

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PROJET D'APPLICATION PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE MÉCANIQUE
M. ING

PAR
RABAH MIHOUBI

DÉVELOPPEMENT D'UN SIMULATEUR D'UNE
MACHINE-OUTIL À TROIS AXES

MONTRÉAL/AOUT 2000

© droits réservés de Rabah Mihoubi



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-56061-9

Canada

CE PROJET D'APPLICATION A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

- M. Jean François Chatelain, professeur-tuteur.
Département de génie mécanique à l'École de technologie supérieure.
- M. Roland Maranzana, professeur
Département de génie de la production automatisée à l'École de technologie supérieure.
- M. Patrick Terriault, professeur.
Département de génie mécanique à l'École de technologie supérieure.

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT JURY ET UN PUBLIC

LE 04 AOUT 2000

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

DÉVELOPPEMENT D'UN SIMULATEUR D'UNE MACHINE-OUTIL À TROIS AXES

Rabah Mihoubi

(Sommaire)

La plupart des champs d'activités techniques aujourd'hui utilisent la simulation comme outil de prédiction et de vérification. Dans le domaine de l'usinage, cette technique trouve une application très adéquate. L'objectif principal de ce projet est le développement et la validation d'un simulateur pour une fraiseuse à trois axes à l'aide du logiciel Vericut. Les grandes étapes de développement sont détaillées, soit la modélisation solide des composantes et leur assemblage, la caractérisation cinématique de la machine ainsi que la modélisation du contrôleur FANUC OM équipant celle-ci.

Les essais de validation effectués sont concluants et permettent d'illustrer la robustesse du simulateur. Il est démontré que le simulateur permet notamment de détecter avec fiabilité les collisions possibles entre les outils de coupe et les composantes du système pièce/machine ainsi que certains types d'erreurs dans les trajectoires d'outils. Il est également montré que le simulateur détecte parfaitement les dépassements dans la course permise pour chacun des axes de la machine. D'autre part, suite aux essais des modules d'analyse des modèles et d'optimisation des trajectoires, certaines limitations du logiciel sont mises en relief. En effet, ces derniers estiment les erreurs de coupe et suggèrent des paramètres optimaux à partir de données purement théoriques sans tenir compte de l'environnement physique et des déflexions d'outils inhérentes à tout processus d'usinage.

Malgré ces dernières caractéristiques optionnelles à utiliser avec discernement, les recommandations émises dans ce travail sont très favorables à l'utilisation du simulateur dans toute activité d'enseignement pertinente ainsi que dans tout travail d'usinage programmé sur la machine. Il est également recommandé de poursuivre le développement de simulateurs pour l'ensemble des équipements à commande numérique de l'école.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à mener ce travail à terme.

En particulier, mon directeur de projet, Monsieur Jean François Chatelain, pour son implication, ses conseils, son soutien moral et la confiance qu'il m'a accordé durant toutes les étapes du projet.

Enfin, je désire remercier chaleureusement les membres de ma famille pour le soutien et l'encouragement qu'ils m'ont apportés tout au long de ce travail.

TABLE DES MATIÈRES

	Page
Sommaire.	i
REMERCIEMENTS.	ii
TABLE DES MATIÈRES.	iii
LISTE DES TABLEAUX.	vi
LISTE DES FIGURES.	vii
LISTE DES ABRÉVIATIONS ET DES SIGLES.	x
INTRODUCTION.	1
CHAPITRE 1 : PROBLÉMATIQUE.	3
CHAPITRE 2 : LA SIMULATION D'UNE MOCN.	6
2.1 Rôle de la simulation dans le processus de fabrication à CN.	6
2.2 Caractéristiques du simulateur de machine-outil à CN.	8
2.3 Avantages du simulateur.	12
2.4 Capacités du simulateur.	13
CHAPITRE 3 : ÉTAPES DE GÉNÉRATION DU SIMULATEUR.	14
3.1 La définition de la machine.	14
3.1.1 Termes clés de la définition d'une machine.	14
3.1.2 Utilisation des primitives simples et modèles CAO.	15
3.1.3 Les concepts clés de la définition d'une machine.	16
3.1.4 Étapes nécessaires à la définition de la machine.	18
3.2 Définition de la MOCN Enshu VMC-40.	19
3.3 Modélisation des composantes de la machine.	20
3.4 Assemblage de la machine.	23

3.5	Création du fichier de contrôle	27
3.5.1	Mots	27
3.5.2	Groupes	27
3.5.3	Fichier exécutable des macros	28
3.6	Modes de fonctionnement de la machine	28
3.7	Définition des outils	30
3.7.1	Procédure de définition de l'outil	31
3.7.2	Définition du "corps d'outil" et du "porte-outil"	32
3.8	Utilisation des tables	34
3.9	Édition du fichier du chemin de l'outil	37
3.10	Définition de la tâche	37
3.10.1	Fichier du chemin de l'outil	38
3.10.2	Méthode de programmation	38
3.10.3	Fichiers de sortie	39
3.10.4	Détection des collisions	39
3.11	Analyse de la simulation	40
3.12	Optipath	40
3.13	Le simulateur fin prêt	41
CHAPITRE 4 : VALIDATION DU SIMULATEUR		42
4.1	Description sommaire des essais	43
4.2	Essai 1 : Usinage d'une attache	45
4.2.1	Définition de l'usinage et des paramètres de coupe dans VERICUT	46
4.2.2	Simulation des hors courses et collisions	50
4.2.3	Comparaison du modèle simulé et du modèle nominal	52
4.2.4	Optimisation du chemin d'outil	54
4.3	Essai 2 : Usinage d'un moule	57
4.4	Essai 3 : Maquette d'automobile	60
4.5	Essai 4 : Plaque	64
4.6	Essai 5 : Poulie	66

CONCLUSION.....	68
RECOMMANDATIONS.....	70
BIBLIOGRAPHIE	71
ANNEXES.....	72
A : Format des registres associés au contrôleur de la machine.....	72
B : Tableaux des codes et autres commandes	74
C : Tableau et exemples des codes optimisables.....	80
D : Listes des fichiers de conception et des essais de validation.....	82

LISTE DES TABLEAUX

	Page
3.1 Caractéristiques de la fraiseuse Enshu VMC-40.....	19
3.2 La définition des outils de coupe.....	31
1 Définition de tous les mots et caractères.....	73
2 Les codes G des états de la machine.....	75
3 Les codes G des cycles de la machine.....	76
4 Liste des codes préparatoires de la machine.....	77
5 Commandes variées inscrites dans le registre de la machine.....	78
6 Les codes M de la machine.....	79
7 Les codes optimisables et non optimisables.....	81
8 Liste des éléments de la machine et des essais avec PRO/ENGINEER.....	83
9 Liste des fichiers de simulation accomplis avec VERICUT.....	84

LISTE DES FIGURES

	Page
2.1 Environnement du processus d'usinage virtuel	7
2.2 Système complet de vérification de VERICUT	9
2.3 Fenêtre du module de base "vericut"	11
3.1 Localisation du zéro de la machine Enshu VMC-40.	16
3.2 Décalage du référentiel de la composante vers le zéro de la machine	17
3.3 Vue d'ensemble du simulateur	20
3.4 La base	21
3.5 La Colonne	21
3.6 La table	22
3.7 Le porte-broche	22
3.8 La fixation	23
3.9 Tableau de définition des composantes	24
3.10 Fenêtre de définition des priorités	25
3.11 Le zéro de la machine et les limites d'axes	26
3.12 Tableau des limites de déplacement	26
3.13 Définition des codes G des cycles de la machine	28
3.14 Les composantes de l'outil de fraisage	30
3.15 Icônes des outils de coupes de fraisage	31
3.16 Formes des outils de coupe de fraisage	32
3.17 Icônes des corps d'outil et porte-outil de fraisage	32
3.18 Exemple de détermination d'un corps d'outil	33
3.19 Exemple de détermination d'un "porte-outil"	33

3.20	L'entrée zéro du programme.....	35
3.21	Fenêtre de définition de la tâche.....	37
3.22	Fenêtre des tolérances de détection des collisions .	39
4.1	Attache.....	45
4.2	Le modèle fini usiné par PRO/MANUFACTURING.....	46
4.3	Aperçu du programme d'usinage .	47
4.4	Le montage fin prêt pour la simulation de l'usinage.....	48
4.5	La simulation pendant la séquence de profilage .	48
4.6	Étapes d'usinage visualisées dans le module VERICUT.....	49
4.7	Pièce finie visualisée dans le module Vericut.....	49
4.8	Dépassement de la course suivant l'axe X.....	50
4.9	Fenêtre des messages indiquant une hors course.....	51
4.10	Collision entre l'outil de coupe et la fixation.....	51
4.11	Fenêtre des messages indiquant des collisions.....	52
4.12	Boite de dialogue de la fonction "Autodiff".....	52
4.13	Résultats de la comparaison avec "Autodiff".....	53
4.14	Les excès et les manques de matière.....	53
4.15	Assignation des outils et autres caractéristiques.....	54
4.16	Fenêtre de contrôle d'"optipath".....	54
4.17	Résultats de l'optimisation.....	55
4.18	Résultat de la simulation du programme original.....	56
4.19	Résultat de la simulation du programme optimisé.....	56
4.20	Moule.....	57
4.21	Définition du modèle.....	57
4.22	Définition des outils.....	58
4.23	Fenêtre de chemin d'outil.....	58
4.24	Fenêtre du settings avec "multiple toolpath files".....	59
4.25	Le moule fini usiné avec le module de base de Vericut.....	59
4.26	Le modèle "Auto".....	60

4.27	Le chemin d'outil généré avec Pro/Engineer	60
4.28	Les étapes de la simulation avec Pro/Engineer	61
4.29	La fenêtre de déroulement du programme d'usinage avec VERICUT.	61
4.30	Étapes de la simulation de l'usinage du modèle "Auto" avec VERICUT.	62
4.31	La séquence de dégrossissage optimisée.	63
4.32	Les excès sur la surface de la plaque.	64
4.33	Correction du programme.	64
4.34	Les étapes de la simulation de la plaque	65
4.35	La poulie.	66
4.36	La poulie en cours de simulation avec Vericut	67
4.37	La poulie semi finie.	67

LISTE DES ABRÉVIATIONS ET DES SIGLES

MOCN	Machine-outil à commande numérique.
CN	Commande numérique.
CFAO	Conception et fabrication assistée par ordinateur.

