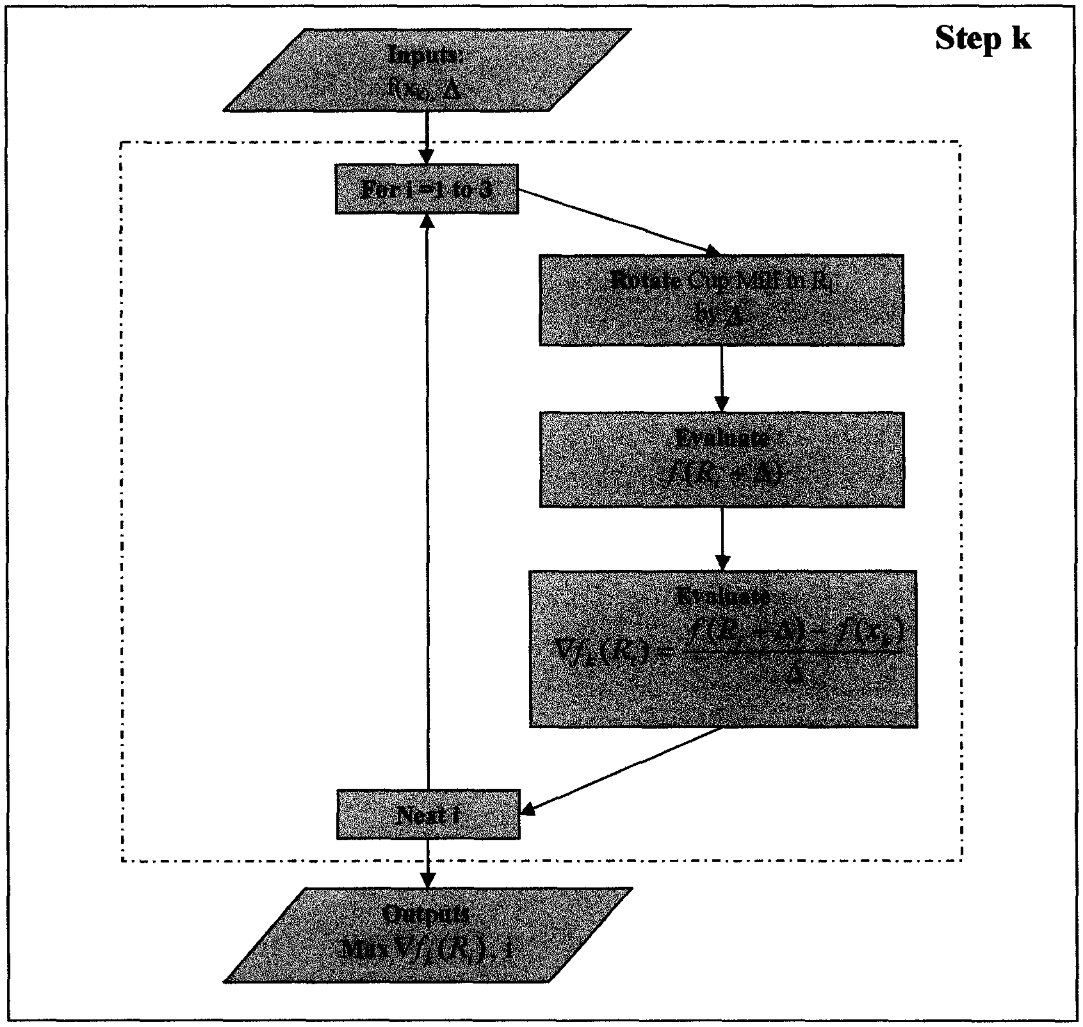


Fig. 5.21 Flowchart of the Gradient sub-program

5.3.3.4 Optimization Main program

Following our modified Newton's method, presented in section 5.3.1.2 (Algorithm 5.1), the main program minimizes the deviations between the target surface and the tool (cutting surface), by rotating the cup mill about the rotation axis found by the gradient subprogram (giving the strongest minimization). In this case, an appropriate step size is found (by the Armijo condition) that sufficiently minimizes the function in that direction.

Then, having decreased the function sufficiently in that direction (rotation) the process is repeated and a new rotation axis is found. The process is stopped when the optimum position is reached, i.e. when none of the three rotations sufficiently decreases the objective function. At this point, the first solution (position) of the tool path is found.

The flowchart of the Main program is presented in Figure 5.22:

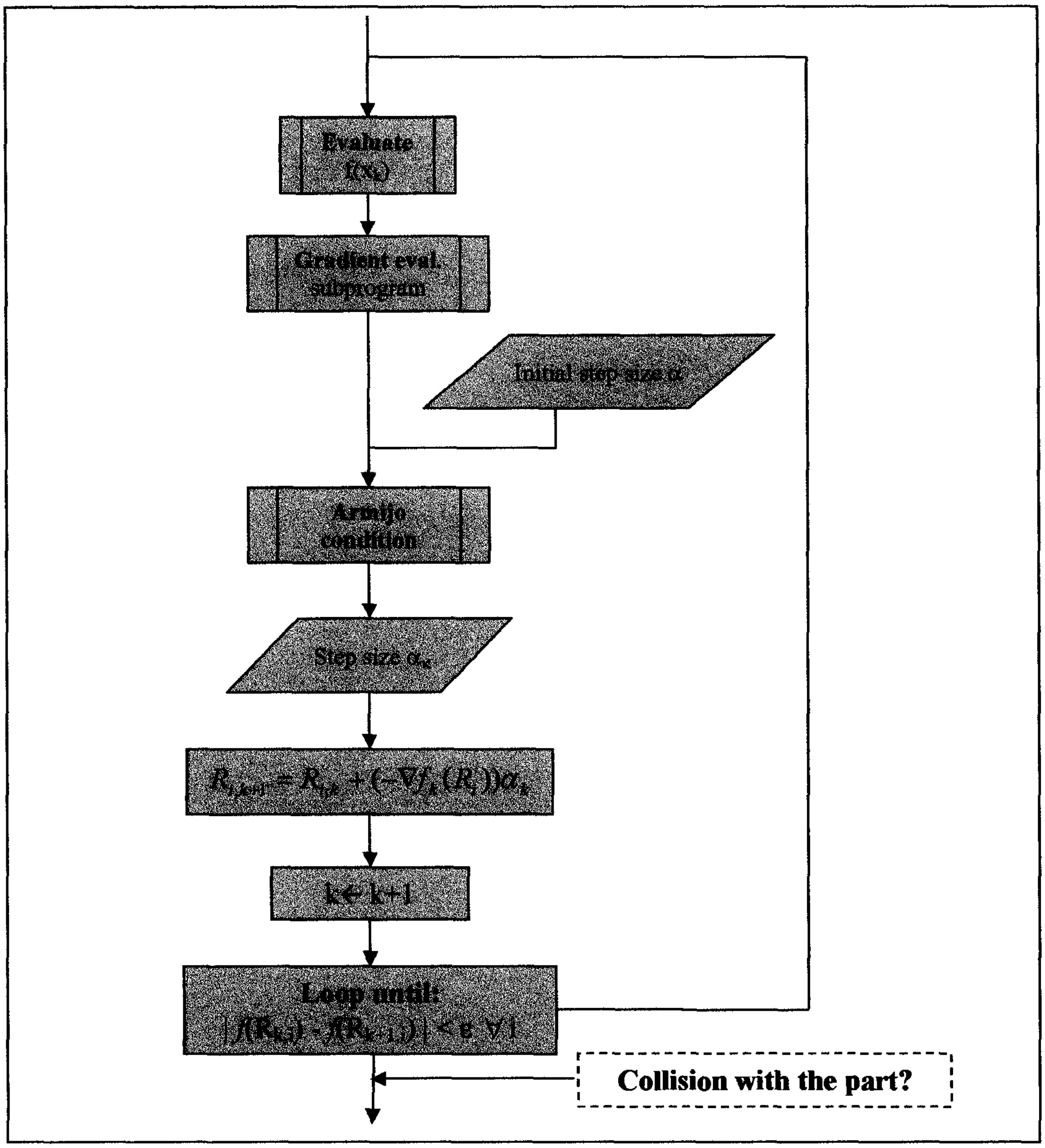

Evaluate f(x_k)
Gradient eval. subprogram
Initial step size α
Armijo condition
Step size α_k
$R_{i,k+1} = R_{i,k} + (-\nabla f_k(R_i))\alpha_k$
k← k+1
Loop until:
$|f(R_{k,i}) - f(R_{k+1,i})| < \varepsilon \ \forall i$
Collision with the part?

Fig. 5.22 Flowchart of the Main program – without collision evaluation

Note: after each rotation of the cup mill about one of the rotation axes, the Translate subprogram is used to bring the tool tangent to the surface, prior to measure the value of $f(x_k)$.

However, there is no guaranty that, at this position, the cup mill is free from any collisions with the part. Another important sub-program is required to ensure that the solution is collision free.

5.3.3.5 Collision evaluation subprogram

The solution obtained by the algorithm must be inside the feasible region. In other words, the final position of the cup mill must be collision free with the part. In this constrained optimization problem, the Simplex or other methods for solving constrained optimization problems are inappropriate, since neither the function nor the constraint equations are known.

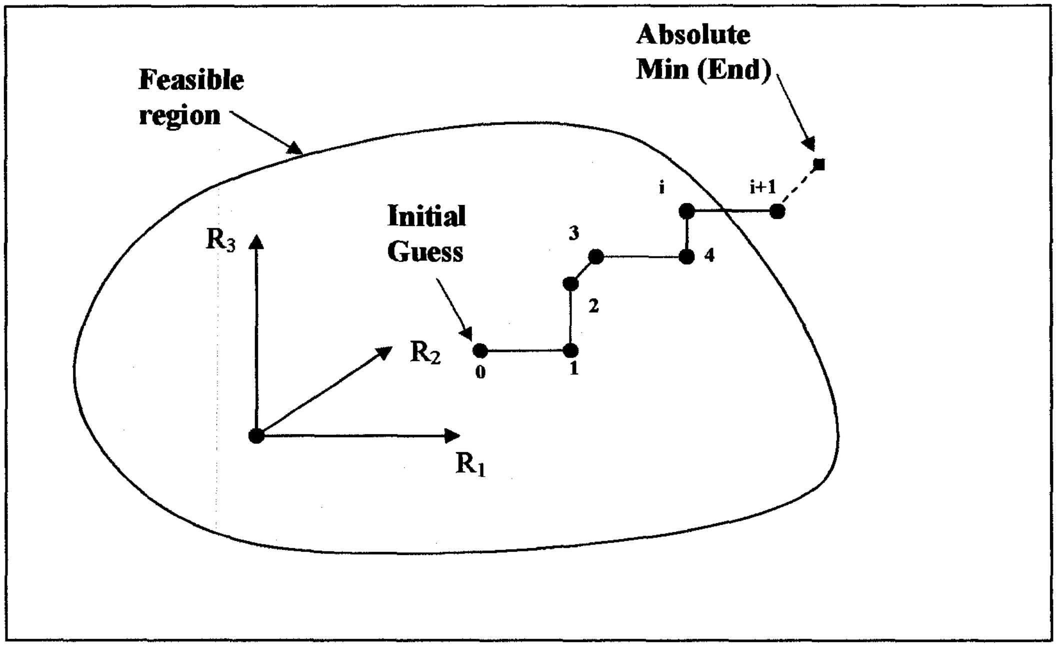


Fig. 5.23 Final solution outside the feasible region

Referring to Figure 5.23, starting from an initial position (0) the algorithm, as previously presented, would converge to the absolute minimum without taking in consideration the feasible region. Chances are that the solution found is outside that region and the cup mill, in that position, would intersect the part.

On the figure, the position of the cup mill crosses the boundary of the feasible region between the i^{th} iteration and the $(i+1)^{th}$ iteration. The constrained solution to the problem is located on the boundary of the feasible region and in the vicinity of the i^{th} position. One possible solution to prevent collisions with the part is to stop the computation once the boundary is crossed (at the i+1 position) and keep the cup mill at the i^{th} position (consider this position to be a close approximation of the optimum position).

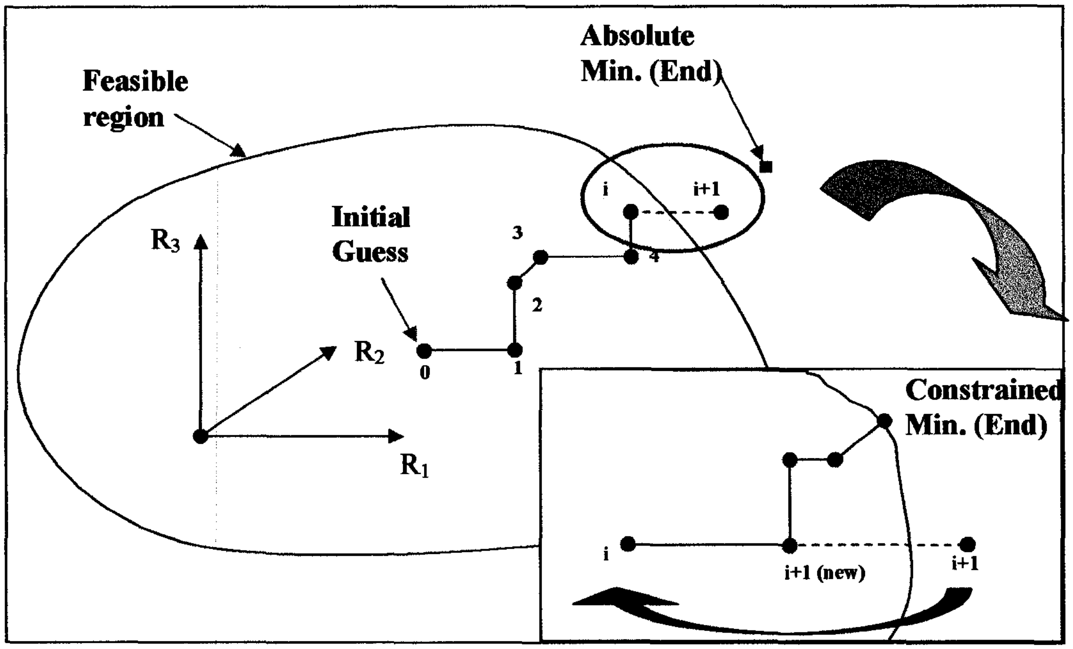


Fig. 5.24 Final solution inside the feasible region

Another solution, which is more precise, is to never allow the cup mill position to cross the boundary surface, as presented in Figure 5.24. At each iteration, the cup mill is verified for intersections and its movement is cancelled if a collision is present. In this case, the step size α_i is reduced by half and the iteration is resumed in the same direction but with the new step size. The step size is adjusted in this manner, until the constrained solution is reached. By never allowing the cup mill to intersect the part, the solutions obtained are collision-free positions.

To eliminate the risk of engaging into an infinite loop, the step size is prevented from being reduced more that a minimum value ε_α, in which case, the iteration is stopped for that direction.