INTRODUCTION

Dans le monde concurrentiel d'aujourd'hui, de plus en plus d'entreprises œuvrant dans le domaine de l'usinage s'orientent vers la simulation. Le respect des exigences de la qualité des pièces et des délais de production parfois très courts sont les préoccupations majeures des décideurs. Le développement des simulateurs de machines-outils a sensiblement diminué le temps des essais et de fabrication, contribuant ainsi à l'accroissement de la productivité des entreprises manufacturières.

La simulation de l'usinage est basée essentiellement sur les techniques de la représentation virtuelle. Pour permettre une meilleure perception du modèle à accomplir et le déroulement de sa fabrication en présence des défaillances possibles, le simulateur doit représenter fidèlement le système et la machine.

Le principal objectif du projet consiste au développement d'un simulateur pour la machine-outil à commande numérique à trois axes ENSHU du laboratoire d'usinage à commande numérique de l'école de technologie supérieure. Le simulateur a pour fonction principale la vérification des programmes d'usinage programmés soit manuellement ou à l'aide d'un système de conception et de fabrication assistées par ordinateur (CFAO) de manière à éviter toute interférence entre les organes de la machine et le système pièce/montage utilisé lors de l'usinage. La validation hors ligne du programme d'usinage assure une grande fiabilité en détectant toute anomalie, collision possible ou mauvais choix de parcours de coupe, avant même la production d'une première pièce.

La première étape vise la construction et la configuration de la machine virtuelle à l'aide des outils logiciels VERICUT et PRO/ENGINEER. Ce dernier, est utilisé dans ce projet afin de modéliser l'ensemble des éléments de la machine selon les spécifications dimensionnelles du manufacturier. La géométrie des différentes composantes est par la suite importée dans le générateur de simulateur VERICUT, à partir duquel la définition de la cinématique de la machine ainsi que les différentes fonctionnalités du contrôleur FANUC OM-A sont configurées.

La validation du simulateur développé revêt également une très grande importance. Celle-ci consiste à la vérification de différents programmes NC spécifiquement sélectionnés, afin de s'assurer du bon fonctionnement et du degré de fiabilité du simulateur.

Le détail de la construction et de la validation proposée pour le simulateur développé sont présentés respectivement aux chapitre 3 et 4, après avoir introduit au préalable l'outil de développement VERICUT, ainsi que la problématique relative à cet important projet.

CHAPITRE 1

PROBLÉMATIQUE

La simulation est utilisée dans de nombreuses disciplines associées autant aux domaines de l'ingénierie qu'aux domaines technologique et scientifique en général. Au cours des cinquante dernières années, depuis que sa définition est devenue formelle, elle a été adaptée pour une grande variété d'applications, telles que la conception et la fabrication des produits, le domaine architectural, l'industrie des jeux vidéo et plusieurs autres. Aujourd'hui, cette technique est utilisée notamment dans la conception de nouveaux systèmes et dans l'analyse des systèmes existants, alors qu'elle s'exerce pour tous les types d'activités.

Dans l'arène interactive, un certain nombre de produits de simulation ont émergé. Les éléments de la CAO fournissent un environnement graphique pour construire les mondes virtuels simulés et les entités qui les peuplent. L'outil permet la création d'un environnement opérationnel. Il fournit une interface utilisateur graphique pour l'exécution de la simulation.

L'usinage est un domaine où l'erreur coûte cher. Chaque décideur doit opter pour des solutions permettant d'améliorer la qualité des produits et de maximiser la productivité. Un des buts ultimes est l'utilisation optimale des ressources et la réduction des temps improductifs des différentes tâches d'essais et de préparation.

Afin de perdre moins de temps à ajuster les paramètres, à mettre au point les machines et à corriger les programmes, l'utilisation d'un simulateur s'avère la solution

idéale. Elle permet d'exploiter des données issues du système de conception et de fabrication CFAO.

Le simulateur est un outil idéal en industrie. Il permet de tester les nouveaux programmes rapidement et à faible coût. C'est un outil de formation parfaitement adapté aux universités où l'apprentissage des différents langages, notamment les codes G, APT et CL FILE, est indispensable.

En industrie, les opérateurs sont une ressource de valeur. En essayant les programmes à l'avance, les opérateurs sont libérés des tâches les plus contraignantes, ce qui leur permet de se consacrer davantage à la qualité et à l'efficacité des performances de la machine, au lieu de s'inquiéter des éventuelles collisions et de leurs conséquences. Leur sécurité s'en trouve protégée.

En plus d'assurer un fonctionnement dénué d'avaries, le simulateur a comme but la vérification et l'optimisation des trajectoires d'outils. Ces dernières peuvent contenir des codes non supportés par la machine causant des arrêts fréquents suite aux messages d'erreurs. Les vitesses d'avance peuvent également être spécifiées avec conservatisme affectant ainsi la productivité de l'entreprise. Les modules d'optimisation offerts par les simulateurs permettent notamment d'ajuster les vitesses de coupe en fonction du type de matériau et des outils utilisés, afin d'alléger les temps d'usinage tout en assurant un haut niveau de qualité pour les pièces produites. Enfin, les outils de simulation permettent également la comparaison du produit fini avec le produit nominal afin de détecter les défauts de forme ainsi que les excès et les manques de matière sur la pièce à produire.

Les avantages de la simulation sont donc importants, de par sa capacité d'approcher la perfection et de réduire sensiblement les erreurs. La construction du simulateur nécessite la modélisation physique des différentes composantes, la définition de la cinématique associée à la machine, ainsi que l'introduction des paramètres associés

au contrôleur de la machine. Les différents outils de coupe, les montages et les programmes d'usinage doivent également être définis en fonction de la production typique liée à la machine en question.

CHAPITRE 2

LA SIMULATION D'UNE MOCN

2.1 Rôle de la simulation dans le processus de fabrication à CN

De la PME aux plus grandes entreprises, la simulation est un atout de productivité adapté à toutes les entreprises industrielles utilisant des machines-outils à commande numérique pour des opérations en 2½ à 5 axes. Elle convient parfaitement aux PME, fournisseurs de pièces et d'équipements, notamment dans les secteurs de l'aéronautique, de l'automobile et des moulistes outilleurs.

Pour la majorité des compagnies, couper de 50% ou plus les cycles d'introduction d'un nouveau produit constitue un but immédiat [2]. Pour ce faire, une forte dépendance aux simulations de la fabrication et à l'usinage en particulier est essentielle.

Les machines-outils virtuelles sont définies comme des représentations électroniques des machines-outils décrivant leurs caractéristiques de précision. Ces dernières sont limitées et dépendent étroitement des étapes 'modélisation 3D – FAO - simulation' où les tolérances ne sont pas prises en compte dans tout le processus. Néanmoins, les MOCN virtuelles permettent aux manufacturiers de tester et d'optimiser leurs plans de fabrication avant de compromettre n'importe quelle ressource et avant qu'aucun rebut n'ait été généré.

Utilisant la pièce brute et le programme (codes G) comme entrée, la machine-outil virtuelle génère une sortie décrivant la pièce finie virtuelle, qui à son tour est utilisée comme entrée au CMM virtuel pour l'inspection (figure 2.1).

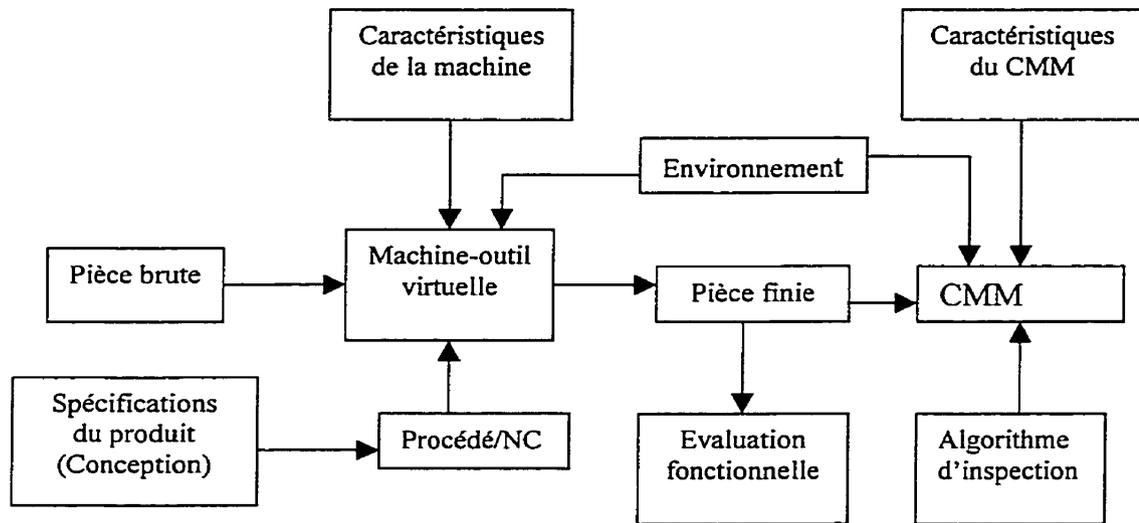


Figure 2.1 Environnement du processus d'usinage virtuel [2].

Il existe plusieurs logiciels de modélisation et de vérification des programmes de pièces sur le marché aujourd'hui. La plupart des outils de modélisation permettent à l'utilisateur de définir la géométrie et les mouvements des composants structurales de la machine. Basé sur ces définitions, l'utilisateur est capable de visualiser la machine comme un modèle solide et d'imiter le mouvement d'usinage afin de déceler les collisions possibles.

Le logiciel de vérification modélise le matériau enlevé de la pièce brute. En général, le simulateur est basé sur un modeleur, intégrant un module de génération d'erreurs, qui incorpore les représentations mathématiques des erreurs de mouvement pour chaque image (slide) dans le modèle cinématique de la machine. Basé sur les erreurs d'une image individuelle, le modèle cinématique permet de calculer l'erreur

résultante au niveau de l'outil de coupe et définir ainsi la position de l'outil relativement à la pièce coupée [2].

Utilisé comme un système, la géométrie de la pièce et son programme associé (codes G) sont sélectionnés et approuvés comme entrée au modèle de la machine. Le chemin de l'outil est généré en spécifiant les positions individuelles des axes en fonction du temps. Durant la simulation graphique, l'outil se déplace le long du profil de la pièce dans l'espace de travail de la machine. À chaque intervalle de calcul, le modèle de la machine détermine la position nominale du point de l'outil dans les coordonnées de la machine. De plus, le module d'erreur détermine l'erreur de parcours correspondante. Une fois la simulation graphique terminée, les positions nominales individuelles de l'outil et les valeurs des erreurs correspondantes sont stockées dans un fichier de sortie. Ce dernier est utilisé par le module d'analyse et de comparaison. En utilisant ce système, les déviations de la géométrie par rapport à la pièce nominale sont visualisées.

Le but de la plupart des logiciels de simulation graphique des commandes numériques est le traitement en temps réel des données de commande numérique présentées. La plupart des méthodes sont basées sur la théorie booléenne, où la pièce forme le bloc de départ et est représentée par un ensemble. Le volume balayé et coupé par l'outil à chaque ligne sur la pièce est représenté par un second ensemble, et est ôté de la pièce en utilisant l'opérateur de différence de l'ensemble [2]. Ces méthodes simulent le processus d'usinage en utilisant les capacités graphiques de l'ordinateur afin de représenter les mises à jour de la pièce en cours d'usinage.

2.2 Caractéristiques du simulateur de machine-outil à CN

Au-delà de sa capacité à montrer l'enlèvement de matière, un simulateur doit réaliser une simulation réaliste d'une machine outil et d'un contrôleur, exactement de la manière dont cela se passe dans l'atelier. Détectant en permanence les éventuelles

collisions entre les différentes composantes de la machine, comme par exemple les montages et les outils, il alerte l'utilisateur des collisions qui peuvent endommager le modèle, la machine, ou mettre en danger les opérateurs. Le programme est flexible et peut être facilement personnalisé pour répondre aux besoins spécifiques d'une application particulière.

Le simulateur développé dans le cadre de ce travail est basé sur le logiciel VERICUT, qui simule et rejoue l'enlèvement de matière d'un parcours d'outil, permettant au programmeur CN de corriger un mouvement inefficace ou une erreur de programmation, sans être obligé de tester le parcours sur une machine outil à commande numérique. Le programme VERICUT décrit des opérations de fraisage, perçage, tournage, électroérosion à fil, ainsi que des opérations de fraisage/tournage de 2 axes à 5 axes en simultanément. Tel qu'illustré à la figure 2.2, il est constitué de deux modules principaux, le premier appelé "simulation machine" sert à la construction et à la simulation de la machine, le second communément appelé "vericut" ou module de base, est utilisé dans la simulation et la vérification d'enlèvement de matière.

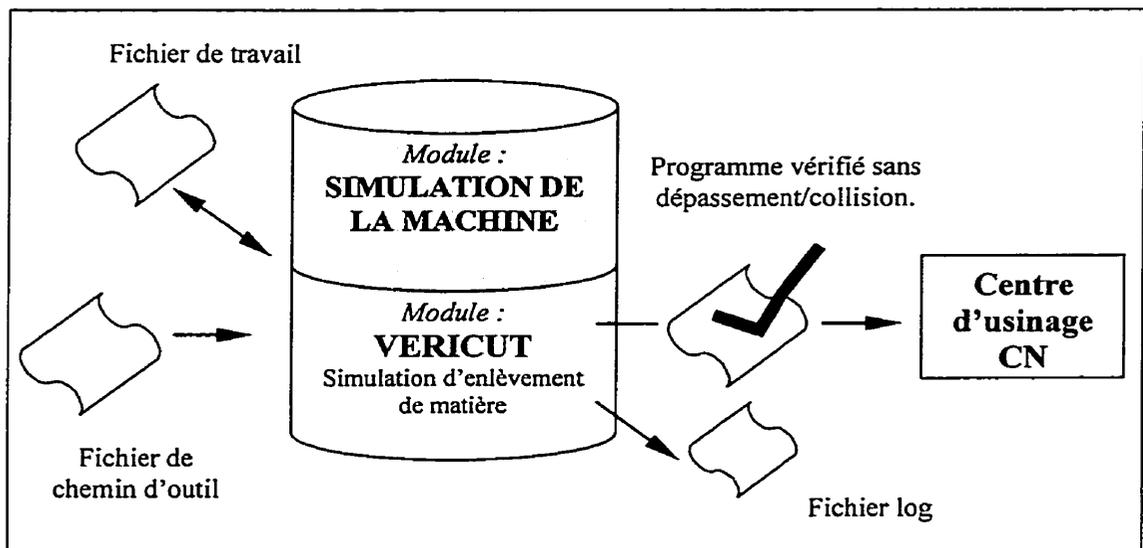


Figure 2.2 Système complet de vérification de VERICUT

Tous les types de machines-outils et de contrôleurs peuvent être définis dans VERICUT (fraisage, tournage, perçage, électroérosion à fil), ou peuvent être conçus en CAO et importés sous forme de fichiers STL. Plusieurs exemples de machines et de contrôleurs standard, facilement modifiables, sont inclus dans le logiciel. Plus qu'un simple simulateur robotique, le programme reproduit la logique exacte du contrôleur, ce qui permet à l'utilisateur de visualiser les effets de différentes techniques de programmation telles que la compensation 3D, la compensation longueur, la programmation en jauge outil, le changement du référentiel, les variables, les boucles, les sous-programmes et les macros CN.

Le logiciel est équipé d'un module "OptiPath" incluant les fonctions d'optimisation qui permettent d'éliminer le temps passé à programmer manuellement les vitesses d'avance, sources d'erreurs. Cette fonction est largement développée pour permettre l'optimisation de la vitesse de rotation de la broche et inclut désormais les données outil/matière/vitesse d'avance de la base de données IAMS METCUT, une base de données très utilisée dans le milieu industriel. Les parcours d'outils sont automatiquement optimisés en déterminant la vitesse d'avance optimum basée sur les conditions d'usinage et les caractéristiques de l'outil. Le programme applique au parcours d'outil les informations ainsi modifiées, éliminant par la même occasion la vitesse initialement programmée. Ce système permet d'accroître la productivité de la machine, éliminant les déplacements trop rapides de l'outil dans la matière, maximisant la durée de vie de l'outil, et libérant les ressources pour la programmation.

Particulièrement utile pour les constructeurs automobiles et les moulistes, le module "FastMill" permet une vérification très rapide des très gros parcours d'outil. Le modèle peut aussi être exporté sous forme d'un fichier STL.

VERICUT est aussi équipé d'un ensemble complet d'outils permettant de vérifier l'exactitude dimensionnelle du modèle usiné par simulation. N'importe quel membre de

l'équipe de fabrication peut s'assurer que le résultat de la simulation correspond bien au modèle initial, en révélant graphiquement les différences entre les deux. Le logiciel réalise également un contrôle permanent en cours d'usinage mettant en évidence les parties trop ou pas assez usinées.

Le logiciel permet de simuler les fichiers ISO ainsi que n'importe quelle sortie de logiciel FAO. Le programme est disponible sur stations de travail et sur PC, et tourne sur les systèmes d'exploitation les plus connus, incluant UNIX, WINDOWS 95 et WINDOWS NT.

La figure 2.3 montre la fenêtre du logiciel avec les différents menus des fonctions du module de base de vericut.

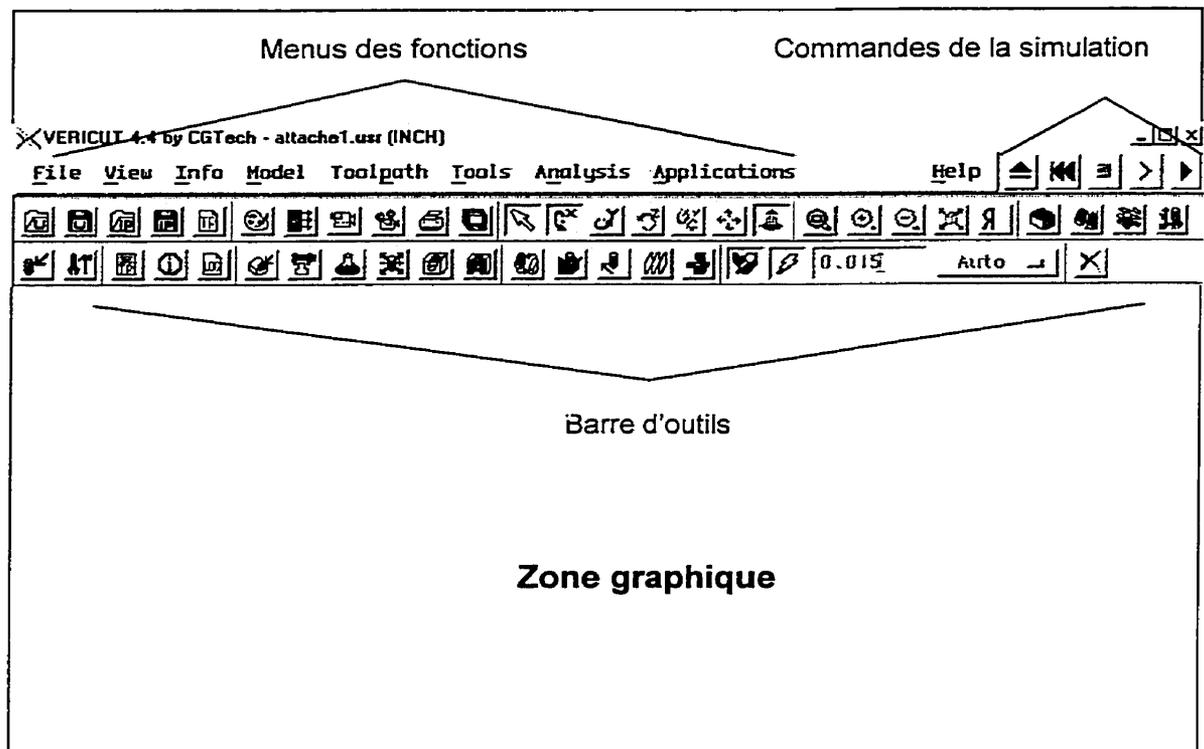


Figure 2.3 Fenêtre du module de base "vericut"