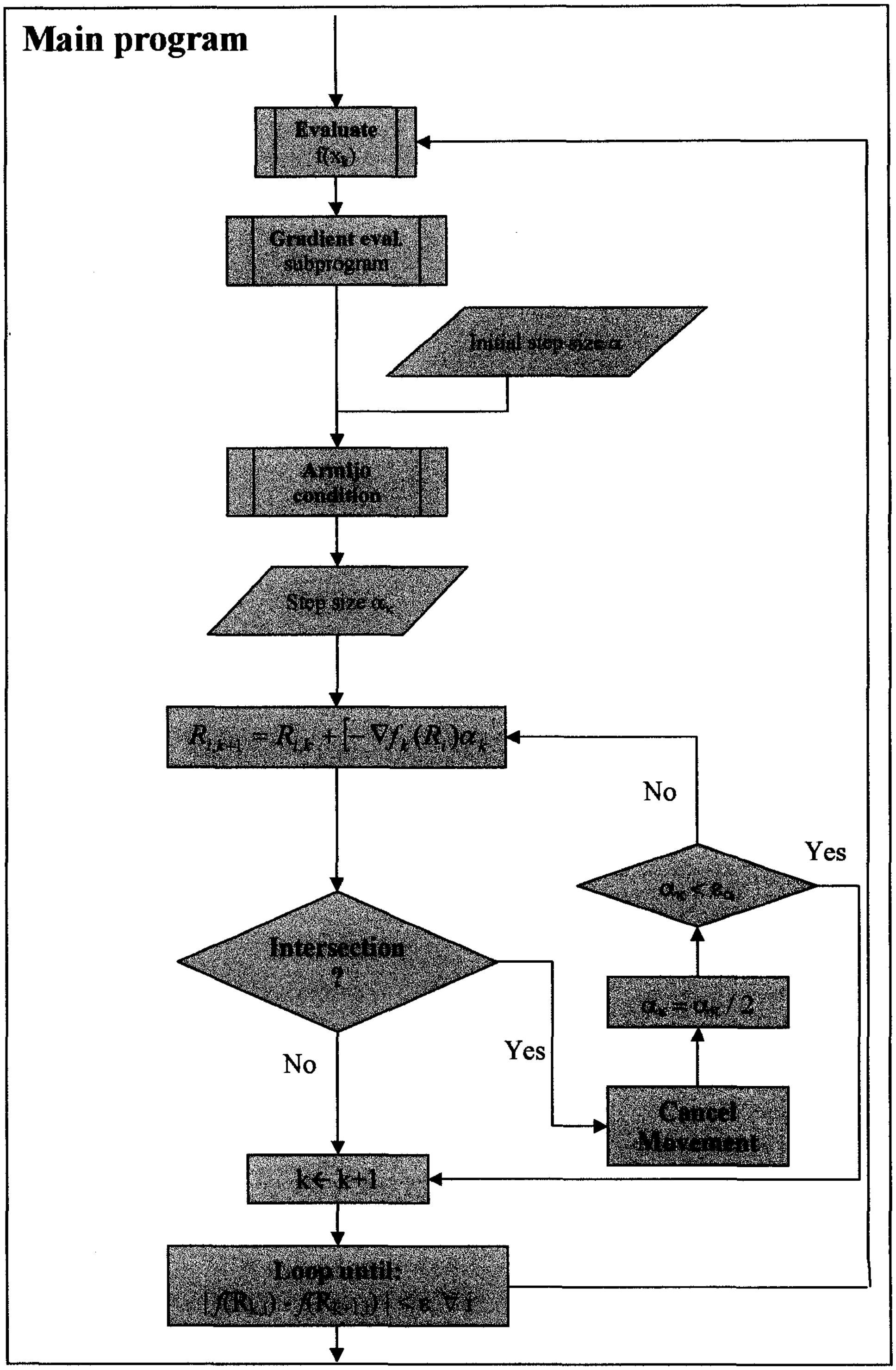

Main program
Evaluate
f(x_k)
Gradient eval.
subprogram
Initial step size α
Armijo
condition
Step size α_k
$R_{i,k+1} = R_{i,k} + [-\nabla f_k(R_i)\alpha_k$
No
Yes
$\alpha_k < \varepsilon_\alpha$
Intersection
?
$\alpha_k = \alpha_k / 2$
No
Yes
Cancel
Movement
k← k+1
Loop until:
$|f(R_{i,j}) - f(R_{(i-1),j})| < \varepsilon \; \forall\, i$

Fig. 5.25 Flowchart of the Main program

5.3.3.6 Subsequent tool positions

The rest of the positions are found by simply translating the cup mill from the position found previously upward and repeating the whole optimization process. This (translated) position of the tool constitutes the new initial guess for the next optimization. The translation of the cup mill is performed in the direction of the tool axis and the translation length depends on the number of points desired (from 0.01" to 0.04"). The positioning strategy presented in section 5.3.2 is, therefore, not applicable after the fist point of the tool path is generated. This is because the tool position (optimum) at the i^{th} point is translated upward and becomes the initial tool position of the following iteration. Thus, the positioning strategy described in section 5.3.2 is only required for the first (0^{th}) iteration.

The tool initial position and orientation for the subsequent optimizations are very close to their optimum positions. Since the transition length is very small, the shape of the blade at the next level of machining is not very different from the one at the previous iteration, thus simplifying the optimization process after the first point. The computation stops when all the positions of the tool for the entire blade surface are found. The curve passing by all these consecutive positions points constitutes the "tool path".

The flowchart of the whole tool path generation algorithm for machining the blade surface is the following:

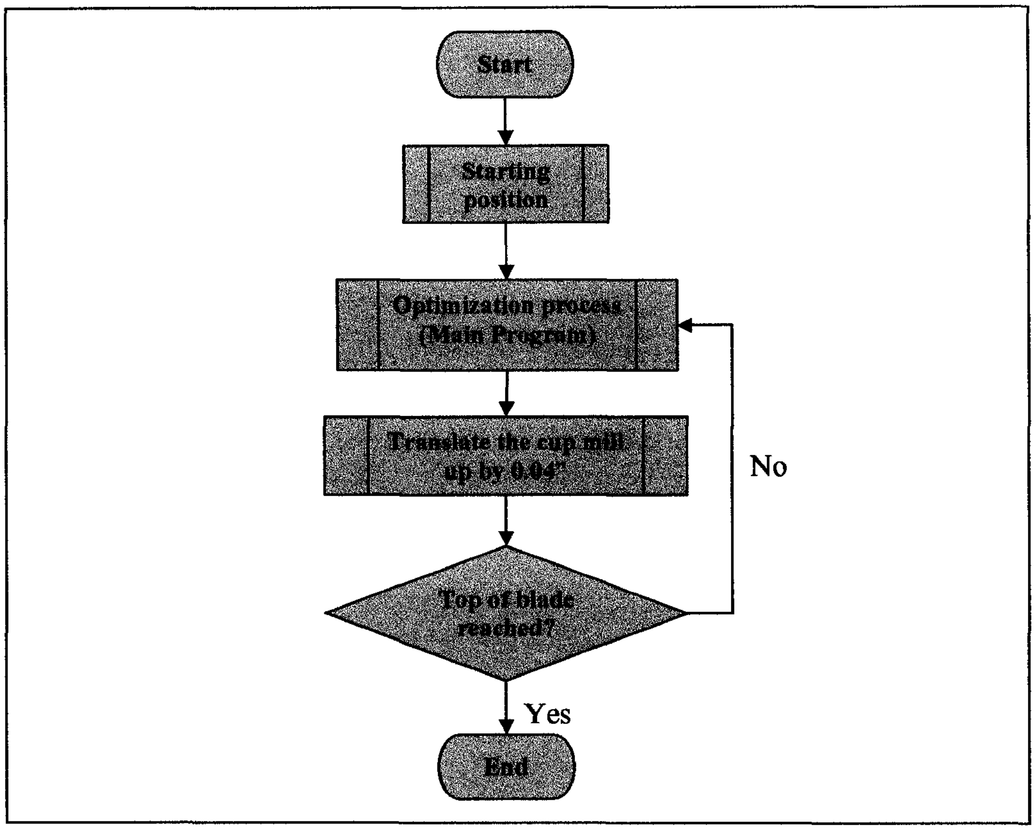

Start
Starting position
Optimization process (Main Program)
Translate the cup mill up by 0.04"
Top of blade reached?
No
Yes
End

Fig. 5.26 Flowchart of the complete algorithm

5.4 Smoothing out the tool path

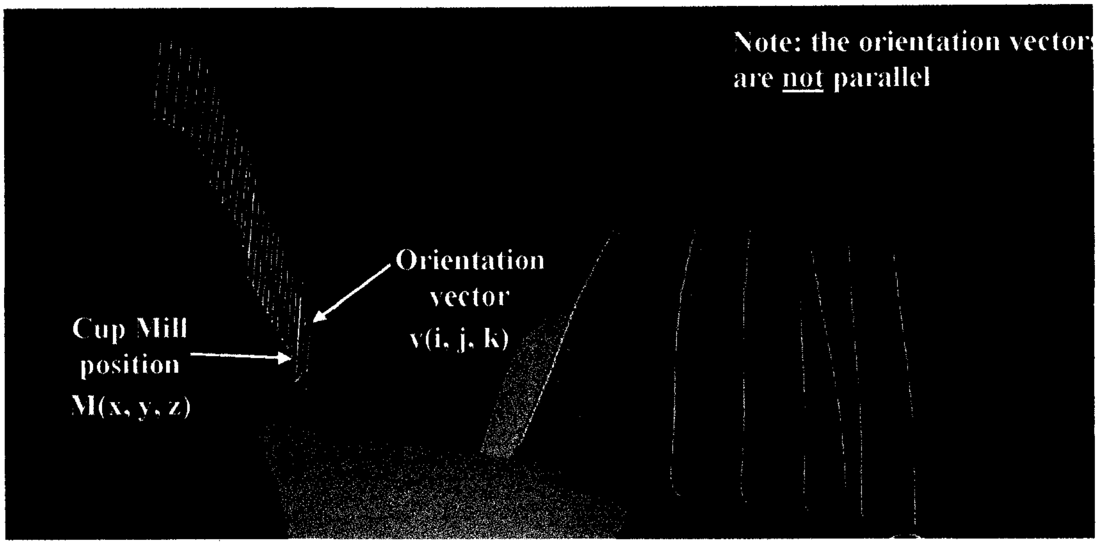


Fig. 5.27 Cup Mill positions and orientations

The complete tool path for machining one side of the blade is shown in Figure 5.27. This path is composed by a series of cup mill positions defined by a point M(x, y, z) and a cup mill orientation vector **v(i, j, k)**. The points define the center position of the tool at each level and the vector attached to it defines the tool axis. Each position in the figure is a solution of the optimization program at different level on the blade surface. The number of points constituting the tool path is given by the distance between levels (there are about 30 positions).

The tool path above is not smooth and cannot be used to perform any machining. The points of the tool path tend to be part a defined pattern. However, they are not perfectly part of a spline due to the limited precision of the program. As a consequence, the transition from one point to the next one is linear (straight line), which gives a very disturbed tool path. Because of the large size of the tool, even the smallest irregularities in the tool path have a major impact at the cutting region. The transition of the tool from one position to the next is very sharp because of the linear interpolation between them, giving a very broken tool motion.

The problem is simply solved by finding a spline that matches as closely as possible the cup mill positions (point and vector). The tool would follow this approximation spline instead of following the original positions, ensuring a smooth tool motion.

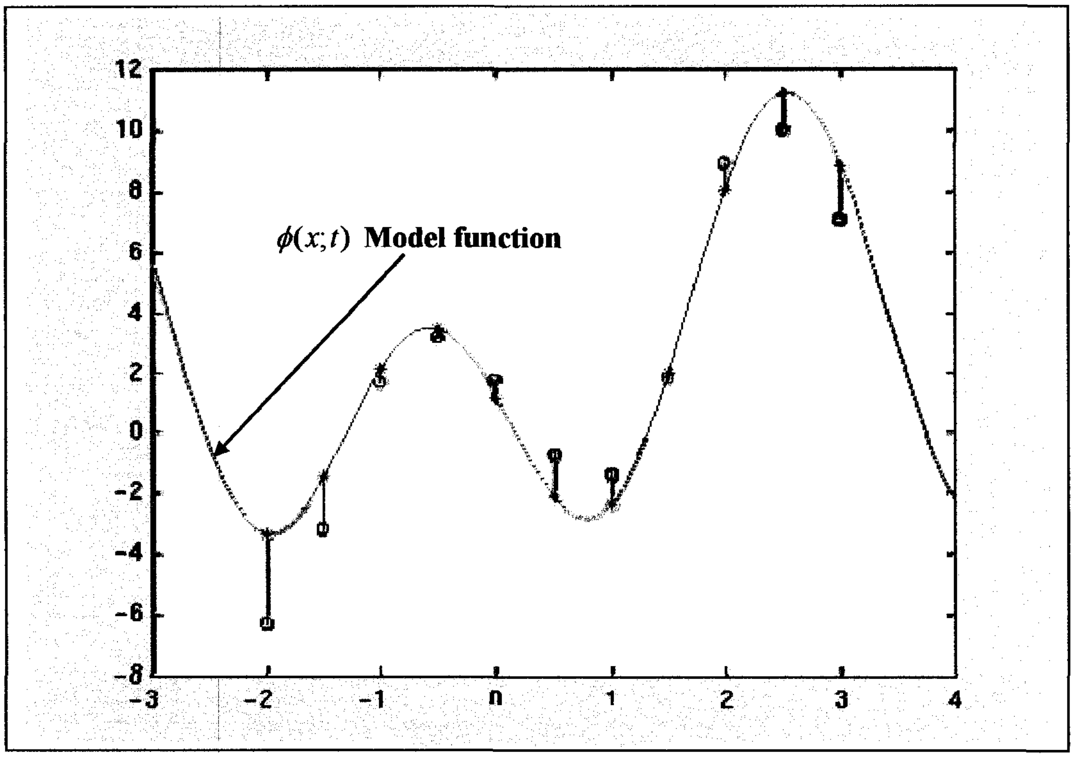


Fig. 5.28 Least Squares

Approximating the current tool positions by a spline involves a curve fitting process which fits equations of approximating curves to the data point ("observed" data). A curve with a minimum deviation from all the data points is desired. This problem is best solved by using the **method of least squares**.

The mathematics software Maple is used to smooth out the tool path. The cup mill position is defined by six coordinates: 3 are given by the position of the cup mill (x, y, z) and 3 define the orientation of the tool; **(i, j, k)** vector. A best-fit curve is found for each coordinate separately. Each one is a function of the "ordinate" *t* which is simply the point number (1, 2, ..., n). In this case, the model function is a polynomial of degree 7 for all the six data points:

$$\phi(x;t) = a_1 + a_2x + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \tag{5.7}$$

The software identifies the parameters of the model functions best fitting the data points. The graphics of the solutions models are presented below in Figure 5.29:

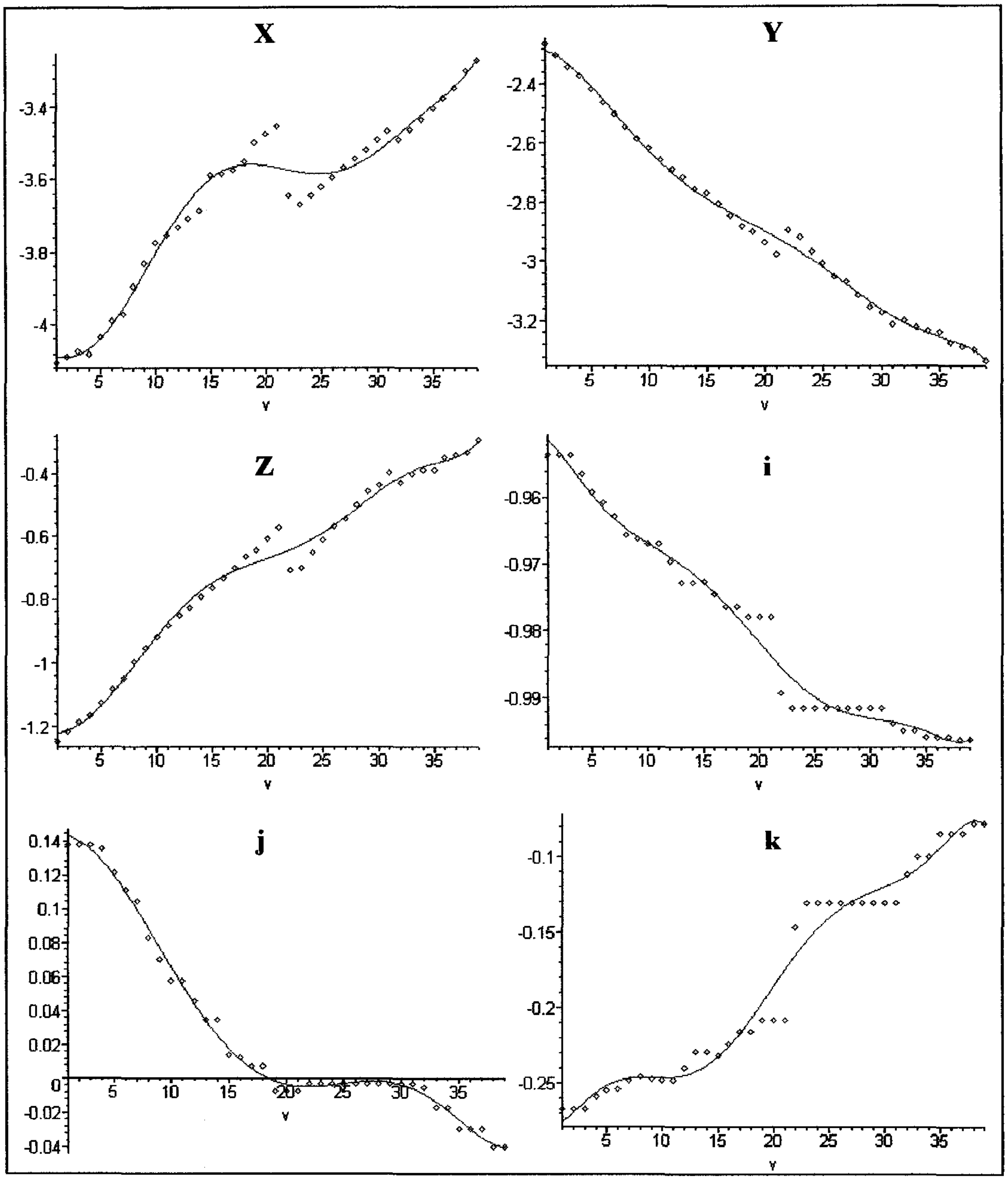


Fig. 5.29 Least Square functions of each of the 6 positioning coordinates

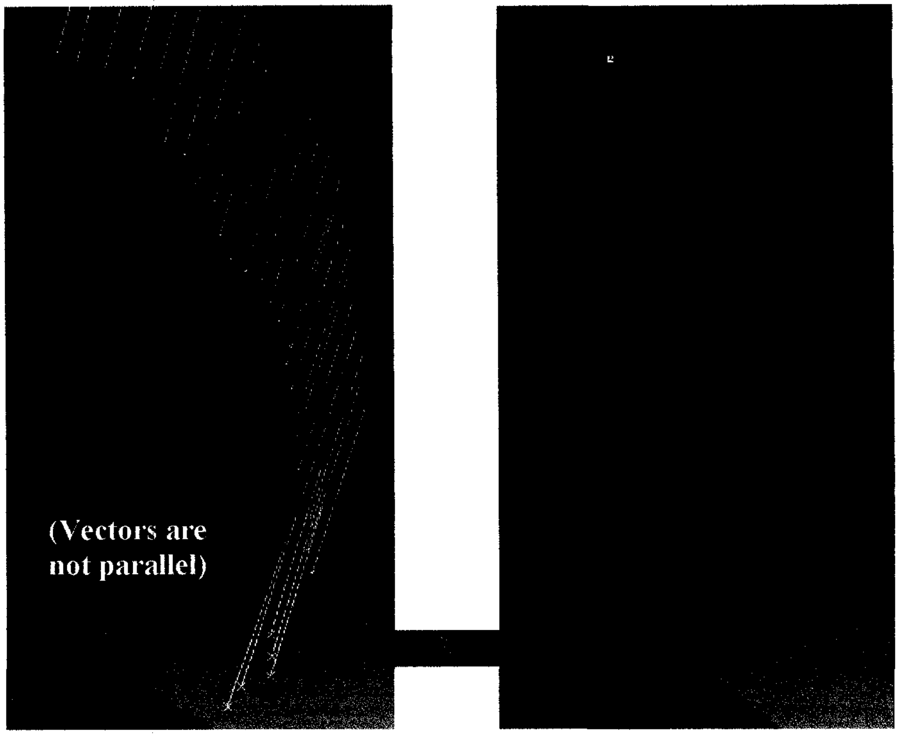


Fig. 5.30 Tool positions before and after smoothing operation

It is important that the model function selected approximates very well the data points: due to the large diameter of the tool, a small error in the position and orientation of the tool can have a major impact at the cutting region. For that reason, a polynomial of degree 7 is selected, since the error obtained by a cubic spline would be larger that what can be accepted.

Figure 5.30 illustrates the new smooth tool path compared to the old one.

5.5 Complete tool path

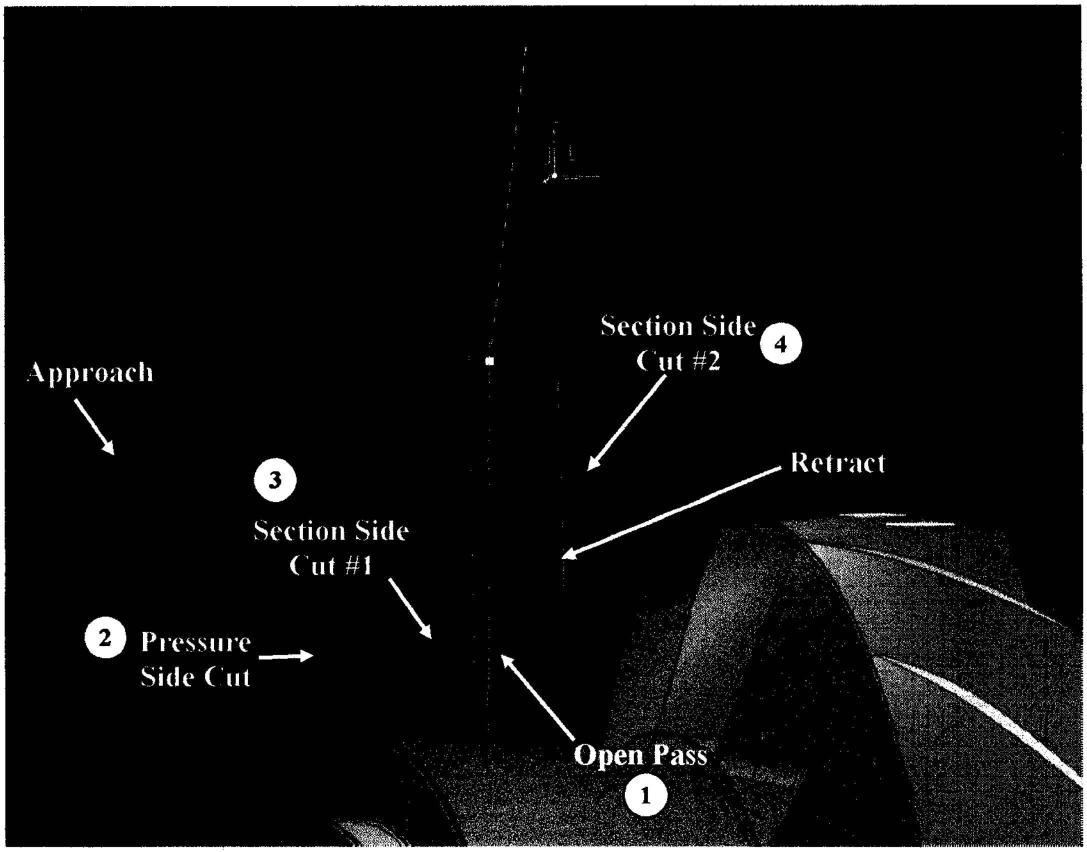


Fig. 5.31 Complete tool path for one blade

Using the procedure described in this chapter we compute all the passes composing the tool path required to machine both opposing surfaces of the blade. The complete tool path is presented in Figure 5.31 after the smoothing operations were performed. This path is composed of an open pass (plunging motion with a fixed tool axis) and three 5-axis machining passes. The path presented in Figure 5.31 includes also the retracting and approaching motions of the tool, but only the 4 tool motions (1 to 4) are performing any machining on the part.

The necessity to have three tool motions to machine two blade surfaces (pressure side and suction side) is due to the pronounced blade curvature of the suction side. Consequently, this side of the blade is very hard to machine in a single pass, because it would leave too much material on the blade surface.

Better results are achieved by separating the suction side surface into two regions (two target surfaces) and machine each one in distinct tool motions. The resulting surfaces are smaller and each one can be machined more precisely by the cup mill. This separation is performed manually by the user. It can be noted that for another blade configuration (larger blade radius of curvature) such a separation might not be necessary and a single pass could be sufficient.

This concludes the path generation for machining the space between two consecutive blades. Finally the tool path is indexed about the part axis (indexing angle depends on the number of blades of the part, 19 here) to generate the tool path to machine all the blades (see Fig. 5.32).

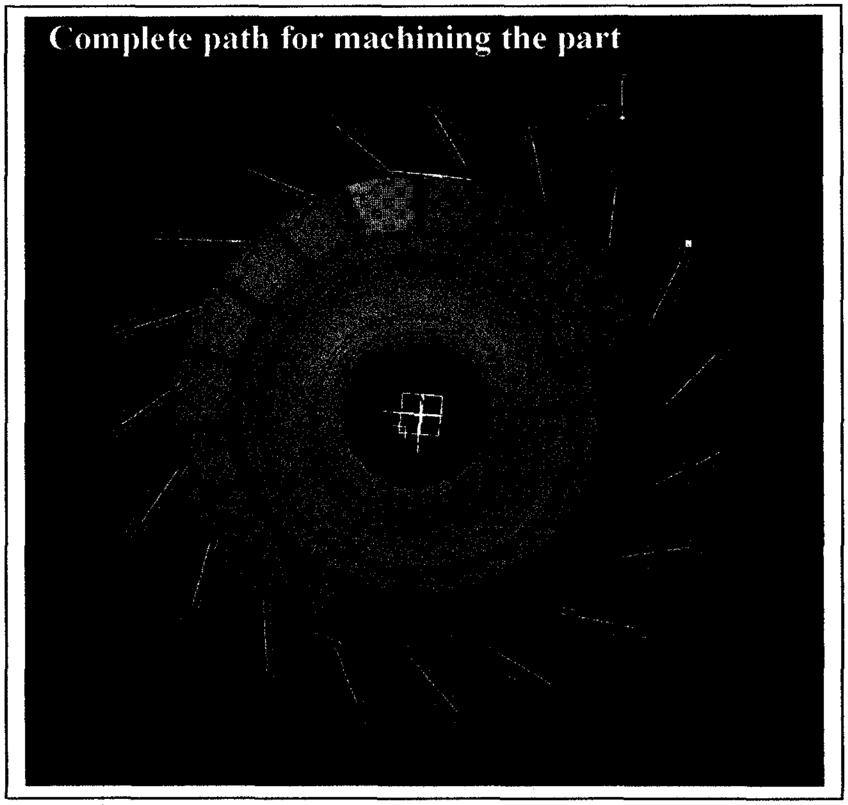


Fig. 5.32 Complete tool path

CHAPTER 6: Tests & Results for IBR machining with cup mill

6.1 Program simulation in CATIA V5

6.1.1 Parts

The parts selected for the simulation of the program are presented in Figure 6.1. These parts are IBRs from two different PWC engines (PW307/308 and PW545). The motives of selecting these parts are:

- small blades
- blades are not highly twisted
- sufficient available space between consecutive blades

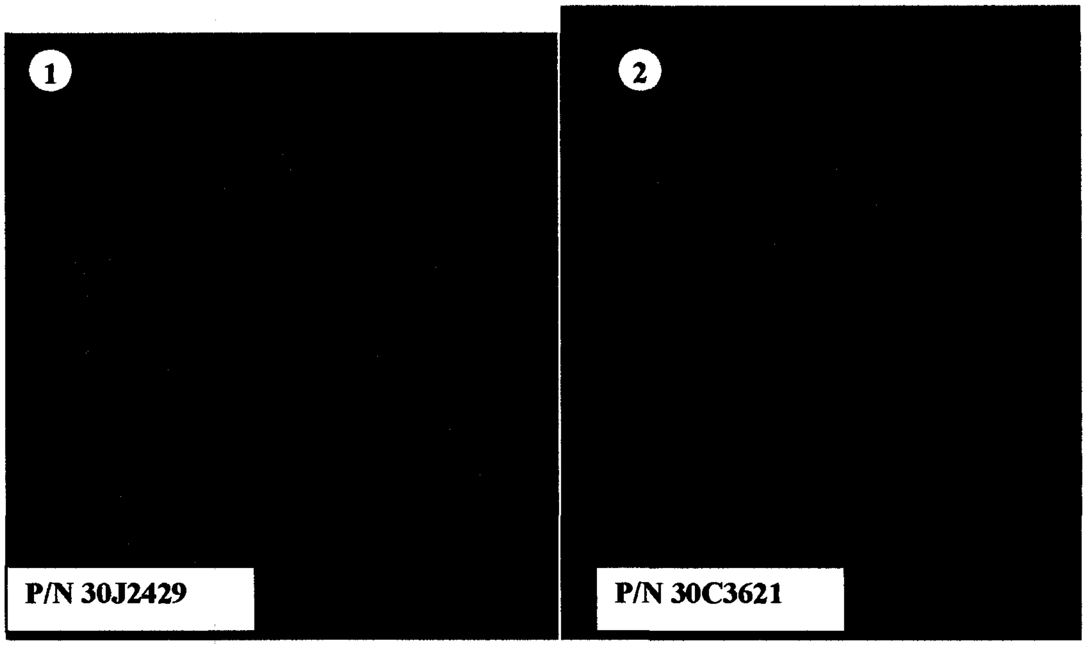


Fig. 6.1 IBRs selected for tests

6.1.2 Tools

Firstly, it is required to specify the cup mills used to machine our two parts. The cup mill dimensions are selected in such a manner as to maximize the accuracy of the positions (solutions) of the cup mill on the blade. For a particular IBR, the accuracy of the solutions obtained depends principally on two factors: the cup mill thickness and the cup mill size (diameter).

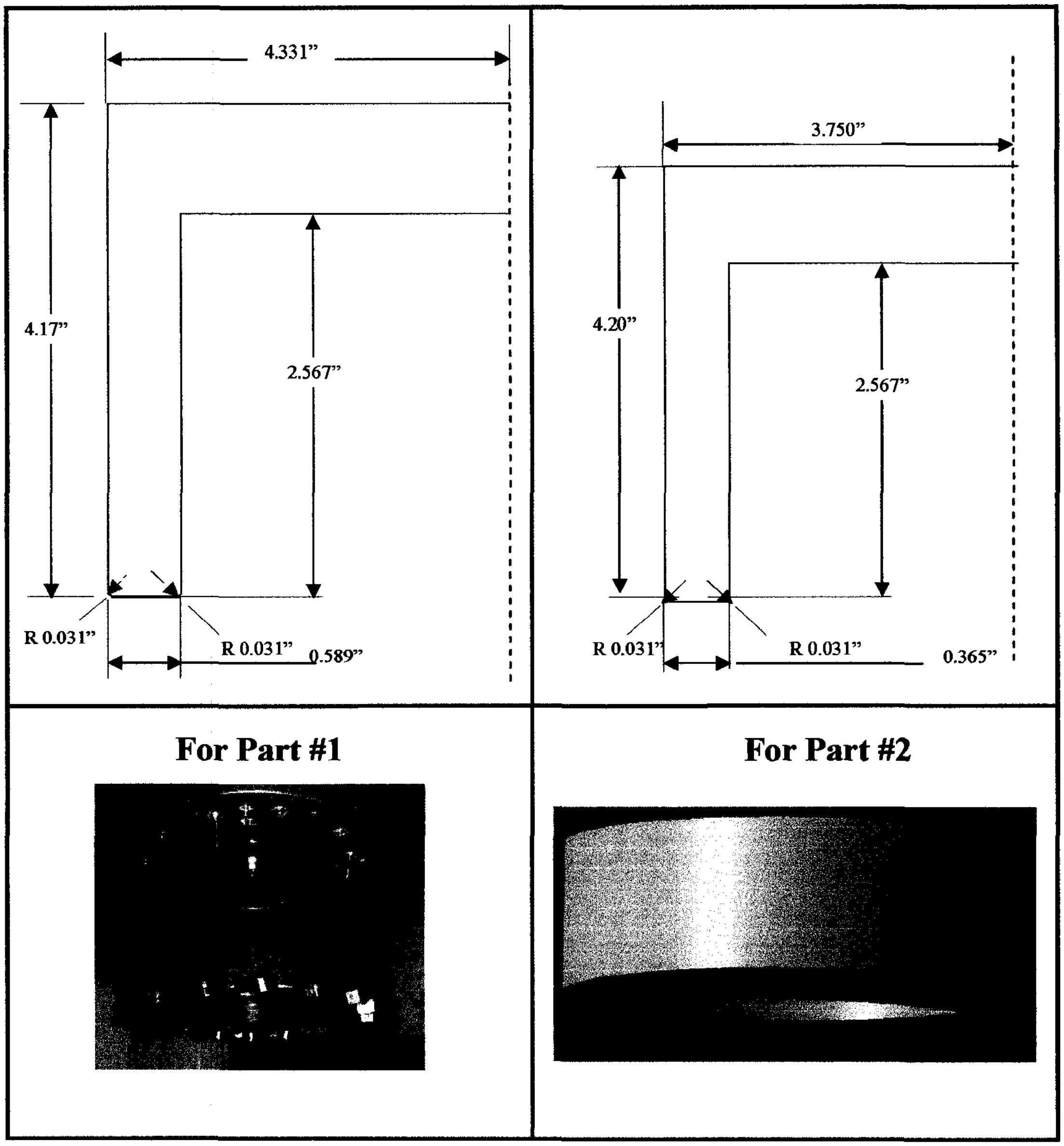
From the tool rigidity point of view, the tool thickness should be as large and the tool diameter as small as possible. On the other hand, such a tool is limited in its freedom of movement between the blades, increasing the excess material left on the blade surface. So, from the machining point of view, a thin and large diameter cup mill is preferred. The cup mill thickness is basically the best compromise between the tool rigidity, tool life and productivity on one side, and machining accuracy on the other. The available standard insert dimensions is another factor to be considered when defining the cup mill thickness.

In parallel, the diameter of the tool is selected in function of the blade radius of curvature. The minimum mismatches between the tool and the cup mill is obtained when the tool and the blade curvature are similar. However, the physical limitations of the machine tool, like the maximum tool weight and size, have to be considered. As an example, most 5-axis machines at PWC have a maximum tool weight capacity of 50 Kg and a maximum tool size of 9 inches so the tool dimensions must be adjusted as to respect these constraints.

Considering the IBRs selected for the test, the optimum cup mill tool is different for each part number since the geometries of the two parts are very different. The first part, having a large blade radius, requires a tool of about 10 inches in diameter. As mentioned however, such a large tool exceeds the limitations of the machine; consequently, the diameter of the tool will be around 8.5 inches. Also, due to the twisted blades and the small spacing available between then, the cup mill thickness cannot not be above 0.600 inches. The cup mill developed by Sandvik respects these conditions.