2.3 Avantages du simulateur

Un simulateur de commandes numériques comporte plusieurs avantages :

- Gain de temps : Il réduit ou élimine les tests, préservant ainsi le temps machine, le temps de l'opérateur, et celui de la programmation.
- Amélioration de la qualité : Il vérifie l'exactitude des dimensions de la pièce et optimise les parcours d'outil pour un meilleur état de surface.
- Gain financier : Il réduit ou élimine les dépenses engendrées par les avaries possibles sur la machine, les reprises, les pièces endommagées, les outils, ou les bris des fixations.
- Augmentation de la productivité : Il réduit le temps machine et les interruptions dans la production.
- Gain de confiance : Les programmes sont testés sur l'ordinateur, donc ils sont corrects dès les premiers montages.
- Préservation des ressources : Il réduit l'usure de la machine-outil et des outils.
- Amélioration de la sécurité et de la formation : Forme les programmeurs, les opérateurs, et les étudiants sans utiliser de temps machine ou risquer un bris dangereux et coûteux de la machine.

2.4 Capacités du simulateur

Dans la "simulation / vérification CN" on distingue quatre fonctions principales :

1. La simulation : Le simulateur propose une simulation 3D, avec modèle solide ombré de l'ensemble de la machine outil et du procédé de coupe. Cette visualisation permet aux programmeurs et aux opérateurs de visualiser l'usinage exactement comme il sera réalisé en atelier, et de contrôler les collisions.
2. La vérification : Le simulateur détecte les problèmes du parcours CN. C'est un outil d'inspection visuelle qui met en évidence les avances rapides dans la matière, les parties trop usinées et les collisions potentielles. Les programmeurs peuvent détecter et corriger des problèmes avant les essais dans l'environnement de production.
3. L'analyse : Le simulateur identifie la ligne du parcours responsable d'une erreur. Un grand nombre d'outils 3D sont mis à disposition pour vérifier la précision de la pièce, et son exactitude dimensionnelle. Cette dernière reste limitée et dépend du processus et du format de la modélisation. Le simulateur compare la pièce simulée avec le modèle conçu initialement pour valider la correspondance parfaite entre les deux. Le simulateur peut également réaliser cette comparaison de manière constante lors de l'usinage. Ce processus demeure virtuel et ne tient nullement compte de l'environnement physique propre à l'usinage où bon nombre de facteurs contribuent aux déviations dimensionnelles et géométriques. Celles-ci ne peuvent évidemment pas être simulées par l'outil Vericut.
4. L'optimisation : Le simulateur détermine automatiquement la meilleure vitesse d'avance pour chaque segment du parcours d'outil, en se basant sur les conditions machine et sur le volume de matière usinée. L'optimisation des vitesses d'avance CN réduit de façon importante le temps d'usinage et améliore la qualité de surface.

CHAPITRE 3

ÉTAPES DE GÉNÉRATION DU SIMULATEUR

La génération du simulateur exige la définition de la machine à simuler, les commandes spécifiques à la machine, les fichiers d'exécution, les outils, les fixations et la tâche à exécuter se référant aux différents programmes.

3.1 La définition de la machine

La définition de la machine décrit la cinématique et les propriétés physiques de la MOCN. Elle consiste au respect du nombre des composantes connectées telles qu'elles le sont sur la MOCN réelle. Des formes 3D connues comme ``primitives'' sont utilisées pour représenter la taille des composantes et leurs formes.

3.1.1 Termes clés de la définition d'une machine

Le zéro de la machine : La condition où toutes les composantes de la machine sont situées à leur zéro respectif.

Le point de référence de la machine : La localisation de l'outil avec la machine (au zéro global de la machine).

Le système de coordonnées de la machine : il utilise la philosophie du mouvement de l'outil et reflète comment la machine maintient la relation entre l'outil et la pièce brute.

Cette philosophie est utilisée quelle que soit la composante en mouvement, soit l'outil ou la pièce.

3.1.2 Utilisation des primitives simples et modèles CAO

VERICUT permet la construction de la machine avec deux types de primitives :

- Primitives simples : incluant les blocs, cylindres, et les cônes.
- Primitives de fichier modèle : incluant les formes de fichiers modèles de polygones de VERICUT ou stéréolithographiques (STL). Ces modèles sont créés typiquement à l'extérieur de la simulation de machine via le système de CAO (PRO/ENGINEER dans notre cas).

L'utilisation de la géométrie du modèle CAO pour représenter les composantes de la machine offre une méthode facile et efficace de définition des machines 3D. La conception CAO doit être solide et faite individuellement pour chaque composante de la machine, en respectant le zéro global de la machine pour une éventuelle localisation correcte sur celle-ci en vue de l'assemblage.

Les modèles CAO utilisés sont CATV, STL, IGES, ou d'autres utilitaires de sortie. La simulation machine interprète directement les fichiers STL et les fichiers VERICUT. Les autres fichiers, tel que IGES, doivent être préalablement convertis en ces deux types de fichiers avant leur introduction dans la simulation machine.

3.1.3 Les concepts clés de la définition d'une machine

En définissant la machine, on s'assure que toutes les composantes physiques de cette dernière sont situées à leurs localisations zéro respectives. Cette condition de la machine à commande numérique est connue comme "le zéro de la machine".

D'après les stipulations du fabricant de la machine, il existe des dimensions de positionnement à respecter relatives au zéro de la machine. Pour notre cas, tel que montré sur la figure 3.1, ce dernier est positionné sur la face de la broche complètement en haut à droite des limites de course de l'axe Z et X respectivement et à environ mi-course de l'axe Y. Il constitue le point de convergence de tous les éléments de la machine pendant l'assemblage.

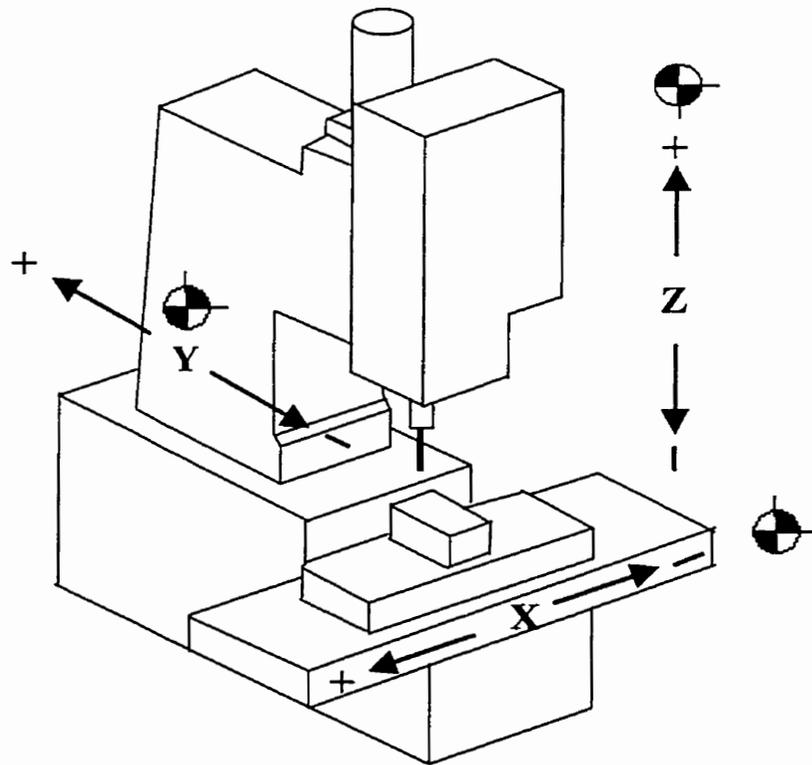


Figure 3.1 Localisation du zéro de la machine Enshu VMC-40

Par exemple, la composante "table" illustrée à la figure 3.2 montre le décalage du référentiel de la composante (X, Y, Z) issu de la modélisation avec PRO/ENGINEER par rapport à celui du zéro de la machine (X_w, Y_w, Z_w). Celui-ci peut s'effectuer avec PRO/ENGINEER pendant la modélisation de la table ou lors de son assemblage dans la machine avec VERICUT en se basant sur les distances connues (X_i, Y_i, Z_i).

Le but du décalage du référentiel est de respecter la localisation de cette composante relativement au zéro de la machine et par conséquent relativement aux autres composantes.

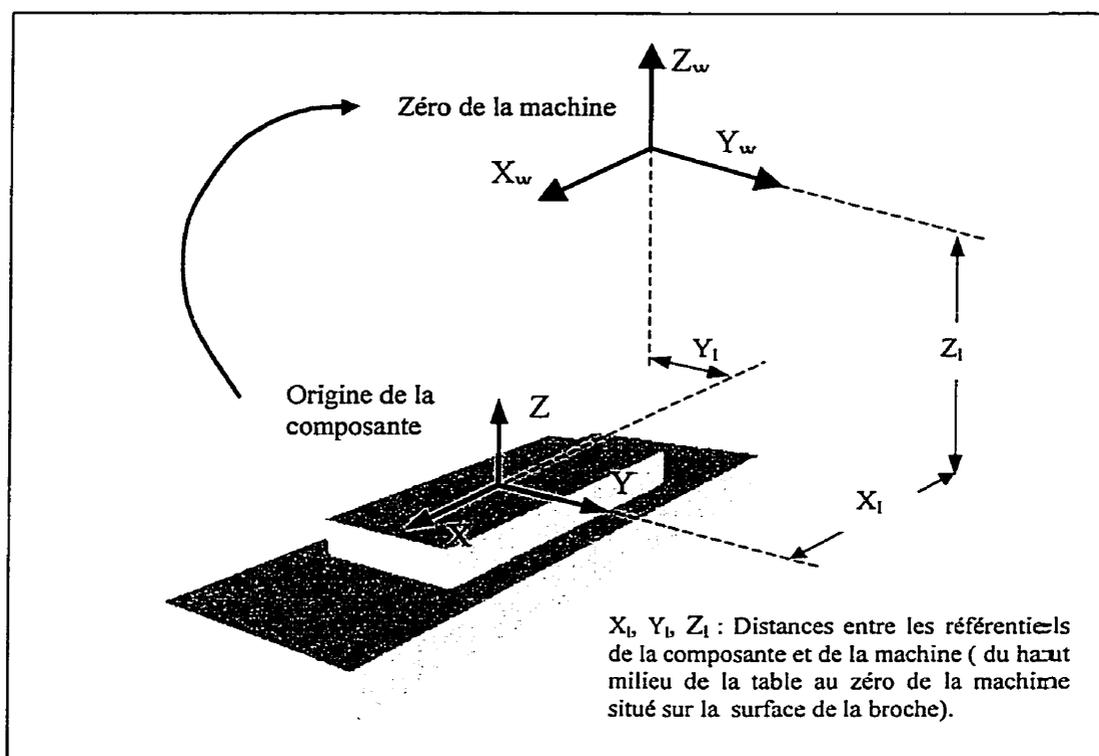


Figure 3.2 Décalage du référentiel de la composante vers le zéro de la machine

Donc, toutes les composantes de la machine sont définies et connectées entre elles à leurs localisations zéro respectives.

La définition de la machine commence avec “la base”, composante fixe touchant le sol. Une machine valide doit contenir la base, la pièce brute et l’outil [4]. Les éléments représentant les axes de la machine, et le reste des autres composantes sont définis et connectés dans le même ordre que sur la machine réelle.

3.1.4 Étapes nécessaires à la définition de la machine

1. Importer les modèles CAO des composantes individuelles de la machine; (base, table, colonne, porte-broche, etc).
2. Débuter la construction de la machine virtuelle avec la base, et ajouter les composantes les unes à la suite des autres en respectant l’ordre propre à la chaîne cinématique de la machine.
3. Ajouter les primitives 3D (si nécessaire) pour représenter la taille et la forme des composantes. La modélisation des composantes peut se faire à l’aide de primitives ou importées en provenance d’un système CAO/FAO.
4. Tester les fonctions de la machine (mouvements, changement d’outils, ...).

Enregistrer la machine ainsi construite comme un fichier dans le menu “machine”.

3.2 Définition de la MOCN Enshu VMC-40

La MOCN Enshu VMC-40 est une fraiseuse verticale à 3 axes (extensible à 4 axes), avec un contrôleur Fanuc OM-A. Les caractéristiques de la machine sont décrites dans le tableau 3.1.

Tableau 3.1

Caractéristiques de la fraiseuse Enshu VMC-40

Table	Dimensions de la surface de travail. mm (po)		900 × 400 (35.4 × 15.7)
	Déplacement maximum. (X × Y × Z). mm (po)		560 × 400 × 460 (22 × 15.7 × 18.1)
	Avance de coupe. mm(po)/min		1 à 5000 (0.01 à 197)
	Traverse rapide	X et Y (mm(po) / min	18000 (708.6)
Z (mm(po) / min		12000 (472.4)	
Broche	Type de broche		JIS No. 40
	Vitesse (tours / min)		100 à 6000
Distance de la broche au haut de la table			190 à 650 (7.4 à 25.5)
Hauteur de la table au sol			650 (25.5)
Poids maximum admissible de travail (kgf)			300
Outils	Capacité du magasin d'outils		20 outils.
	Dimensions max. outils (diam. × Long)		145 (90) × 250 (mm)
	Poids max. de l'outil (kg)		5
Moteurs	Broche		AC 5.5 KW (taux continue)
			7.5 KW (taux aux 30 minutes)
	Avance le long des axes		X et Y : modèle 5 AC 0.9 KW
			Z : modèle 10 AC 1.8 KW
	Pompe de refroidissement		180 W
	Magasin d'outils		0.2 KW × 2 moteurs.
Pompe de lubrification		4W	
Puissance			200/220 ± 10% 50/60 Hz 3-phases; 17.3kVA
Espace au sol (longueur × largeur × hauteur)			2230 × 2340 × 2635 (87.7 × 92.1 × 104)
Poids de la machine. (kgf)			3700

3.3 Modélisation des composantes de la machine

La première étape de la construction de la MOCN Enshu VMC-40 consiste à modéliser les principales composantes sur un modelleur solide (Pro/Engineer) avant de les insérer dans le module de simulation de Vericut. Telles qu'illustrées à la figure 3.3, les composantes modélisées sont constituées de la base fixe faisant contact avec le sol, et soutenant la colonne qui se déplace le long de l'axe Y, le porte-broche le long de l'axe vertical Z, et la table le long de l'axe X, ainsi que la broche assemblée au porte-broche.

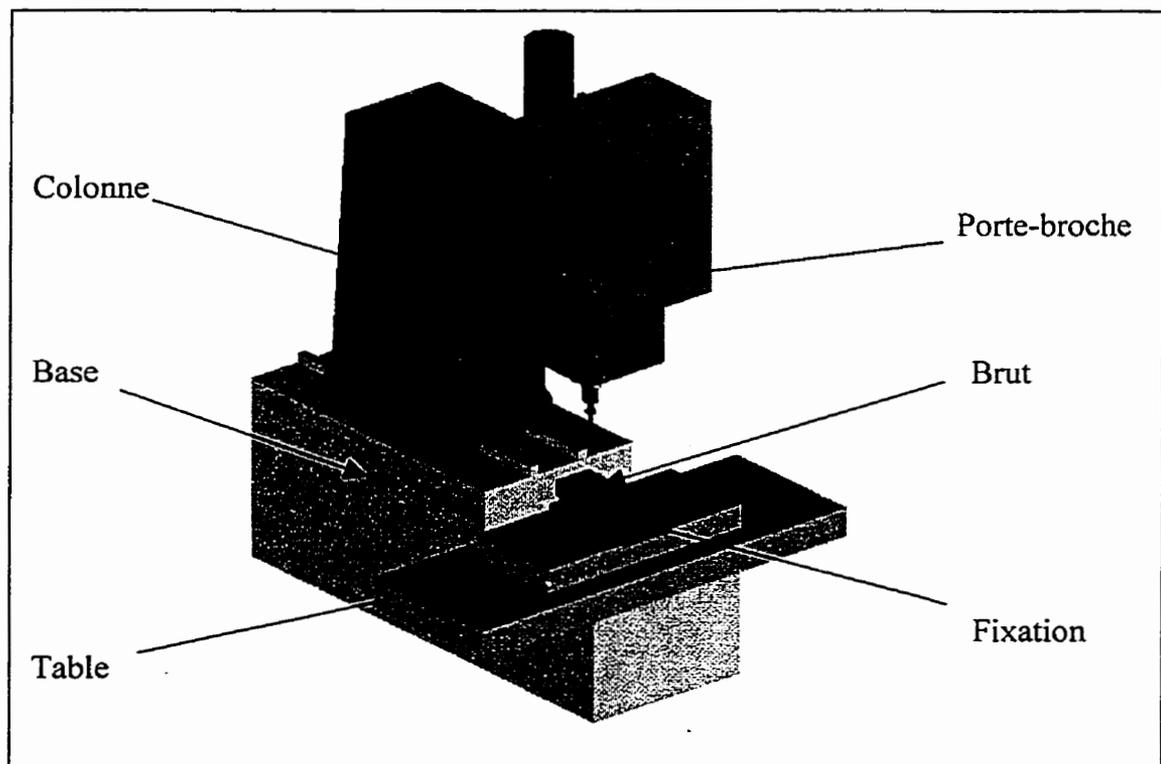


Figure 3.3 Vue d'ensemble du simulateur

Les caractéristiques dimensionnelles des composantes du simulateur sont montrées dans les figures 3.4 à 3.8, représentant la base, la colonne, le porte-broche et la fixation respectivement.