For the second IBR, the tool supplied by Sandvik is too thick to fit between the blades, so another tool is designed. This part has a smaller blade radius of curvature, so a tool of 7.5 inches of diameter is selected, with a thickness of 0.365 inches. This tool is only used to test the program in CATIA V5 for this IBR; it was not manufactured, since no physical tests are forsaken for this part number at this time.

The dimensions of the tools used for the two parts numbers are presented below in Figures 6.2.



For Part #1

For Part #2

Fig. 6.2 Cup mill dimensions

6.1.3 Results

The algorithm created following the methodology presented in chapter 5 is used to compute the tool path for the two part numbers.

The mismatches between the cup mill cutting surface and the blade are within the requirement of 0.03" for both parts. To illustrate the accuracy of the algorithm, a sample position (solution) of the cup mill on the blade for each part is illustrated in Figures 6.3 and 6.4. These figures present measures taken in CATIA between the 5 target lines on the blade (pressure side) and the cup mill cutting surface, for both parts. The measures are tabulated below (rounded to the nearest thousand of an inch):

Table 6.1 Tabulated deviation between the cup mill and the target curves

	Line 1	**Line 2**	**Line 3**	**Line 4**	**Line 5**
Part #1	**0.019"**	**0.002"**	**0**	**0.003"**	**0.004"**
Part #2	**0.011"**	**0.008"**	**0**	**0**	**0**

From Table 6.1, it can be noted that, in both cases, the maximum deviations are well within the imposed tolerances (less than 0.030").

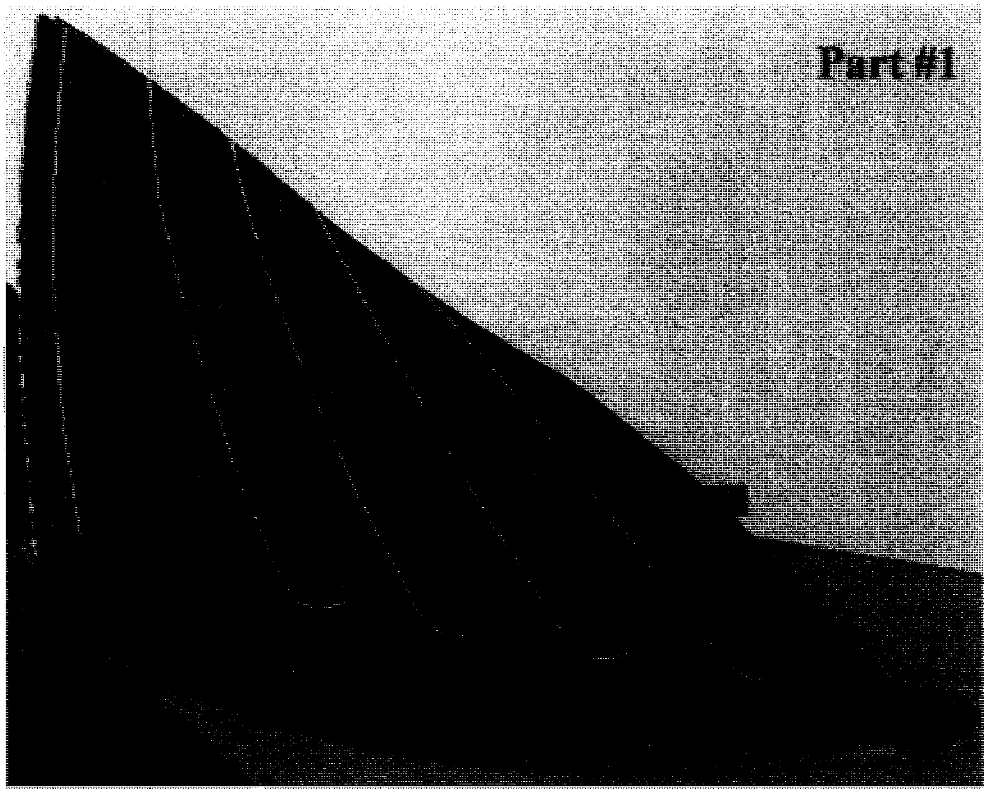


Fig. 6.3 Measured distances between the curves on the part surface and the cup mill (part #1)

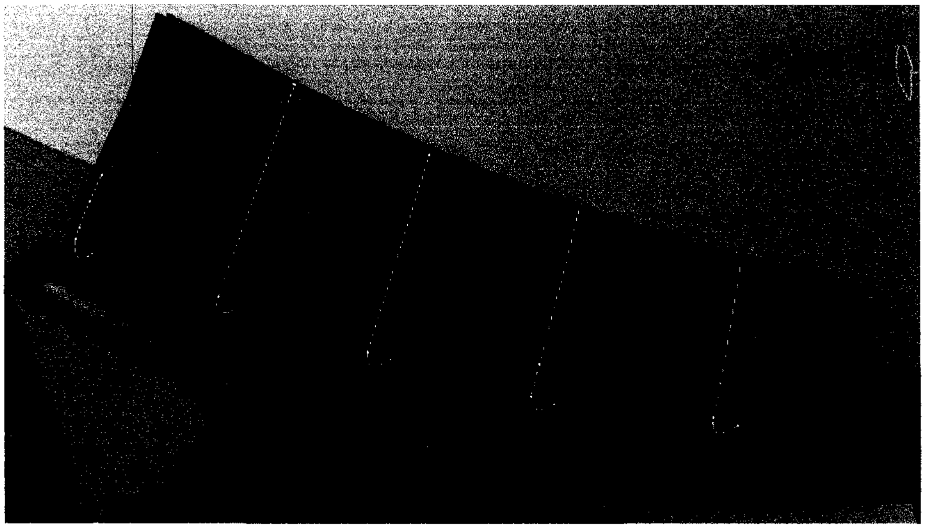


Fig. 6.4 Measured distances between the curves on the part surface and the cup mill (part #2)

The tool path found for the first blade is indexed about the rotor central axis to machine the rest of the blades. The first compressor has 19 blades, so the path is indexed 18 times by an angle of 17.947° and the second compressor has 31 blades, so the path is indexed 30 times by 11.613°.

To validate the tool path, a machining simulation is performed in CATIA V5. Both parts are machined (virtually) from a stock of material using a cup mill and videos of both simulations are presented in the CD-ROM attached to this thesis.

The next set of pictures illustrates the resulting cup mill roughed parts, in comparison to the finished ones. From the figures, it is obvious that the precision of the roughing obtained by the program is excellent.

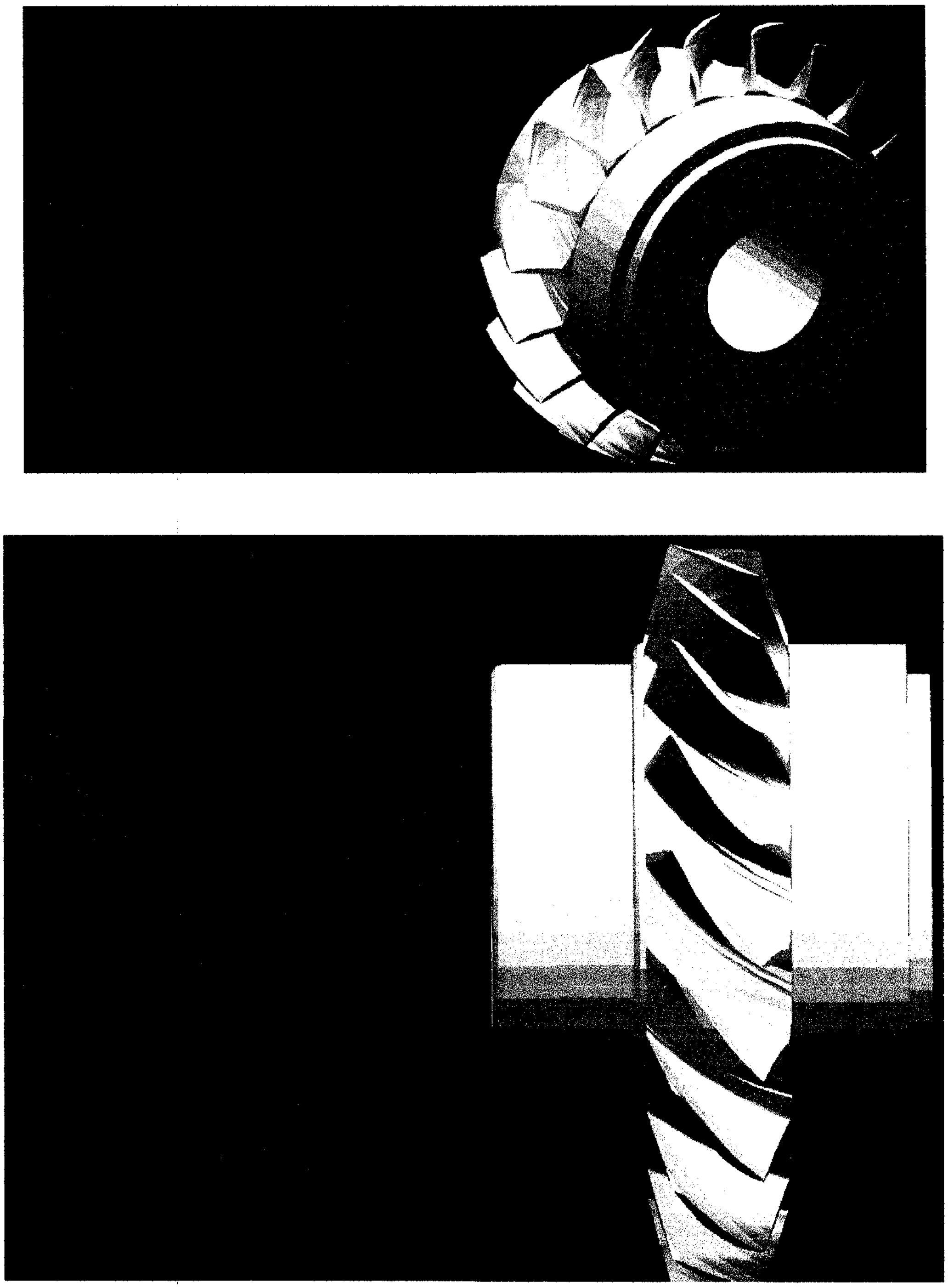


Fig. 6.5 Rough with cup mill vs. finished part (part #1)

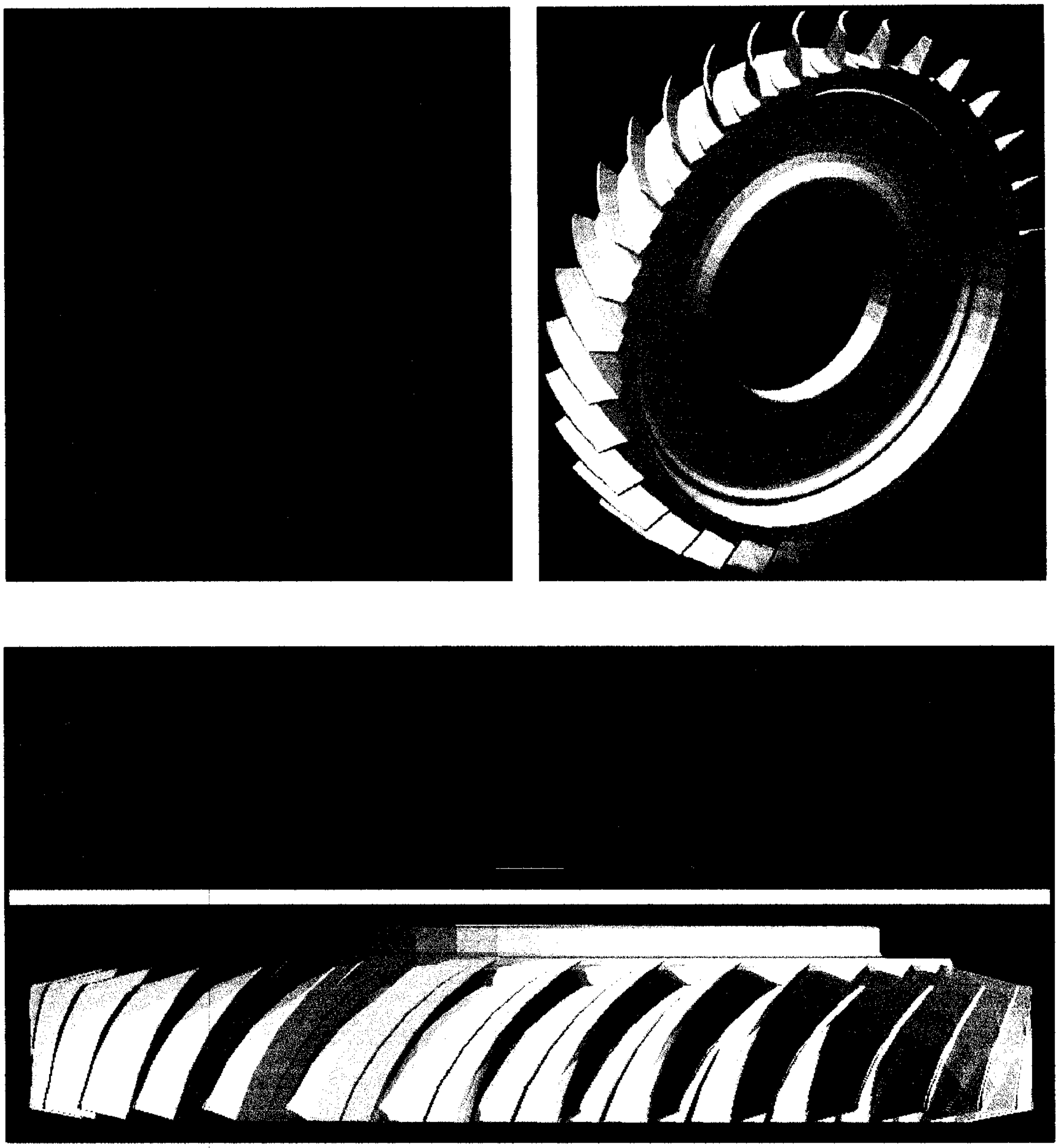


Fig. 6.6 Rough with cup mill vs. finished part (part #2)

Compared to the finished parts, the roughed blades of both parts are slightly thicker. This is due to the deviations between the cup mill and the target surface, but also due to the fact that the target surfaces themselves are voluntarily offset from the finished surface by 0.020" each side of the blade to eliminate any possibility of gauging into the part.

6.2 Machining tests with the cup mill

6.2.1 Machine tool

Real machining tests have been performed using a Taurus 3S 5-axis milling machine, manufactured by Boehringer (see Figure 6.7). This is a strong milling machine that can accept at the same time large parts and large cutting tools. The spindle power consumption during this test is low because, even if the diameter of the tool is large, the spindle speed is low, at about 70 RPM. Thus, all the specified limitations of the machine tool are respected.

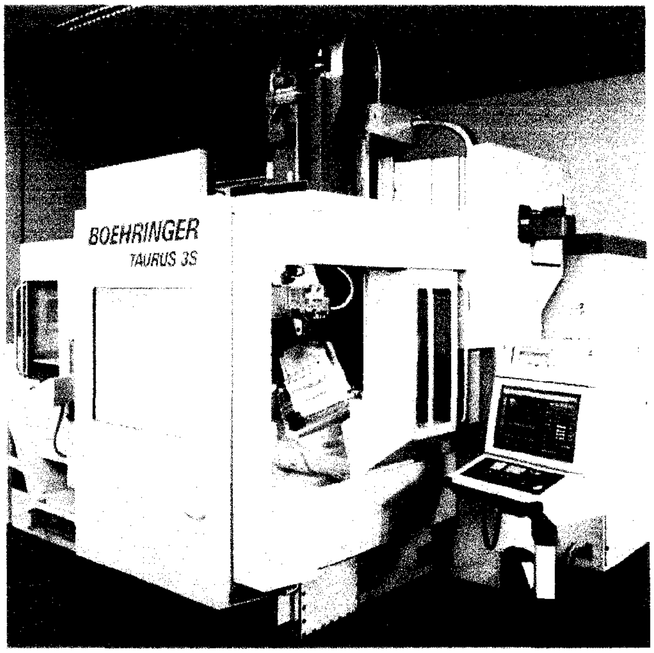


Fig. 6.7 Taurus 3S machine tool

6.2.2 G-Code

The tool path obtained by the program is in the form of an APT (Automatically Programmed Tools) file. However, this file only contains the successive tool positions with respect to a reference point on the part that do not coincide with the reference point (zero) of the machine-tool. This file has to be transformed to take in consideration the machine tool and the fixture used to hold the part in the machine.

This preparation of the machining code (G-Code) is performed by another software called VERICUT by GTech. In VERICUT is inserted as inputs: the 3-D model of the stock part, the finished part (for verification purposes), the fixture and the 3-D model of the machine-tool. Then, the parts and the fixture are positioned correctly in the machine and the tool path is referenced with respect to the zero of the machine-tool. The effective length of the tool (i.e. the length from the end of the tool to the spindle) is set to 15", which is a temporary "dummy" effective length of the tool. Prior to testing, the actual effective length of the tool is measured by the operator and the compensation (dummy minus actual) is performed directly on the machine.

The machining program is validated in VERICUT. This simulation permits to verify the following elements:

- part collision
- stock collision
- fixture collision
- machine-tool collision
- machine-tool movement limitations
- visualize the resulting machined part

6.2.3 Fixture

The fixture used to hold the part is a modular fixture and is shown below in Figure 6.8.

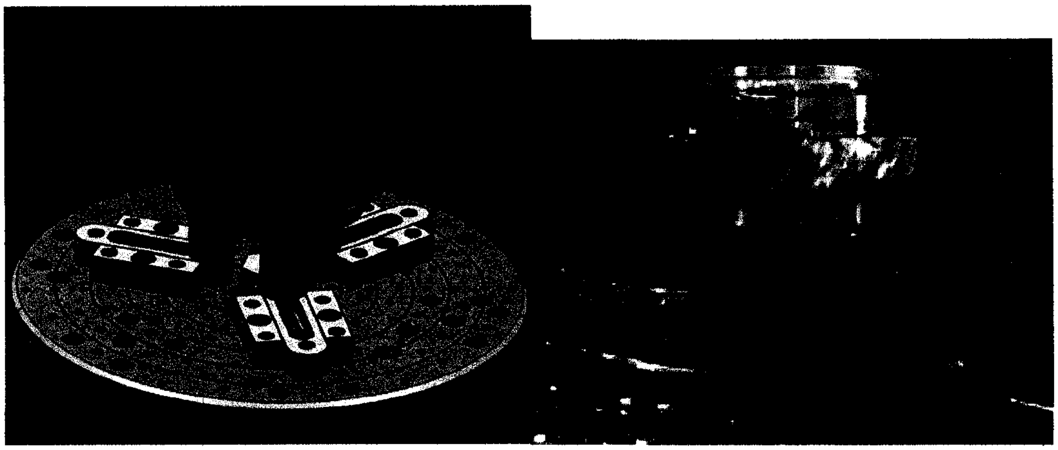


Fig. 6.8 Modular fixture

6.2.4 Tool

The tool used for this test is the tool developed by Sandvik, presented in Figure 6.9.

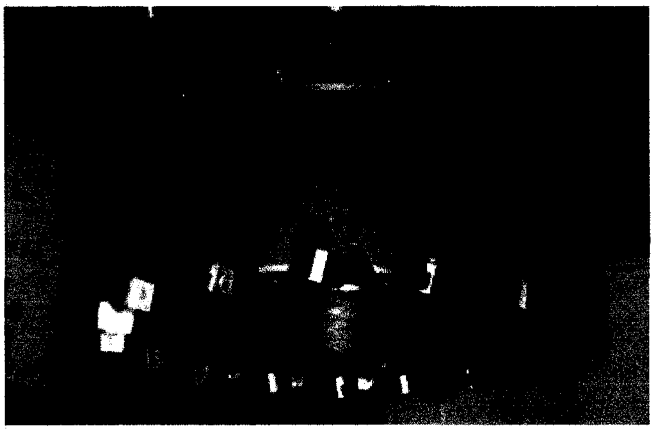


Fig. 6.9 Tool used

6.2.5 Cutting parameters

Table 6.2 IBR roughing tests – Cutting parameters used

Speed	1900 in/min
	70 RPM
Feed	0.045 IPR (0.00577 IPT)
	3.00 IPM
Coolant	Water base
Machine	Taurus 3S

6.2.6 Results

Fore this test, IBR #1 (30J3429) is roughed using the cup mill produced by Sandvik presented earlier. The stock part was made from aluminum instead of titanium for safety and monetary reasons. In Figure 6.10 is presented the pictures taken during this test:

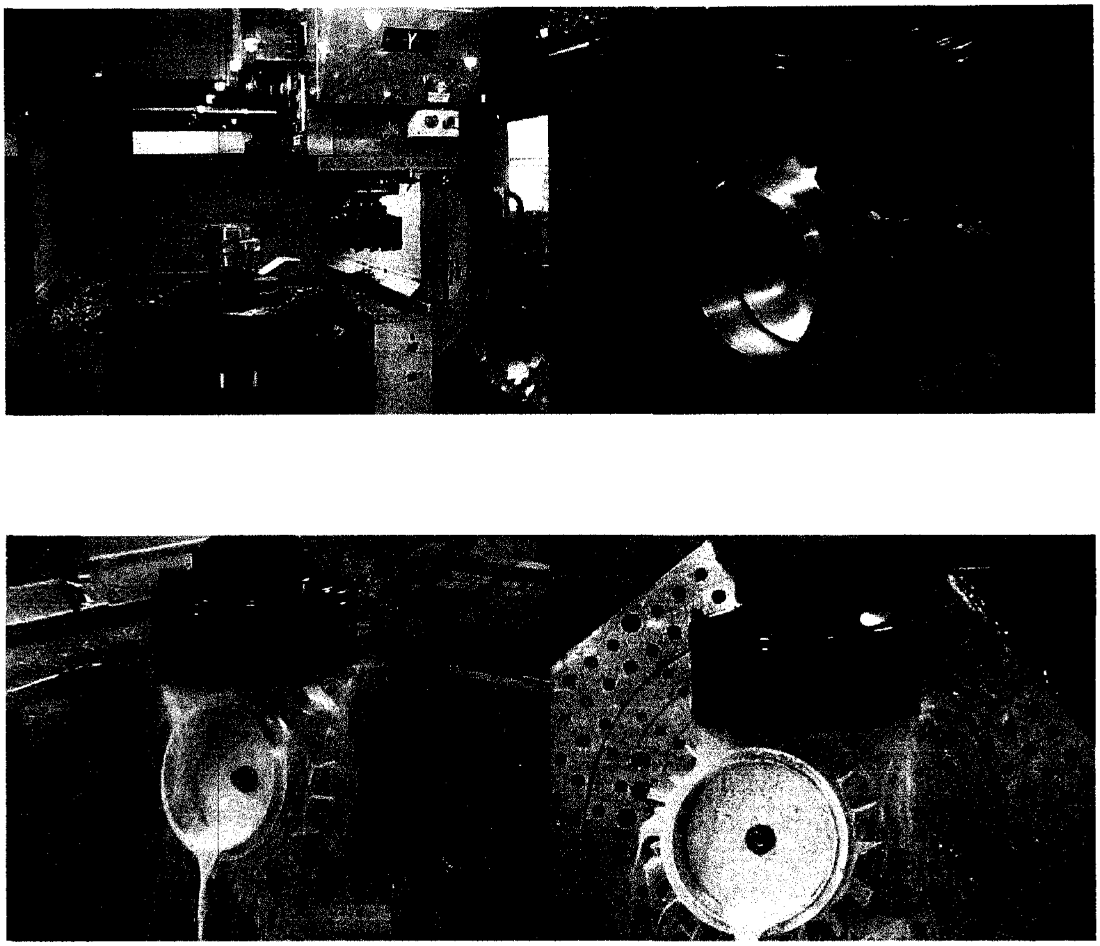


Fig. 6.10 Machining tests

The resulting machined part using the cup mill is presented in figure 6.11:

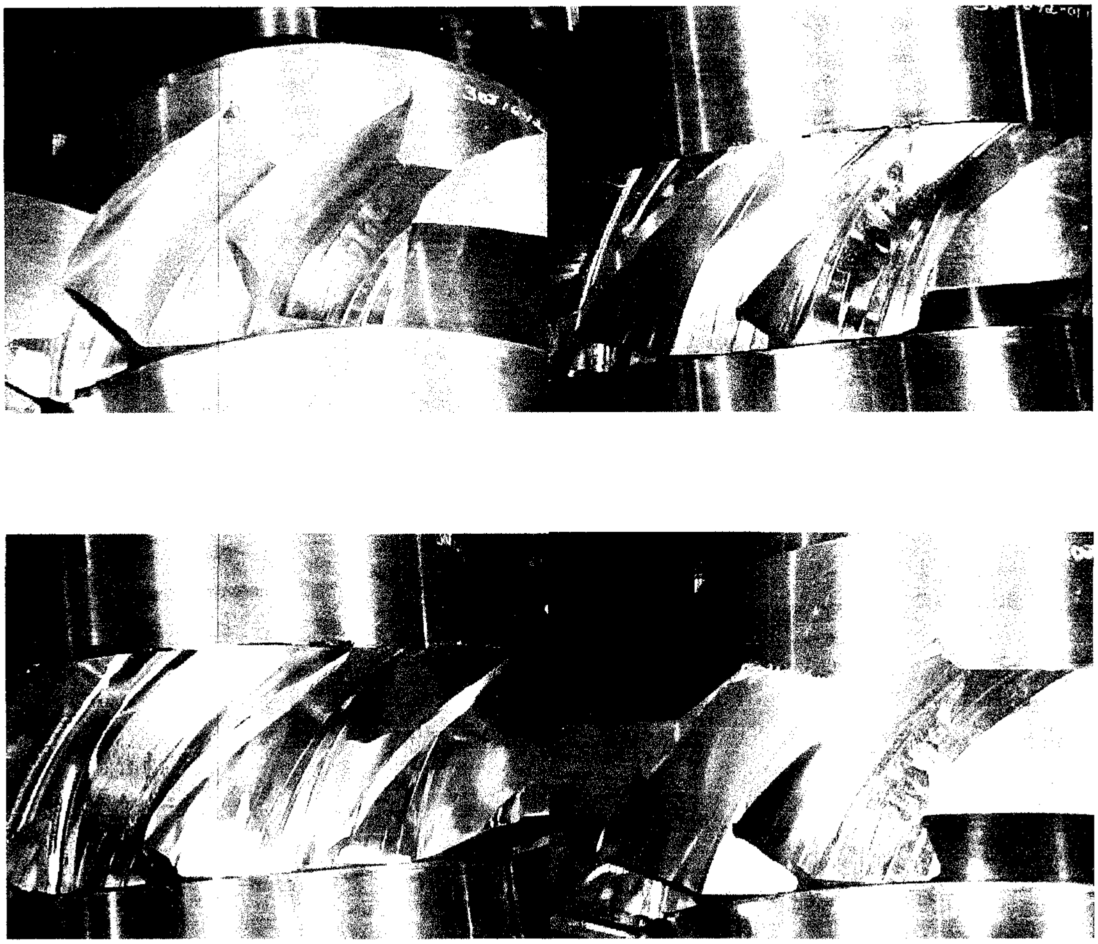


Fig. 6.11 Results

The roughed blades obtained during this test confirmed that the maximum volume of material was removed by the cup mill cutter. The roughed shaped of the blade has only a maximum of 0.02" to 0.03" inches excess material over the desired surface. The high machining accuracy obtained suggests that this technology could replace the current roughing tools since a similar amount excess material is left on the blade after the current roughing operation. The surface of the blade is very smooth due to the smoothing operation performed on the tool path, discussed in the previous chapter.

It is difficult to evaluate exactly the productivity of this new technology since the test was carried out in an aluminum part instead of a titanium one. However, based on the results obtained during the machining tests in titanium performed by Sandvik, at least a three time increase in productivity is expected. The large chips obtained during the tests confirm the high volume removal capacity of the cup mill cutter (presented in Figure 6.12).

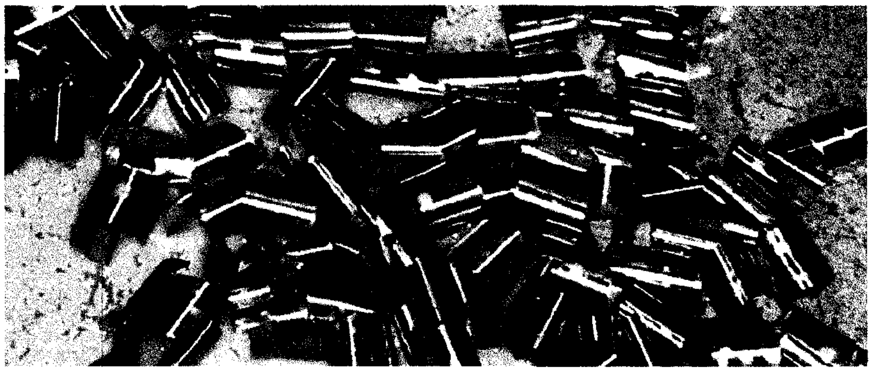


Fig. 6.12 Chips

In conclusion, table 6.3 summarizes the positive and negative aspects observed during this test:

Table 6.3 Positive & negative aspects of the performed tests.

Positive	Negative
• Stable machining • Very productive • Maximum material volume removal • Good runout control of the tool (less than 0.0005") • Large chips • Smooth blade surface	• Manual insertion of the tool - again • (missing slot on tool necessary to be handled by the robotic arm of the machine tool) • Friction between the cassettes and a corner of the blade, due to weaknesses present in the algorithm (discussed in recommendation section)

CHAPTER 7: Conclusions & Future Work

7.1 Observations

The following observations can be made from the presented research:

- cup mill cutter is 3 times more productive than the current ball endmills used;
- this technology is easy to implement in production;
- the use of standard inserts reduces the cost of tool and are easily supplied;
- the algorithm generates a tool path that:
    - removes a maximum volume of material;
    - high precision (<0.030 inches of excess material are left on the blade surface);
    - machines efficiently the blade;
    - applicable to many part numbers, using different tools;
    - tested in production.

7.2 Conclusions

The principle presented in the previous chapter is a generic approach, theoretically applicable for different IBRs using different cup mill cutter of different dimensions and shapes. Form all the IBRs manufactured at PWC, the ones that can be potentially roughed using the new method are identified. The shape of the blades and the available space between them determines if the IBR can be machined using a cup mill: IBRs with highly twisted blades and/or with blades that are positioned close together are not suitable candidates. In general, the first stages IBRs from all the engines are discarded, due to the severe twist of their blades, making it impossible for a cup mill, indifferently of its size, to fit between the blades. Figure 7.1 presents the different PWC IBRs that could potentially be roughed using a cup mill cutter. The two part numbers in which the new technology was tested are identified.

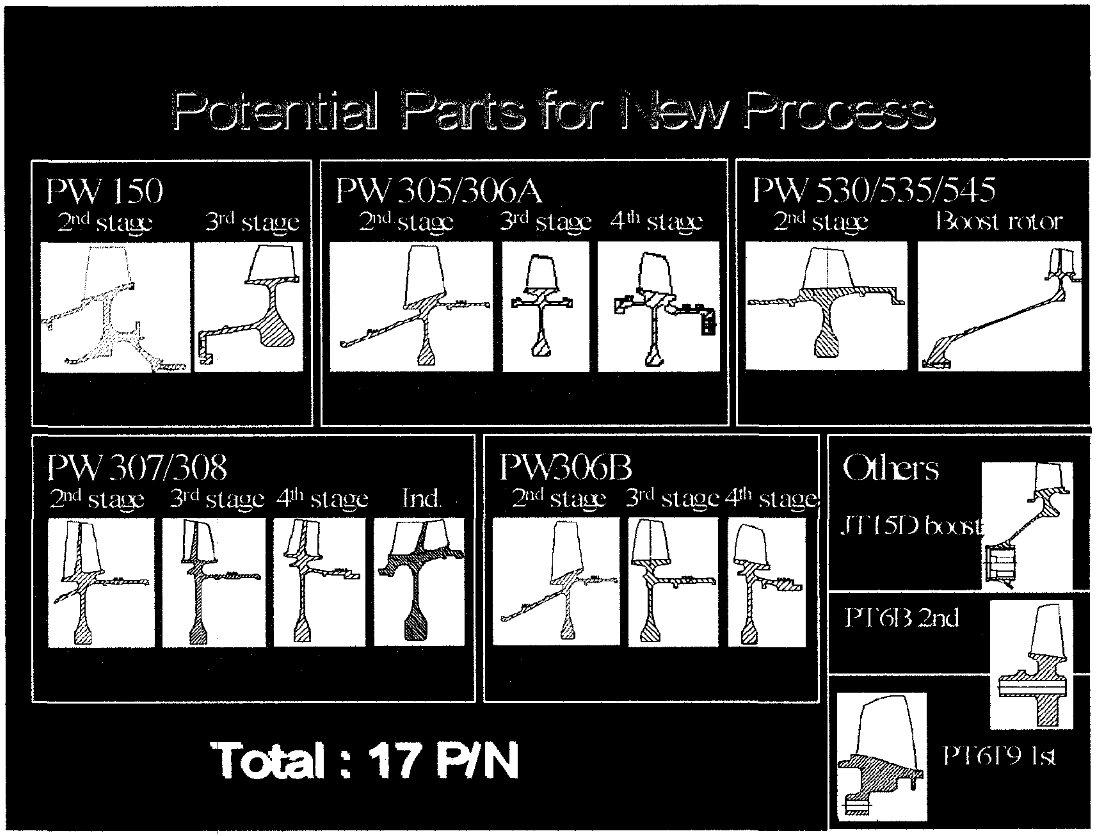
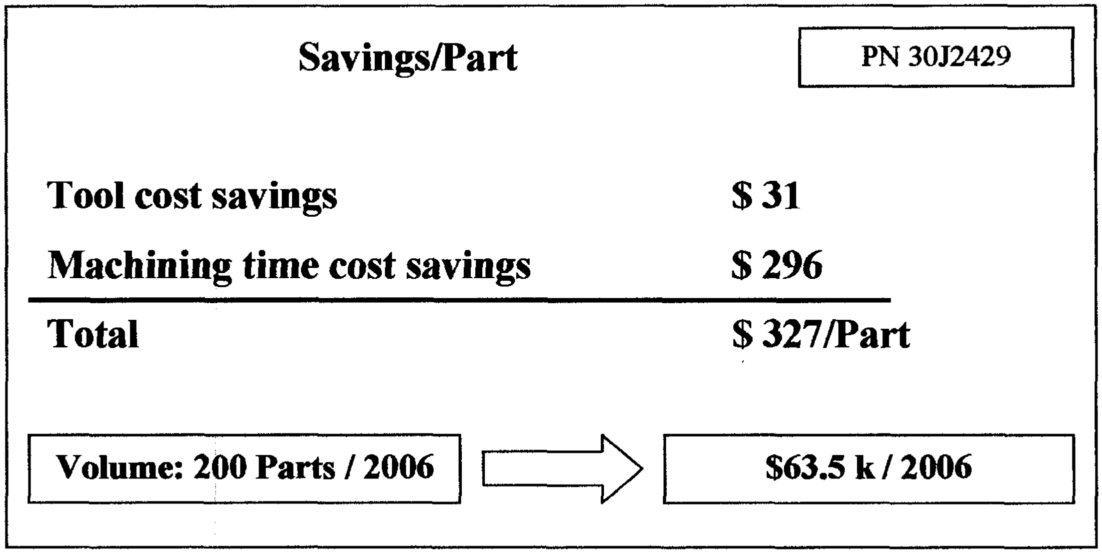