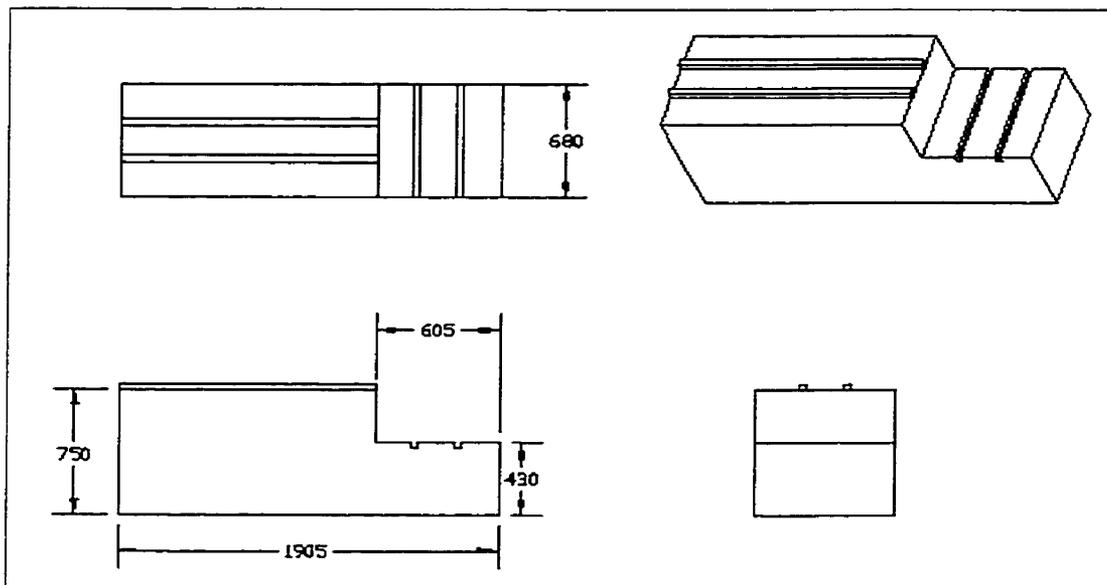


Figure 3.4 La base

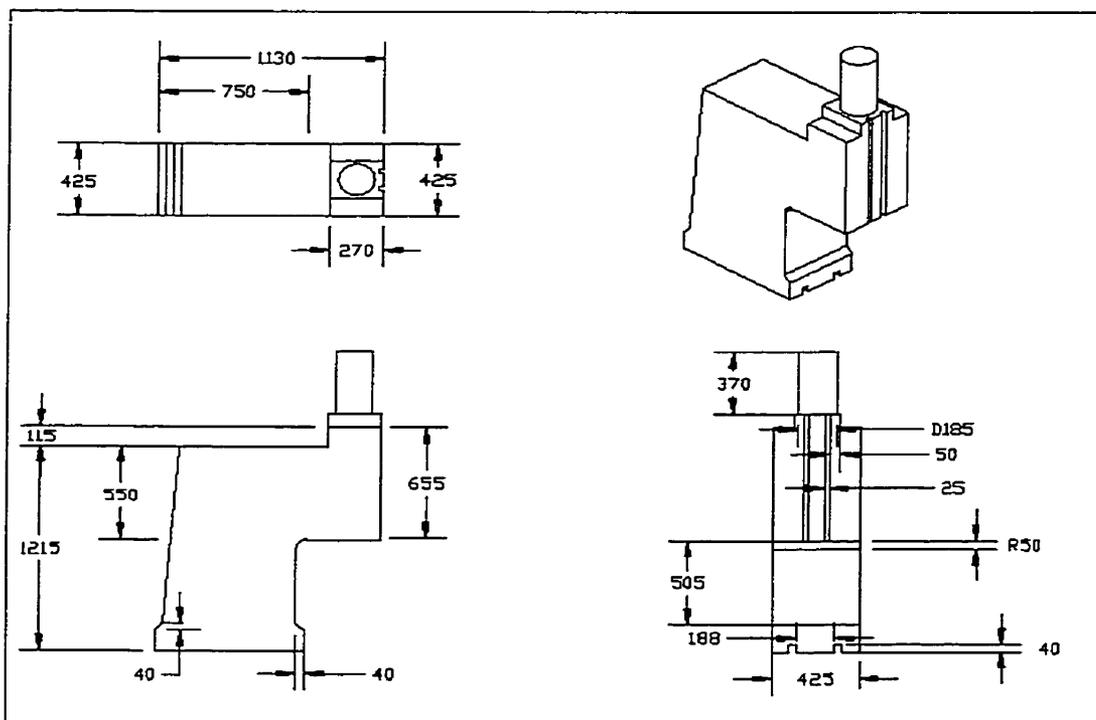


Figure 3.5 La Colonne

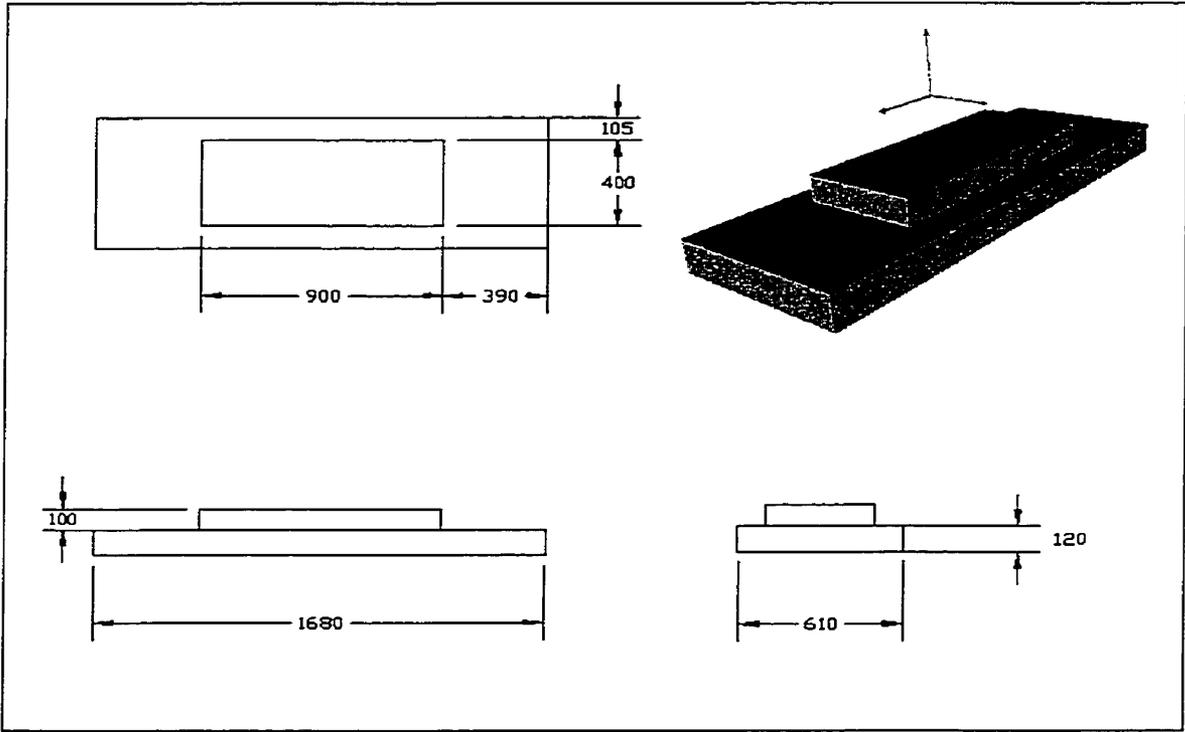


Figure 3.6 La table

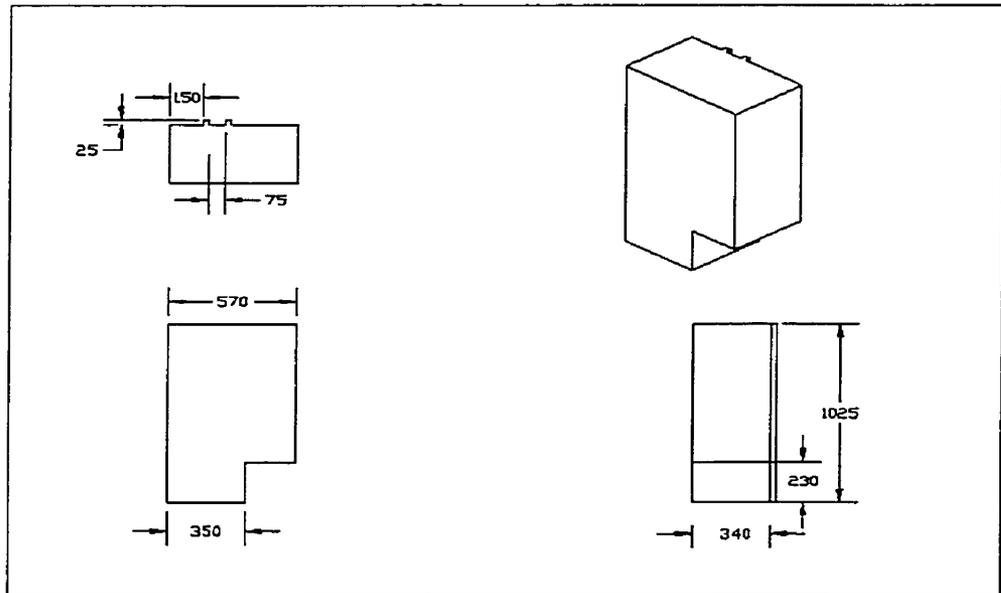


Figure 3.7 Le porte-broche

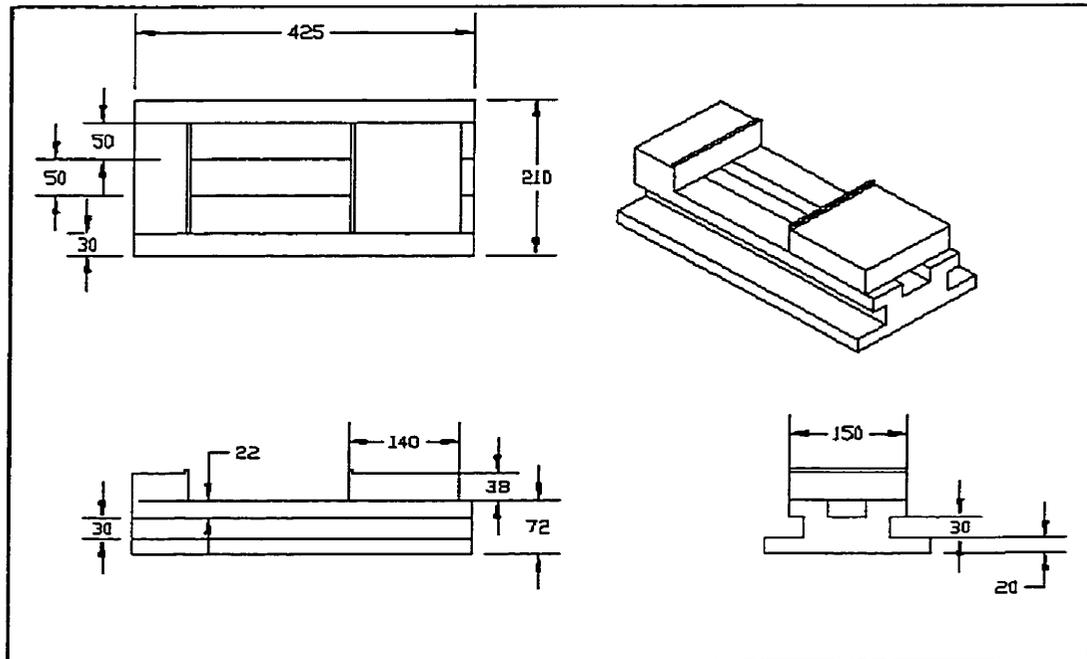


Figure 3.8 La fixation

3.4 Assemblage de la machine

La modélisation des composantes principales est suivie par la construction et l'assemblage de la machine. Cette étape comporte la création d'un fichier ENSHU.mch dans le menu "MACHINE", et la définition de toutes les composantes :

La définition des composantes "Base, table (X), colonne (Y), porte-broche (Z), fixation, outil, pièce brute" est réalisée à l'aide de la fonction "components" incluse dans le menu Machine. L'ajout au fur et à mesure des éléments dans l'ordre de connexion se fait par la commande "add".