Fig. 7.1 Potential Parts for New Process

7.2.1 Potential annual savings

Considering the annual production at PWC of the part number selected for the test (PN 30J2429), roughing this part using this new technology represents savings of more that 63,000 dollars of 2006:



Savings/Part

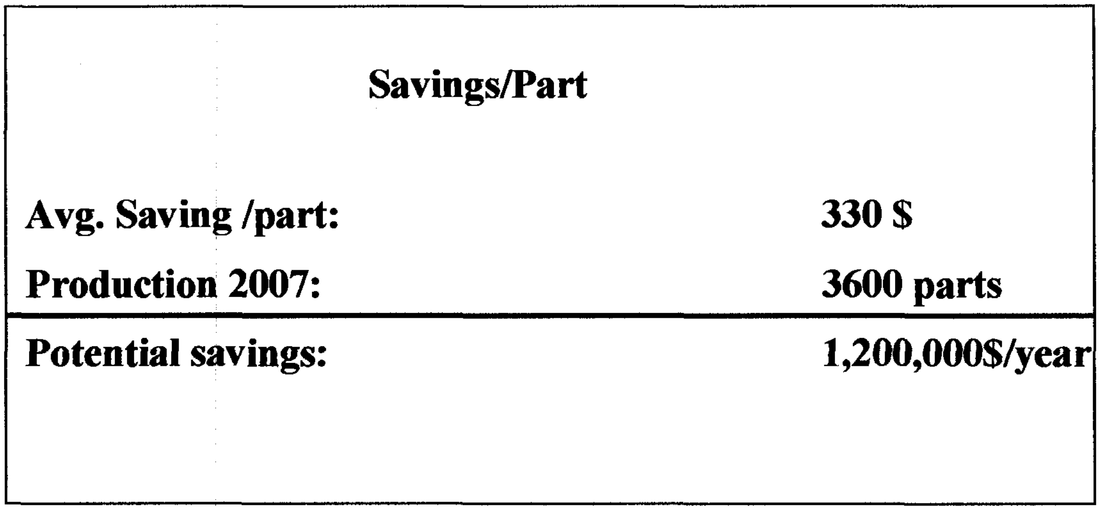
PN 30J2429

Tool cost savings	**$ 31**
Machining time cost savings	**$ 296**
Total	**$ 327/Part**

Volume: 200 Parts / 2006 ⇒ **$63.5 k / 2006**

Fig. 7.2 Savings evaluations of new rouging method

In Figure 7.1 are presented 17 IBR part numbers that could potentially be roughed using the cup mill method. Implementing the cup mill method for all these part numbers could represent major saving for Pratt & Whitney Canada. An evaluation for all these part of the costs reduction associated with the introduction of the cup mill tool in their production process is impossible due to the lack of time. However, using an average saving per part of 330 $ (saving of PN 30J2429), an a cumulative production for 2007 of all those 17 part number of 3600 parts, the introduction of the cup mill tool in the manufacturing process represents a potential saving of over 1 M$ per year.



Savings/Part	
Avg. Saving /part:	330 $
Production 2007:	3600 parts
Potential savings:	1,200,000$/year

Fig. 7.3 Saving estimation / 2007

7.3 Research contributions

This paper proposed a new technology to rough integrally bladed rotors. The research contributions of the projects can be summarized as follows:

- **New tool for machining IBRs**

The cup mill cutter was developed in collaboration with major tooling suppliers and different tool prototypes were proposed. The most important tool characteristics that this tool possesses over the currently used taper ball endmills are:

- Good tool rigidity,
- Usage of inserts.

This enables the cup mill cutter to have a good chip evacuation potential. Consequently, a higher material removal rates is possible than the current IBR machining tools.

The particularity of this tool is that it is machining with both the interior and the exterior of the tool simultaneously. Those two cutting regions must be controlled at the same time to prevent and gouging or collisions with the part. The particularity of this tool requires developing an algorithm to generate the tool path since no suitable CAD/CAM software exists.

- **Development of the tool path generation algorithm**

The algorithm developed for positioning the tool is very different from the tool positioning strategies developed for point milling and flank milling. This is due to the particular shape of the cup mill cutters compared to the endmills and to the difference of tool kinematics. Although it is inspired from the "Triple tangent flank milling method" [8], this "cup milling" positioning method comprises important differences. In our case, the tool is rotating about a tangency point and the algorithm tries to minimize the deviations throughout the whole length of the blade simultaneously by a minimization process. This algorithm has also a build-in collision verification sub-routine that takes corrective measures to prevent collisions. This is not found in flank or point milling algorithms since the shape of the tool prevent the possibility of collisions with the part.

- **Demonstrated productivity and low machining costs**

The productivity of the cup mill tool to rough IBRs was demonstrated during machining tests performed by Sandvik. It was demonstrated that this tool is at least 3 times more productive than the current taper ball endmills. As a result, over one million dollar of savings can be expected if the cup mill tool is introduced in the manufacturing process of all the part numbers presented in Figure 7.1.

- **Potential solution to machine inconel IBRs**

Roughing IBRs using the cup mill cutter represents a cheep and effective solution to manufacture axial compressor in inconel 718:

- o total cost per part, comparable to actual manufacturing costs for titanium compressor
- o roughing time per part less than actual time for titanium compressors.

7.4 Future Work

Computation time

The algorithm developed in this project requires an extremely long time to compute the tool path. Just to determine one of the three passes constituting the tool path, the computation can take up to 24 hours of computing. A total of three to four days are required to generate the full tool path to machine an IBR. It is essential to solve this problem, prior of entering this technology in the production process of axial compressors.

The long computational time of the algorithm is due to the interfacing between CATIA V5 and Visual Basic. Each operation performed by the program is displayed on the computer monitor. A single operation takes only a few seconds to be displayed but, thousands of operations are performed, and thus, total time required is considerable.

A possible solution to this problem is to create an algorithm without interfacing with a CAD software. All the build-in functions simply used by the current algorithm from CATIA would have to be created in the future version of the algorithm. The IBR part and the tool would have to be published into graphical formats, understandable by the algorithm. The resulting algorithm would certainly be more complicated to create, but this is the only effective solution to shorten the computation time. In this case, only the final result is displayed.

Collision with stock

Some problems were observed during the machining tests concerning small collisions between the tool body and the part. These collisions are due to weaknesses present in the Collision Evaluation subprogram. The feasible region is defined as all the possible positions of the tool not intersecting the part body. However, no verifications are performed concerning the material on the blade not yet removed and, thus, collisions with this stock material are possible (see Figure 7.4). In the future version of the algorithm the feasible region must be restrained by considering also the exceeding part material. The collision verification should be performed in two distinct steps:

- Collision verification between the **tool** (body and cutting surface) and the **part body** (current verification of the algorithm).

- Collision verification between the **tool body** and the **stock material** not yet removed (additional verification).

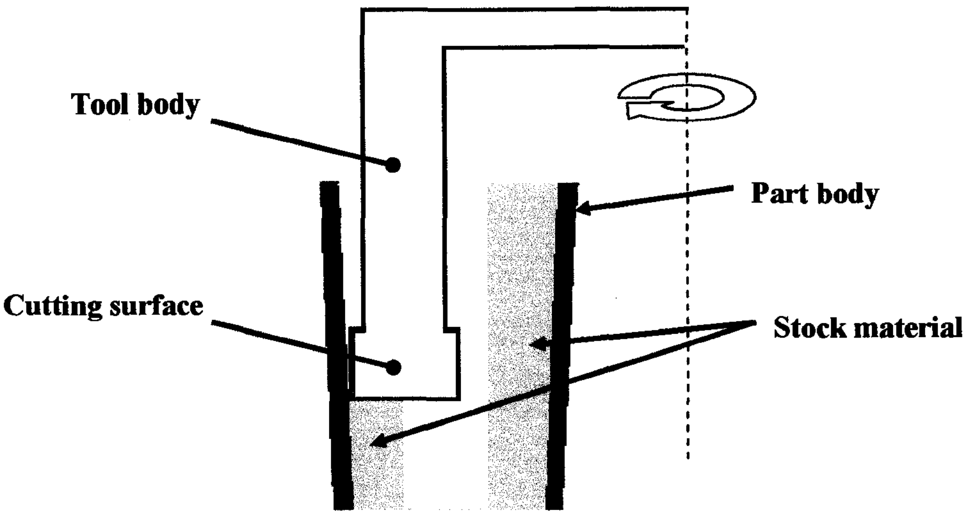


Fig. 7.4 Verification of collisions between the tool body and the stock

The tool path is computed from the top of the blade moving to the bottom (like the direction of machining). After a tool position (at each level) was identified, the stock material is "updated", i.e. the machined portion, corresponding to the material machined by the current tool position, is (virtually) removed from the stock material. The next position of the tool is compared for collisions with the remaining stock material. The Main program presented in Chapter 6 remains the same, with the exception that the intersection verification is performed with respect to two elements: the stock material and the part body and there is an extra "updating" operation of the stock material after each tool position is found.

New tool path smoothing strategy

Currently, the least squares method is used to obtain a smooth tool path. The inconvenient of this method is that the model function is not passing exactly by the tool positions; rather, the errors between the model function and the data points are minimized. Due to the large tool diameter, a small error between the smooth-out positions and the original positions can have a major impact at the cutting region. In most of the (virtual) tests performed with the algorithm, these errors were so small that no problems were observed. But, in other cases, however, the errors, although small, were unacceptable and the tool was colliding with the part after the tool path was smooth-out.

A solution to this problem is to use an "exact" interpolation method. A popular method that could be used in this case is the **cubic spline** method. Each successive points of the tool path are approximated by piecewise-polynomials (cubic). Each polynomial is passing exactly by the successive pair of nodes. Conditions are imposed to each polynomial to ensure that the interpolant is not only differentiable along the whole tool path, but also has a continuous second derivative.

REFERENCES

[1] BEDI, S. and S. MANN, *Flank Milling with Flat Endmilling Cutters,* Computer-Aided Design, 35 (2001), pp. 293-300.

[2] BURDEN, R. L. and J. D. FAIRES, *Numerical Analysis*, Brooks/Cole, 7th Edition, 2001.

[3] CHRISTMAN, A., *CATIA V5 CAM from IBM/Dassault Systeme – A Product Review, CIMdata Report, 2008.*

[4] EZUGWU, E. O., *High Speed Machining of Aero-Engine Alloys*, Journal of the Brazilian Society Mechanical Sciences and Engineering, 2004, Vol. XXVI, No. 1

[5] GRAY, P.; S. BEDI, F. ISMAIL and N. RAO, *Comparison of 5-Axis and 3-Axis Finish Machining of Hydroforming Die Inserts,* Int J Adv Manuf Technol, 17 (2001), pp. 562-569.

[6] HUNT, A. I., *A method and apparatus for producing an integrally bladed rotor,* Int. Cl. B23C 3/18, United Kingdom patent.

[7] LI, C.; S. MANN and S. BEDI, *Flank milling of ruled surface with conical tools – an optimization approach*, Int J Adv Manuf Technol, 29 (2006), pp. 1115-1128.

[8] MENZEL, C.; S BEDI and S. MANN, *Triple tangent milling of ruled surfaces,* Computer-Aided Design, 36 (2004), pp. 289-298.

[9] NOCEDAL, J. and S. J. WRIGHT, *Numerical Optimization*, Springer, New York, NY, 1999.

[10] PETEAN, M., *Roughing integrally bladed rotors using cup mill cutters method, cutter and program development*, Master's thesis, École Polytechnique de Montréal, 2004.

[11] SARAVANAMUTTO, H.I.H.; C.F.C. ROGERS and H. COHEN, *Gas Turbine Theory*, Pearson Education, 5th Edition, 2001.

[12] TONSHOFF, H. K.; C. GEY and N. RACKOW, *Flank Milling Optimization – The Flamingo Project*, Air & Space Europe, Vol. 3 (2001), No 3/8.

[13] WARKENTIN, A.; F. ISMAIL, and S. BEDI, *Comparison between multi-point and other 5-axis tool positioning strategies,* International Journal of Machine Tools & Manufacture, 40 (2000), pp. 185-208.

[14] WENLIANG, C.; Z. JIANJIANG, L. LEI and D. QIULIN, *An approach to gouging avoidance for sculptured surface machining,* Journal of Material Processing Technologies, 138 (2003), pp. 458-460.

[15] WU, C. Y, *Arbitrary Surface Flank Milling of Fans, Compressor, and Impeller Blades*, Journal of Engineering for Gas Turbine and Power, Vol. 117 (1995), pp. 534-539.

[16] YAMAMOTO, K.; H. OMOKAWA, M KURODA and K. ITAKURA, *High-Efficiency Cutting of Super-Heat-Resistant Alloy*, Mitsubishi Heavy Industries, Ltd. Technical Review, Vol. 38 (Feb. 2001), No. 1.

[17] _________, *Key to Understand and Comparing 3D File Format and their application to 3D Publishing*, Lattice3D Technical Division, Mars 2008.

[18] IOAN SASU, Private conversations.

APPENDIXES

Appendix A

Optimization algorithms

For this new algorithm, we require to maximize the volume of material removed. In that scope, different optimizations algorithms are studied.

Optimization algorithms are iterative. They begin with an initial guess of the optimal value(s) of the variable(s) and generate a sequence of improved estimates, until the solution is found. What distinguish one algorithm from another is the strategy used to move from one iterate to the next. Most algorithms make use of the values of the objective function *f*, the constraints *c*, and possibly the first and second derivative of these functions.

In this our case, two optimization algorithms were studied: the **Newton's method** and a derivate of this method, the **BFGS method** (which is a "Quasi-Newton method").

A.1 Newton method

For finding the optimum of a function, the most powerful is the Newton method. The algorithm of the Newton method is the following:

```
Given: Starting point $x_0$ , tolerance $\varepsilon > 0$
$k \leftarrow 0$;
While $\|\nabla f_k\| > \varepsilon$
    $d_k = -\nabla^2 f_k^{-1} \nabla f_k$
    $x_{k+1} = x_k + \alpha_k d_k$
    $k \leftarrow k + 1$
end (while)
```

Algorithm A.1 Newton's method [9]

where: $\nabla^2 f$ is the Hessian matrix of the function f.

∇f is the gradient of f

d_k is the search direction

α_k is the step length and is computed from the line search procedure to satisfy the Wolf condition (see section 3.4.3).

The Newton method is very powerful due to its quadratic convergence (error is squared at each step). There is some draw backs to this method however:

- The method requires the gradient vector as well as the Hessian matrix of the function to be calculated as each step(manual two times differentiation of the objective function);
- If the initial guess is to far from the solution, the method might fail to converge.

A.2 BFGS Method (Broyden, Fletcher, Goldfarb, and Shanno)

Other methods derived of the Newton's method are called Quasi-Newton methods. These methods do not require the computation of the second derivative of the objective function by hand and an approximation is used instead. The algorithm of the BFGS method is the following:

```
Given: Starting point $x_0$ , tolerance $\varepsilon > 0$
$k \leftarrow 0$;

While $\|\nabla f_k\| > \varepsilon$

        $d_k = -B_k \nabla f_k$

        $x_{k+1} = x_k + \alpha_k d_k$

        $k \leftarrow k + 1$

end (while)
```

Algorithm A.2 BFGS method [9]

where: ∇f is the gradient of f

d_k is the search direction

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \qquad \text{(BFGS)}$$

$s_k = x_{k+1} - x_k$

$y_k = \nabla f_{k+1} - \nabla f_k$

α_k is the step length and is computed from the line search procedure to satisfy the

Wolf condition (see section 3.4.3).

(Note that the fists Hessian approximation, B_0, is usually taken to be the *identity* matrix, or multiples of the identity matrix.)

A.3 Wolf (Armijo) condition

The vector d_k indicates the decreasing direction of the function f at the current x_k location. However, an appropriate step length α_k in this direction has to be calculated to have a substantial reduction of f. Algorithms that try to find suitable values for α_k are called *line search* algorithms.

An intuitive condition would be to impose on α_k is that it provides a reduction in f, i.e. $f(x_k + \alpha_k d_k) < f(x_k)$. This is not an appropriate strategy because the reduction in f might not be sufficient at each step and the algorithm might take a tremendous amount of time to converge to the solution or it might even not converge at all (might stall).

A popular line search condition stipulates that α_k should first of all give *sufficient decrease* in the objective function f, as measured by the following inequality:

$$f(x_k + \alpha\, d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k, \text{ where } c_1 \in (0,1) \qquad (3.1)$$

This inequality is known as the Armijo condition (known also as the Wolf condition). The sufficient decrease condition state that α is acceptable only if the Armijo condition is respected. In practice, c_1 is selected to be quite small, say $c_1 = 10^{-6.}$

The line search algorithm for determining an appropriate step length α_k proceeds as follows:

```
Choose $\bar{\alpha} > 0$, $c_1 \in (0,1)$, $\rho$; set $\alpha \leftarrow \bar{\alpha}$

Repeat until $f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k$

        $\alpha \leftarrow \rho \alpha$

end (repeat)

Terminate with $\alpha_k \leftarrow \alpha$
```

Algorithm A.3 Line Search – Armijo condition

Starting the line search, the initial step length $\bar{\alpha}$ is chosen large enough (typically 1). Each time, if the condition is not respected, the step length is reduces by a factor of ρ ($0 < \rho < 1$). An acceptable step length will be found after a finite number of trials because α_k will eventually become small enough that the sufficient decrease condition holds.

Appendix B

Algorithms

B.1 Main Program

```
Public referenceAs(1) As Reference
Public TheMeasurable
Public referenceAss(1) As Reference
Public referenceT(8) As Reference
Public MinimumDistance As Double
Public MinimumDistancecheck As Double
Public MinimumDistanceR As Double
Public refl As Reference
Public toward As Boolean
Public pi As Double
Public referencePcheck As Reference
Public referenceO2 As Reference
Public referenceO1 As Reference
Public referenceO3 As Reference
Public cv As Integer
Public fin As Boolean
Public numb As Integer
Public PScheck As Boolean
Public SScheck As Boolean
Public SScheck2 As Boolean
Public HUBcheck As Boolean
Public trans(1) As HybridShapeTranslate
Public mycoord(5)
Public myResule
Public grad(2) As Double
Public Mgrad As Double
Public referenceMinPt As Reference
Public referenceMinPt2 As Reference
Public hybridShapeRotateR(1, 2) As HybridShapeRotate
Public referenceRot(1, 2) As Reference
Public referenceLR(2) As Reference
Public counterr As Integer
Public vect()
Public referenceAx As Reference
Public firstpointt As Boolean
Public referenceEj As Reference
Public refX(1) As Reference
Public xsect(1) As Boolean
```

```
Sub CATMain()
'30J2429-01'

Dim documents1 As Documents
   Set documents1 = CATIA.Documents

Dim partDocument1 As PartDocument
   Set partDocument1 = documents1.Open("F:\Projet\newdev\NEXT.CATPart")

****************************GENERALITIES***************************
Dim TheSPAWorkbench As SPAWorkbench
   Set TheSPAWorkbench =
CATIA.ActiveDocument.GetWorkbench("SPAWorkbench")

Dim productDocument1 As Document
   Set productDocument1 = CATIA.ActiveDocument

Dim part1 As Part
   Set part1 = partDocument1.Part

Dim hybridShapeFactory1 As HybridShapeFactory
   Set hybridShapeFactory1 = part1.HybridShapeFactory

Dim shapeFactory1 As ShapeFactory
   Set shapeFactory1 = part1.ShapeFactory

Dim hybridBodies1 As HybridBodies
   Set hybridBodies1 = part1.HybridBodies

Dim selection1 As Selection
   Set selection1 = productDocument1.Selection
**********************************************************************


****************************GEOMETRICAL*SETS**********************
Dim hybridBody1 As HybridBody
   Set hybridBody1 = hybridBodies1.Item("Geometrical Set.1")

Dim hybridShapes1 As HybridShapes
   Set hybridShapes1 = hybridBody1.HybridShapes
```

```
Dim hybridBody5 As HybridBody
   Set hybridBody5 = hybridBodies1.Add()          '(geometrical set.3)
*************************************************************************




********************CONSTRUCTION*FROM*SET*1********************
Dim hybridShapePlaneOffsetPt1 As HybridShapePlaneOffsetPt
   Set hybridShapePlaneOffsetPt1 = hybridShapes1.Item("Plane.8")
   Dim reference1 As Reference
   Set reference1 = part1.CreateReferenceFromObject(hybridShapePlaneOffsetPt1)
'-----------------------------------------------------------------------------------------
Dim hybridShapeAssemble1 As HybridShapeAssemble
   Set hybridShapeAssemble1 = hybridShapes1.Item("Join.11")
Dim reference3 As Reference
   Set reference3 = part1.CreateReferenceFromObject(hybridShapeAssemble1)
'-----------------------------------------------------------------------------------------
Dim hybridShapeBoundary1 As HybridShapeBoundary
Dim hybridShapeBoundary2 As HybridShapeBoundary
Set hybridShapeBoundary1 = hybridShapes1.Item("Boundary.4")
Set hybridShapeBoundary2 = hybridShapes1.Item("Boundary.1")
Set referenceN2 = part1.CreateReferenceFromObject(hybridShapeBoundary1)
Set referenceN1 = part1.CreateReferenceFromObject(hybridShapeBoundary2)
'-----------------------------------------------------------------------------------------
Dim hybridShapeAssembleIB(1) As HybridShapeAssemble
   Set hybridShapeAssembleIB(0) = hybridShapes1.Item("Join.16")
Dim referenceIB(1) As Reference
   Set referenceIB(0) = part1.CreateReferenceFromObject(hybridShapeAssembleIB(0))
   Set hybridShapeAssembleIB(1) = hybridShapes1.Item("Join.17")
   Set referenceIB(1) = part1.CreateReferenceFromObject(hybridShapeAssembleIB(1))
'-----------------------------------------------------------------------------------------
Dim hybridShapeIntersection7 As hybridShapeIntersection
   Set hybridShapeIntersection7 = hybridShapes1.Item("Intersect.7")
Dim reference7 As Reference
   Set reference7 = part1.CreateReferenceFromObject(hybridShapeIntersection7)
'-----------------------------------------------------------------------------------------
Dim hybridShapeOffset1 As HybridShapeOffset
   Set hybridShapeOffset1 = hybridShapes1.Item("Offset.1")
   Set referenceO2 = part1.CreateReferenceFromObject(hybridShapeOffset1)
Dim hybridShapeOffset2 As HybridShapeOffset
   Set hybridShapeOffset2 = hybridShapes1.Item("Offset.2")
```

```
Dim referenceO1 As Reference
   Set referenceO1 = part1.CreateReferenceFromObject(hybridShapeOffset2)
Dim hybridShapeOffset3 As HybridShapeOffset
   Set hybridShapeOffset3 = hybridShapes1.Item("Offset.3")
Dim referenceO3 As Reference
   Set referenceO3 = part1.CreateReferenceFromObject(hybridShapeOffset3)
'-----------------------------------------------------------------------------------------
Dim referenceS As Reference
   Set referenceS =
part1.CreateReferenceFromBRepName("BorderREdge:(BEdge:(Brp:(GSMOffset.1;(Brp:
(FeatureRSUR.12;(Brp:(Solide.1 ( *SOL1 - wsp *MASTER -  ) ( Brep from *SOL1 -
wsp *MASTER -  );122);Brp:(Solide.1 ( *SOL1 - wsp *MASTER -  ) ( Brep from
*SOL1 - wsp *MASTER -
);123)))));None:(Limits1:();Limits2:();+1);Cf9:());WithPermanentBody;WithoutBuildErr
or;WithSelectingFeatureSupport;MFBRepVersion_CXR14)", hybridShapeOffset1)
'-----------------------------------------------------------------------------------------
Dim hybridShapeSplit1 As HybridShapeSplit
   Set hybridShapeSplit1 = hybridShapes1.Item("Split.5")
'-----------------------------------------------------------------------------------------
Dim hybridShapeExtremum1 As HybridShapeExtremum
   Set hybridShapeExtremum1 = hybridShapes1.Item("Extremum.10")
'-----------------------------------------------------------------------------------------
Dim hybridShapeLinePtDir1 As HybridShapeLinePtDir
   Set hybridShapeLinePtDir1 = hybridShapes1.Item("DIRECTION_line")

Dim referenceD3 As Reference
   Set referenceD3 = part1.CreateReferenceFromObject(hybridShapeLinePtDir1)

Dim hybridShapeDirectionD As HybridShapeDirection
   Set hybridShapeDirectionD = hybridShapeFactory1.AddNewDirection(referenceD3)
****************************************************************************


Counter = 0
ReDim vect(5, 0) 'As Double

Dim dir As Double
Dim aa As Double
Dim bb As Double


firstpointt = True
j = 31
```

```
*******************************CONSTANTS*************************
Dim step As Double
   step = 8.032 / 2
Dim ddf As Integer
Dim v  As Integer

Dim body2(1) As Body

Dim F As Double
Dim Fp As Double


pi = 4 * Atn(1)



Dim astep As Double
   astep = 0.2
Dim SS(2) As Integer
   SS(0) = 4: SS(1) = 4: SS(2) = 4
   numb = 0
   cv = 0
   fin = False
   PScheck = True
   HUBcheck = False
Dim red As Integer
red = 0
*******************************************************************




cv = 0
Dim hybridBody2 As HybridBody
   Set hybridBody2 = hybridBodies1.Add()
***********************PARALLEL*CURVES*GENERATION*************
If j <= 6 Then
   v = 3
   ddf = 3
Else
   v = 4
   Set referenceT(5) = part1.CreateReferenceFromObject(hybridShapeSplit1)
   Set referenceT(7) = part1.CreateReferenceFromObject(hybridShapeSplit1)
   Set referenceT(6) = part1.CreateReferenceFromObject(hybridShapeSplit1)
   Set referenceT(8) = part1.CreateReferenceFromObject(hybridShapeExtremum1)
   ddf = 8
End If
'-----------------------------------------------------------------------------------------
For I = 1 To (v)
Dim hybridShapeCurvePar1 As HybridShapeCurvePar
   Set hybridShapeCurvePar1 = hybridShapeFactory1.AddNewCurvePar(referenceS,
referenceO2, I * 9, False, False)

   hybridShapeCurvePar1.SmoothingType = 0

   hybridBody1.AppendHybridShape hybridShapeCurvePar1

   part1.InWorkObject = hybridShapeCurvePar1

   Set referenceT(I) = part1.CreateReferenceFromObject(hybridShapeCurvePar1)
part1.Update
Next I
**************************************************************************



************************DEFINING*THE*CUP*RADIUS********************
Dim cup_OD As Double
   cup_OD = 10.123

Dim cup_tickness As Double
   cup_thickness = 0.625

Dim radius As Double
   radius = ((cup_OD - cup_thickness) / 2) * 28.4
**************************************************************************

Open "PathPS" For Output As #1



*******CONSTRUCTION*TO*BRING*THE*CUP*TO*POSITION***************
Dim hybridShapePlaneOffset1 As HybridShapePlaneOffset
   Set hybridShapePlaneOffset1 = hybridShapeFactory1.AddNewPlaneOffset(reference1,
j * step, False)

   hybridBody8.AppendHybridShape hybridShapePlaneOffset1
   part1.InWorkObject = hybridShapePlaneOffset1

   Dim reference2 As Reference
   Set reference2 = part1.CreateReferenceFromObject(hybridShapePlaneOffset1)
part1.Update
'-----------------------------------------------------------------------------------------
Dim hybridShapeIntersection1 As hybridShapeIntersection
   Set hybridShapeIntersection1 = hybridShapeFactory1.AddNewIntersection(reference2,
reference3)

   hybridShapeIntersection1.PointType = 0

   hybridBody8.AppendHybridShape hybridShapeIntersection1

   part1.InWorkObject = hybridShapeIntersection1

   Dim referenceI As Reference
   Set referenceI = part1.CreateReferenceFromObject(hybridShapeIntersection1)
   part1.Update
part1.Update

'-----------------------------------------------------------------------------------------
Dim hybridShapeNear1 As HybridShapeNear
   Set hybridShapeNear1 = hybridShapeFactory1.AddNewNear(referenceI, referenceN1)

   hybridBody8.AppendHybridShape hybridShapeNear1

   part1.InWorkObject = hybridShapeNear1
   Dim referenceI1 As Reference
   Set referenceI1 = part1.CreateReferenceFromObject(hybridShapeNear1)
part1.Update

'-----------------------------------------------------------------------------------------
Dim hybridShapeNear2 As HybridShapeNear
   Set hybridShapeNear2 = hybridShapeFactory1.AddNewNear(referenceI, referenceN2)

   hybridBody8.AppendHybridShape hybridShapeNear2

   part1.InWorkObject = hybridShapeNear2
   Dim referenceI2 As Reference
   Set referenceI2 = part1.CreateReferenceFromObject(hybridShapeNear2)
part1.Update
'-----------------------------------------------------------------------------------------
```

```
Dim hybridShapeCircle2PointsRad1 As HybridShapeCircle2PointsRad
   Set hybridShapeCircle2PointsRad1 =
hybridShapeFactory1.AddNewCircle2PointsRad(referenceI1, referenceI2, reference2,
False, radius, -1)

   hybridShapeCircle2PointsRad1.SetLimitation 1

   hybridBody8.AppendHybridShape hybridShapeCircle2PointsRad1

   part1.InWorkObject = hybridShapeCircle2PointsRad1

   Dim referenceC1 As Reference
   Set referenceC1 = part1.CreateReferenceFromObject(hybridShapeCircle2PointsRad1)
part1.Update

'-----------------------------------------------------------------------------------------
Dim hybridShapeCircle2PointsRad2 As HybridShapeCircle2PointsRad
   Set hybridShapeCircle2PointsRad2 =
hybridShapeFactory1.AddNewCircle2PointsRad(referenceI1, referenceI2, reference2,
False, radius - 1, -1)

   hybridShapeCircle2PointsRad8.SetLimitation 1

   hybridBody8.AppendHybridShape hybridShapeCircle2PointsRad2

   part1.InWorkObject = hybridShapeCircle2PointsRad2

   Dim referenceC2 As Reference
   Set referenceC2 = part1.CreateReferenceFromObject(hybridShapeCircle2PointsRad2)
part1.Update
'-----------------------------------------------------------------------------------------
Dim hybridShapePointCenter1 As HybridShapePointCenter
   Set hybridShapePointCenter1 =
hybridShapeFactory1.AddNewPointCenter(referenceC1)

   hybridBody8.AppendHybridShape hybridShapePointCenter1

   part1.InWorkObject = hybridShapePointCenter1
   Dim referencePC1 As Reference
   Set referencePC1 = part1.CreateReferenceFromObject(hybridShapePointCenter1)
part1.Update
'-----------------------------------------------------------------------------------------
Dim hybridShapePointCenter2 As HybridShapePointCenter
```

```
   Set hybridShapePointCenter2 =
hybridShapeFactory1.AddNewPointCenter(referenceC2)

   hybridBody8.AppendHybridShape hybridShapePointCenter2

   part1.InWorkObject = hybridShapePointCenter2
   Dim referencePC2 As Reference
   Set referencePC2 = part1.CreateReferenceFromObject(hybridShapePointCenter2)
part1.Update
'-----------------------------------------------------------------------------------------
Dim hybridShapeLinePtPtD As HybridShapeLinePtPt
   Set hybridShapeLinePtPtD = hybridShapeFactory1.AddNewLinePtPt(referencePC1,
referencePC2)

   hybridBody8.AppendHybridShape hybridShapeLinePtPtD

   part1.InWorkObject = hybridShapeLinePtPtD

   Dim referenceDr1 As Reference
   Set referenceDr1 = part1.CreateReferenceFromObject(hybridShapeLinePtPtD)

   Dim hybridShapeDirectionDr1 As HybridShapeDirection
   Set hybridShapeDirectionDr1 =
hybridShapeFactory1.AddNewDirection(referenceDr1)
part1.Update
'-----------------------------------------------------------------------------------------
Dim hybridShapeTranslateP As HybridShapeTranslate
   Set hybridShapeTranslateP = hybridShapeFactory1.AddNewEmptyTranslate()

   hybridShapeTranslateP.ElemToTranslate = referenceI2

   hybridShapeTranslateP.VectorType = 0

   hybridShapeTranslateP.Direction = hybridShapeDirectionDr1

   hybridShapeTranslateP.DistanceValue = 28.4

   hybridShapeTranslateP.VolumeResult = False

   hybridBody8.AppendHybridShape hybridShapeTranslateP

   part1.InWorkObject = hybridShapeTranslateP
   Set referencePcheck = part1.CreateReferenceFromObject(hybridShapeTranslateP)
part1.Update
```

```
'-----------------------------------------------------------------------------------------
Dim hybridShapeTranslate1(1) As HybridShapeTranslate
For s = 0 To 1
   Set hybridShapeTranslate1(s) = hybridShapeFactory1.AddNewEmptyTranslate()

   hybridShapeTranslate1(s).ElemToTranslate = referenceIB(s)
   hybridShapeTranslate1(s).VectorType = 1

   hybridShapeTranslate1(s).FirstPoint = reference7

   hybridShapeTranslate1(s).SecondPoint = referencePC1

   hybridShapeTranslate1(s).VolumeResult = False

   hybridBody8.AppendHybridShape hybridShapeTranslate1(s)

   part1.InWorkObject = hybridShapeTranslate1(s)
   Set referenceAs(s) = part1.CreateReferenceFromObject(hybridShapeTranslate1(s))
part1.Update
Next s
'-----------------------------------------------------------------------------------------
Dim hybridShapeTranslate2(1) As HybridShapeTranslate
For s = 0 To 1
   Set hybridShapeTranslate2(s) = hybridShapeFactory1.AddNewEmptyTranslate()

   hybridShapeTranslate2(s).ElemToTranslate = referenceAs(s)

   hybridShapeTranslate2(s).VectorType = 0

   hybridShapeTranslate2(s).Direction = hybridShapeDirectionDr1

   hybridShapeTranslate2(s).DistanceValue = 0#

   hybridShapeTranslate2(s).VolumeResult = False

   hybridBody8.AppendHybridShape hybridShapeTranslate2(s)

   part1.InWorkObject = hybridShapeTranslate2(s)
   Set referenceAs(s) = part1.CreateReferenceFromObject(hybridShapeTranslate2(s))
   Set trans(s) = hybridShapeTranslate2(s)
part1.Update
Next s
************************************************************************
```

```
startt:

'#########################################
Call Create_Body(part1, shapeFactory1, body2) '#
'#########################################
Set TheMeasurable = TheSPAWorkbench.GetMeasurable(referenceAss(0))


******************DECISION*FOR*TRANSLATE*************************
   MinimumDistance = TheMeasurable.GetMinimumDistance(referenceO2)
   MinimumDistancecheck = TheMeasurable.GetMinimumDistance(referencePcheck)
   MinimumDistanceR = Round(MinimumDistance, 8)
'--------------------------------------------------------------------------------------------------------------
Set ref1 = referenceO2
If MinimumDistanceR = 0 Then
   toward = False
Else
   toward = True
End If
********************************************************************


'####################################
Call translate1(part1, productDocument1) '#
'####################################
selection1.Clear
selection1.Add body2(0)
selection1.Add body2(1)
selection1.Delete


Set referenceAs(0) = part1.CreateReferenceFromObject(hybridShapeTranslate2(0))

Do

aaa:
Set TheMeasurable = TheSPAWorkbench.GetMeasurable(referenceAs(1))

Set ref1 = referenceO2
'#########################################################################
Call rotation_axes(part1, hybridShapeFactory1, hybridShapeDirection1, hybridBody2,
TheSPAWorkbench)
'#########################################################################
```

```
'########################################
Call Create_Body(part1, shapeFactory1, body2) '#
'################ev######################

Set TheMeasurable = TheSPAWorkbench.GetMeasurable(referenceAss(1))


'###########################################
Call Function_eval(F, ddf, Fp, TheSPAWorkbench) '#
'###########################################
F = Fp


Dim ang(1, 2)
Dim Rot(1, 2) As Parameter
For s = 0 To 1
For I = 0 To 2
   ang(s, I) = hybridShapeRotateR(s, I).Angle.Value
   Set Rot(s, I) = hybridShapeRotateR(s, I).Angle
Next I
Next s

Dim param_dist(1) As Parameter
Dim distt(1) As Double
For s = 0 To 1
   Set param_dist(s) = trans(s).Distance
   distt(s) = (param_dist(s).Value) / 1000
Next s

Dim test As Double

If numb = 3 And fin = True Then
   GoTo stopp
End If
'######################################################################
Call gradfinding(part1, F, ddf, Fp, ang, Rot, r, SS, TheSPAWorkbench, referenceO1)  '#
'######################################################################
dir = -Mgrad

aa = 2
bb = 1
```

```
'AAAAAAAAAAAAAAAA**ARMIJO**AAAAAAAAAAAAAAAAAAAAAAAAA
While aa >= bb
   For s = 0 To 1
   Rot(s, r).ValuateFromString (ang(s, r) + astep * dir)
   part1.Update
   test = Abs((ang(s, r) + astep * dir) * 360 / 2 / pi)
   Next s

   If test > 4 And firstpointt = False Then
      Rot(0, r).ValuateFromString (0)
      Rot(1, r).ValuateFromString (0)
      param_dist(0).ValuateFromString (distt(0))
      param_dist(1).ValuateFromString (distt(1))
      part1.Update
      GoTo nextt
   End If

   If test < 0.05 Then
   cv = cv + 1
      Rot(0, r).ValuateFromString (0)
      Rot(1, r).ValuateFromString (0)
      param_dist(0).ValuateFromString (distt(0))
      param_dist(1).ValuateFromString (distt(1))
   part1.Update
   GoTo stoppl
   End If

******************DECISION*FOR*TRANSLATE*****************
   MinimumDistance = TheMeasurable.GetMinimumDistance(referenceO2)
   MinimumDistancecheck = TheMeasurable.GetMinimumDistance(referencePcheck)
   MinimumDistanceR = Round(MinimumDistance, 8)
   Set ref1 = referenceO2
   '-----------------------------------------------------------------------------------
   If MinimumDistance < 0.1016 And MinimumDistanceR <> 0 Then
      GoTo nextt
   End If
   '-----------------------------------------------------------------------------------
   If MinimumDistanceR = 0 Then
      toward = False
   Else
      toward = True
   End If

**********************************************************************
```

```
  '########################################
      Call translate1(part1, productDocument1) '#
  '########################################
'XXXXXXXXXXXXXXXXXX
nextt:

Set refX(0) = reference01
Set refX(1) = reference03


Call xsection_eval1(part1, hybridBody2, hybridShapeFactory1, selection1)

If xsect(0) = False Or xsect(1) = False Then

    Rot(0, r).ValuateFromString (0)
    Rot(1, r).ValuateFromString (0)
    param_dist(0).ValuateFromString (distt(0))
    param_dist(1).ValuateFromString (distt(1))
    part1.Update
    aa = 2: bb = 1
    astep = astep * 0.5
    GoTo nextt2
End If



'XXXXXXXXXXXXXXXXXXX

  '############################################
  Call Function_eval(F, ddf, Fp, TheSPAWorkbench) '#
  '############################################

  aa = Fp
  bb = F + 0.1 * astep * (grad(r) * dir)
  astep = astep / 2

nextt2:
 Wend
'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

stopp1:

  astep = 0.2
```

```
If cv = 1 And firstpointt = False Then
   numb = 1
   SS(0) = 2
   SS(1) = 4
   SS(2) = 4
   cv = 2
End If


selection1.Clear
selection1.Add body2(0)
selection1.Add body2(1)
selection1.Delete


Loop Until cv >= 4


stopp:



'##########################################################################
Call pointandline_gen(hybridShapeFactory1, part1, hybridBody2, hybridBody5,
hybridShapeLinePtPt1, selection1, TheSPAWorkbench, hybridShapeAxisLine1,
hybridShapeDirection1, hybridShapeExtremum1, hybridShapePointCoord2)
'##########################################################################
firstpointt = False


   Set hybridBody2 = hybridBodies1.Add()
For s = 0 To 1
Set hybridShapeTranslate2(s) = hybridShapeFactory1.AddNewEmptyTranslate()
   hybridShapeTranslate2(s).ElemToTranslate = referenceAs(s)

   hybridShapeTranslate2(s).VectorType = 0

   hybridShapeTranslate2(s).Direction = hybridShapeDirectionD

   hybridShapeTranslate2(s).DistanceValue = -0.762

   hybridShapeTranslate2(s).VolumeResult = False

   hybridBody8.AppendHybridShape hybridShapeTranslate2(s)
   part1.InWorkObject = hybridShapeTranslate2(s)
   Set referenceAs(s) = part1.CreateReferenceFromObject(hybridShapeTranslate2(s))
part1.Update


Set hybridShapeTranslate2(s) = hybridShapeFactory1.AddNewEmptyTranslate()
   hybridShapeTranslate2(s).ElemToTranslate = referenceAs(s)

   hybridShapeTranslate2(s).VectorType = 0

   hybridShapeTranslate2(s).Direction = hybridShapeDirectionDr1

   hybridShapeTranslate2(s).DistanceValue = 0#

   hybridShapeTranslate2(s).VolumeResult = False

   hybridBody8.AppendHybridShape hybridShapeTranslate2(s)

   part1.InWorkObject = hybridShapeTranslate2(s)
   Set referenceAs(s) = part1.CreateReferenceFromObject(hybridShapeTranslate2(s))
   Set trans(s) = hybridShapeTranslate2(s)
part1.Update
Next s

   SS(0) = 0: SS(1) = 1: SS(2) = 4
   numb = 2
   cv = 0
   fin = False

Write #1, vect(0, counterr - 1), vect(1, counterr - 1), vect(2, counterr - 1), vect(3, counterr
- 1), vect(4, counterr - 1), vect(5, counterr - 1)
'Close #1

If red >= 20 Then
   red = 0
End If

red = red + 1

GoTo startt
Close #1

End Sub
```