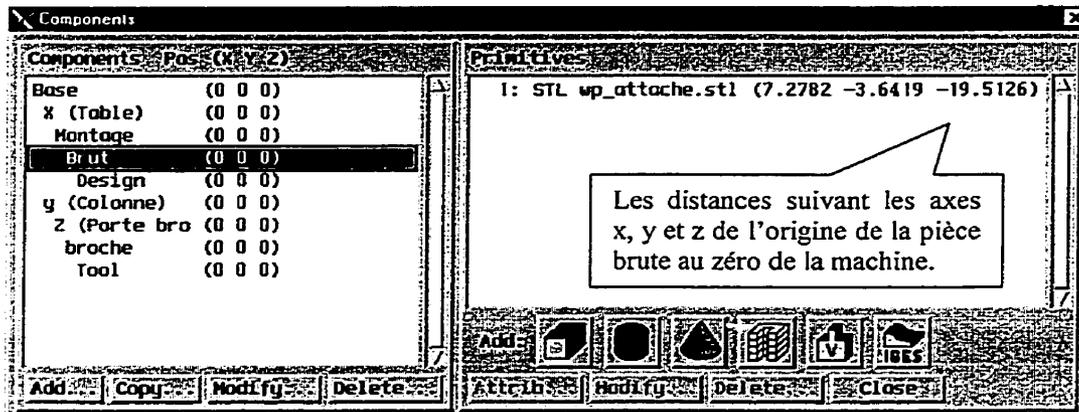


Figure 3.9 Tableau de définition des composantes

La procédure commence par la connexion de "X" à l'élément "Base". "X" représente le déplacement de l'élément "Table" le long de l'axe x, qui à son tour, est liée aux éléments de la fixation et à la "pièce brute". D'autre part, comme montré à la figure 3.9, l'élément "base" est aussi connecté à "Y" qui représente le déplacement de l'élément "Colonne" le long de l'axe y. Par conséquent, l'élément "porte-broche" représentant "Z" et se déplaçant le long de l'axe z, l'élément "broche" et l'ensemble de l'outil sont reliés à ce dernier.

La fraiseuse Enshu est dotée de 3 axes à déplacements simultanés, la définition des priorités est simple et se fait à l'aide de la fonction "priority". Cette dernière contrôle les mouvements des axes de la machine en mode de positionnement rapide (G0). Tel qu'illustré à la figure 3.10, les mouvements d'axes ayant les priorités élevées "Priority 1" sont placés avant ceux ayant des priorités plus faibles "Priority 2 ou 3". Par défaut, les axes de la machine ont des priorités égales et arrivent à leurs destinations en même temps.

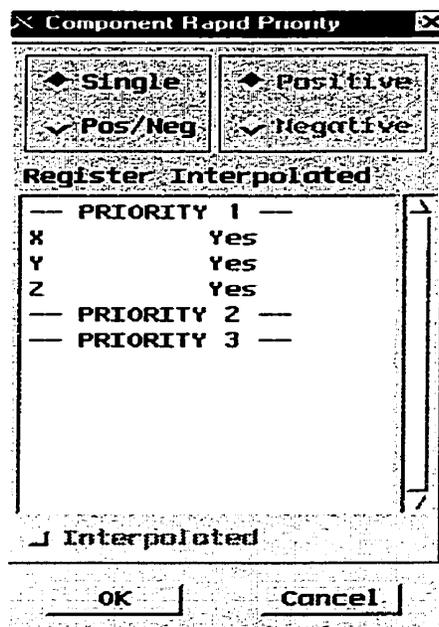


Figure 3.10 Fenêtre de définition des priorités

Une fois la machine construite, le zéro de la machine (figure 3.11), ainsi que les limites de déplacement des différents axes sont spécifiés.

Le tableau de la figure 3.12 montre la définition des limites de déplacement. Ces dernières, illustrées à la figure 3.11 b sont introduites suivant les indications du fabricant de la machine. Lors des simulations, on peut distinguer les dépassements grâce à la fonction de détection et les visualiser par une couleur distincte.

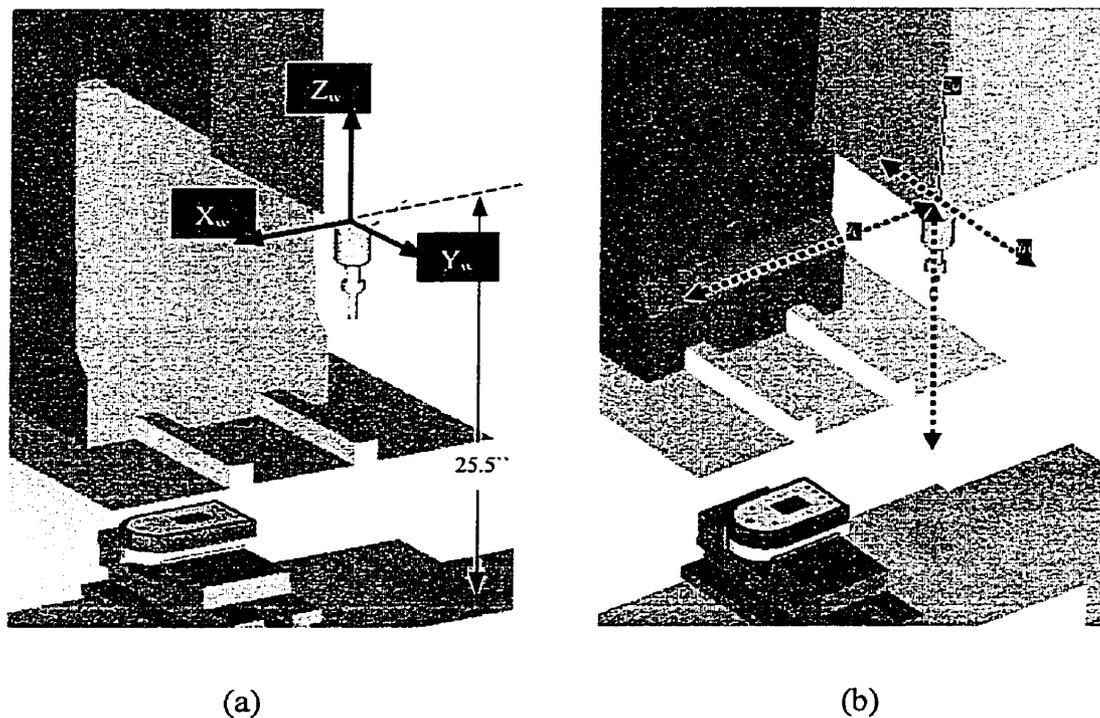


Figure 3.11 Le zéro de la machine et les limites d'axes

Name	Minimum	Maximum	Ignore
X (Table)	0	22	<input type="checkbox"/>
Y (Colonne)	-5.9	9.84	<input type="checkbox"/>
Z (Porte bro)	-18.1	0	<input type="checkbox"/>

Overtravel Detection: Off On
 Overtravel Color: Red

OK Cancel

Figure 3.12 Tableau des limites de déplacement.

3.5 Création du fichier de contrôle

Le fichier contrôle (ENSHU.CTL) contient les données décrivant le traitement des données (codes G) par le contrôle CN. Les données de ce fichier incluent la liste des codes machine interprétables, l'information sur le format des codes machine, les règles d'appels des macros et la spécification du fichier CME fournissant les macros qui contrôlent le traitement du fichier de chemin d'outil.

Le menu "control" est composé de deux groupes principaux permettant la spécification du fichier de commande : "mots" et "groupes".

3.5.1 Mots

Tel que décrit dans le tableau 1 à l'annexe A , ce groupe contient la liste des mots et caractères que la commande peut interpréter, chacun de ces derniers se voit spécifier la fonction, le format, etc.

3.5.2 Groupes

Cet élément spécifie les associations des macros avec leurs mots/valeurs de codes G respectifs. Tel que spécifié dans les tableaux 2 à 6 à l'annexe B, il est constitué de cinq sous-groupes distincts comprenant les codes G des états, les codes G des cycles et les codes G préparatoires de la machine, ainsi que les codes M et les commandes inscrites dans le registre de la machine. Chaque sous-groupe défini peut être configuré de manière à faire appel à une ou plusieurs macros pour exécuter des actions variées. Cette fonctionnalité permet de personnaliser la commande. Comme exemple d'introduction des codes, la figure 3.13 montre la fenêtre de spécification des codes G des cycles de la machine. En cliquant sur "Modify" une liste de fonctions et descriptions permet de spécifier l'utilité de chaque macro et ses conditions.