B.2 Translate Sub-program

```
'####################################
Sub translate1(part1, productDocument1) '#
'####################################



Dim d As Integer
d = 1


Dim OK As Boolean
OK = True

Dim param_dist As Parameter
Set param_dist = trans(0).Distance


Dim check As Double
check = MinimumDistancecheck

Dim a As Integer
a = 0

ReDim dist(a) 'As Double

'*******************************INITIAL*STEP***************************
Dim inc As Double
If toward = True Then
   inc = MinimumDistance
Else
   inc = 8.08
End If
'************************************************************************
dist(a) = param_dist.Value

Dim k As Integer
k = 0

Dim b As Double

Do
beg:
```

```
a = a + 1

ReDim Preserve dist(a)

'*********************************TRANSLATE*VALUE**********************
dist(a) = dist(a - 1) + d * inc
b = dist(a) / 1000
param_dist.ValuateFromString (b)
part1.Update
'****************************************************************************


'*******************************MEASUREMENT****************************
****************************
MinimumDistancecheck = TheMeasurable.GetMinimumDistance(referencePcheck)
'*
MinimumDistance = TheMeasurable.GetMinimumDistance(ref1)
MinimumDistanceR = Round(MinimumDistance, 3)
'****************************************************************************
************************


'***************************STOPPING*CONDITION*************************
  If MinimumDistance < 0.0009 And MinimumDistanceR <> 0 Then
    GoTo end2
  End If
'****************************************************************************

'***************************GOOD*DIRECTION*FIND***********************
  If MinimumDistancecheck > check And toward = True And OK = True Then
    a = a - 1
    d = -d
    OK = False
    GoTo beg
  End If
'-----------------------------------------------------------------------------------------------
  If MinimumDistancecheck < check And toward = False And OK = True Then
    a = a - 1
    d = -d
    OK = False
    GoTo beg
  End If
'****************************************************************************
OK = False
```

```
'************************************************************************
  If MinimumDistanceR <> 0 And toward = False Then
    toward = True
    d = -d
  End If
'-----------------------------------------------------------------------------------------
  If toward = True Then
    inc = MinimumDistance - 0.000254
  End If
'************************************************************************
If MinimumDistance > 10 Then
d = -d
End If

Loop Until (MinimumDistance < 0.00254) And (toward = True)

end2:

Set param_dist = trans(0).Distance
dist(0) = param_dist.Value
Set param_dist = trans(1).Distance
b = dist(0) / 1000
param_dist.ValuateFromString (b)
part1.Update


End Sub
```

B.3 Gradient Sub-program

```
'##########################################################################
Sub gradfinding(part1, F, ddf, Fp, ang, Rot, r, SS, TheSPAWorkbench, referenceO1) '#
'##########################################################################

Mgrad = 0
Dim dda As Double
dda = 0.1
Dim Modd As Double

For I = 0 To 2
   For s = 0 To 1
   Rot(s, I).ValuateFromString (2 * pi * (dda + ang(s, I)) / 360)
   part1.Update
   Next s
Set TheMeasurable = TheSPAWorkbench.GetMeasurable(referenceAss(0))

'********************************DECISION*FOR*TRANSLATE************
   MinimumDistance = TheMeasurable.GetMinimumDistance(ref1)
   MinimumDistancecheck = TheMeasurable.GetMinimumDistance(referencePcheck)
   MinimumDistanceR = Round(MinimumDistance, 8)
'-------------------------------------------------------------------------------------
If MinimumDistanceR = 0 Then
   toward = False
Else
   toward = True
End If
'***********************************************************************
   '#####################################
   Call translate1(part1, productDocument1) '#
   '#####################################
Set TheMeasurable = TheSPAWorkbench.GetMeasurable(referenceAss(1))

   If PScheck = True Then
   '##############################################
   Call Function_eval(F, ddf, Fp, TheSPAWorkbench) '#
   '##############################################
   End If

   If HUBcheck = True Then
   '#############################
   Call Function_eval_Hub(F, Fp) '#
   '############################
   End If


   If SScheck2 = True Then
   '#############################################
   Call Function_eval_SS2(F, Fp, TheSPAWorkbench)  '#
   '#############################################
   End If

   grad(I) = (Fp - F) / dda

   For s = 0 To 1
      Rot(s, I).ValuateFromString (2 * pi * ang(s, I) / 360)
      part1.Update
   Next s

   If Abs(Mgrad) < Abs(grad(I)) And I <> SS(0) And I <> SS(1) Then

      Mgrad = grad(I)
      r = I

   End If

Next I

Modd = Sqr(grad(0) ^ 2 + grad(1) ^ 2 + grad(2) ^ 2)
Mgrad = Mgrad / Modd


SS(numb) = r
numb = numb + 1

If numb = 3 And fin = False And PScheck = True Then
   numb = 0
   SS(0) = 4
   SS(1) = 4
   SS(2) = 4
End If

If cv = 0 And fin = False And PScheck = True And firstpointt = False Then
   numb = 2
   SS(0) = 0
   SS(1) = 1
   SS(2) = 4
End If

If cv = 0 And fin = False And HUBcheck = True Then
   numb = 2
   SS(0) = 1
   SS(1) = 2
   SS(2) = 4
End If


If numb = 3 And fin = False And HUBcheck = True Then
   numb = 2
   SS(0) = 1
   SS(1) = 2
   SS(2) = 4
End If


If numb = 3 And fin = False And SScheck2 = True Then
   numb = 0
   SS(0) = 4
   SS(1) = 4
   SS(2) = 4
End If

End Sub
```

B.4 Intersection evaluation sub-program

```
'###############################################################
Sub xsection_eval1(part1, hybridBody2, hybridShapeFactory1, selection1) '#
'##############################################################


Dim hybridShapeIntersection(1) As hybridShapeIntersection

For s = 0 To 1
  xsect(s) = False
  Set hybridShapeIntersection(s) =
hybridShapeFactory1.AddNewIntersection(referenceAss(0), refX(s))

    hybridShapeIntersection(s).PointType = 0

    hybridBody8.AppendHybridShape hybridShapeIntersection(s)

    part1.InWorkObject = hybridShapeIntersection(s)

  On Error GoTo ErrorHandler
  part1.Update


  If xsect(s) = False Then
  selection1.Clear
  selection1.Add hybridShapeIntersection(s)
  selection1.Delete
  End If

Next s

GoTo forward:


ErrorHandler:

selection1.Clear
selection1.Add hybridShapeIntersection(s)
selection1.Delete
xsect(s) = True
Resume
```

```
forward:

End Sub
```