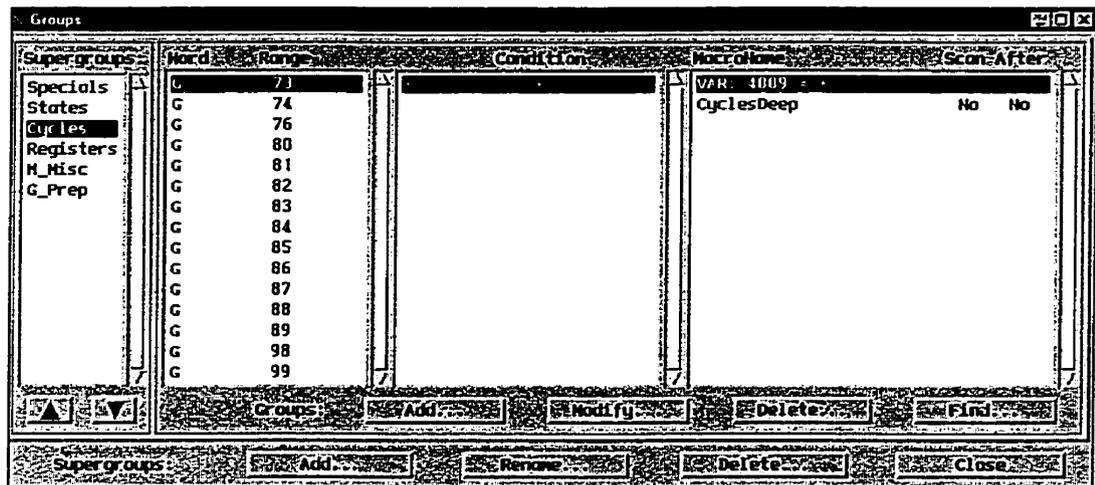


Figure 3.13 Définition des codes G des cycles de la machine

3.5.3 Fichier exécutable des macros

Le fichier exécutable des macros , aussi connu comme le fichier CME, contient les fichiers personnalisés qui contrôlent le traitement des codes G. En plus des macros, ce fichier peut établir les modalités qui fournissent la possibilité de personnalisation des instructions et l'interprétation des codes G, et permet l'accès aux tableaux personnalisés contenant les données de travail et de la machine. Ce fichier peut être créé avec le kit de développement de la machine, et est destiné à des traitements complexes et inhabituels, car il créera de nouvelles instructions qui sont compilées et liées au fichier de la librairie standard de VERICUT.

3.6 Modes de fonctionnement de la machine

Les fonctions de ce menu personnalisent l'interprétation des codes G et la simulation des mouvements de la machine-outil, de façon à établir les mouvements par défaut et les conditions de décalage. La machine est configurée par les modes suivants :

- Skip block : Commande la spécification du caractère du skip block.
- Cercles : Commande l'interprétation des données du cercle (I, J, K) pendant le traitement des blocks de mouvement cercle/hélice (G02, G03).
- Compensation de l'outil : Commande la compensation du diamètre de l'outil.
- Cycles : Commande l'interprétation des cycles canés.
- Debug : Commande la destination des messages de sortie.
- Général : Etablit les grandes lignes de traitement des codes G, le mode d'usinage par défaut, l'arrêt du traitement et procède aux opérations mathématiques.
- Message : Commande la destination et le format des messages de sortie.
- Mouvement : Etablit les états de commande CN et les conditions de mouvement, la quantité des types de mouvement affectant le traitement des cycles canés et la précision utilisée à l'interprétation des valeurs calculées.
- Décalages : Commande le décalage du système de coordonnées.
- Optipath : Commande les séquences des blocks et l'espace de la valeur des mots dans un fichier de chemin de l'outil de codes G optimisé.
- Sous-routines : Commande la référence des noms de sous-routines.
- Outillages : Établit les conditions par défaut des outillages.

3.7 Définition des outils

VERICUT requiert la description des outils de coupe afin de procéder à la simulation du processus d'enlèvement de matière ou détecter les collisions. Les outils sont décrits suivant la spécificité de la machine à simuler. Dans notre cas, les outils de fraisage sont considérés.

L'ensemble de l'outil est constitué de trois composantes, la partie de coupe de l'outil connue comme "outil de coupe", la partie non coupante de l'outil connue comme le corps d'outil "shank" et le porte-outil "holder", qui soutient la partie outil de coupe/corps d'outil et qui relie l'ensemble à la broche de la machine.

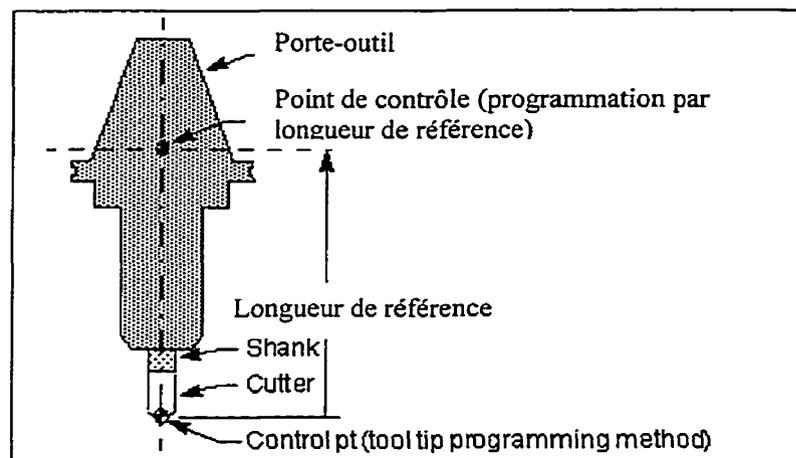


Figure 3.14 Les composantes de l'outil de fraisage

En général, les outils de fraisage assurent la conduite des pointes de l'outil (tool tip). Cependant, la relation peut être changée suivant la méthode de programmation, les commandes de mouvements et la localisation de l'outil relative au point de contrôle.

3.7.1 Procédure de définition de l'outil

Dans le groupe "tool manager" du menu "outils", le profil et les caractéristiques de l'outil de coupe (la partie coupante de l'outil) sont définies à l'aide des icônes de la figure 3.15 suivant les stipulations du programme (codes G).

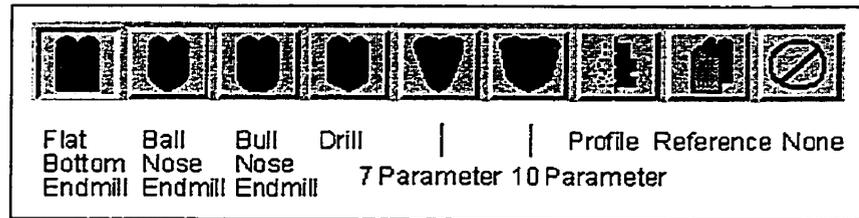


Figure 3.15 Icônes des outils de coupes de fraisage

Le tableau 3.2 décrit les principaux paramètres de définition d'outils. Les formes des outils de fraisage sont définies suivant 7 paramètres tel qu'illustré à la figure 3.16.

Tableau 3.2

La définition des outils de coupe

Type de la primitive :	Défini par :
Ball Nose End Mill	Le diamètre (D) et la hauteur (H)
Bull Nose End Mill	Le diamètre (D), le rayon du coin (R) et la hauteur (H)
Flat Bottom End Mill	Le diamètre (D) et la hauteur (H)
Drill	Le diamètre (D), l'angle de pointe, en degrés (A) et la hauteur (H)

La liste des dix paramètres de définition de l'outil est comme suit :

D : Le diamètre, mesuré du point d'intersection du côté de l'outil de coupe avec le bottom flat.

R : Rayon des coins.

E, F : Décalage de l'axe et de la pointe de l'outil vers le centre du rayon du coin.

A, B : Angle de base et angle de coté, en degrés.

H : Hauteur (longueur de la lèvre de coupage)

R, E & F : Paramètres fournis pour la description du second rayon de coin.

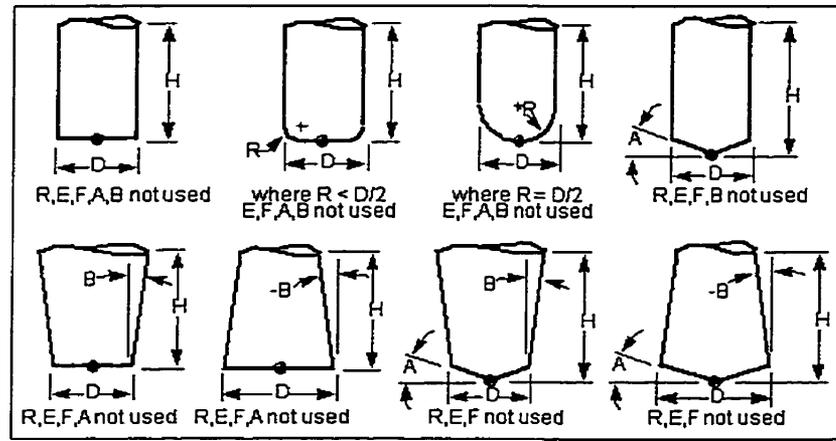


Figure 3.16 Formes des outils de coupe de fraisage [4]

3.7.2 Définition du "corps d'outil" et du "porte-outil"

La détermination du "corps d'outil" et du "porte-outil" se fait à l'aide des icônes conçues à cet effet. Des exemples sont illustrés aux figures 3.18 et 3.19 respectivement.

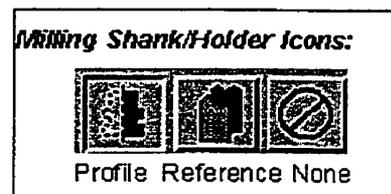


Figure 3.17 Icônes des corps d'outil et porte-outil de fraisage

Le profil du corps d'outil décrit la portion non coupante de l'outil. Les corps d'outils sont utilisés essentiellement dans la détection des collisions entre la partie non coupante et la pièce brute. Ils peuvent aussi représenter les extensions du porte-outil, et des adaptateurs qui sont définis séparément de ce dernier.

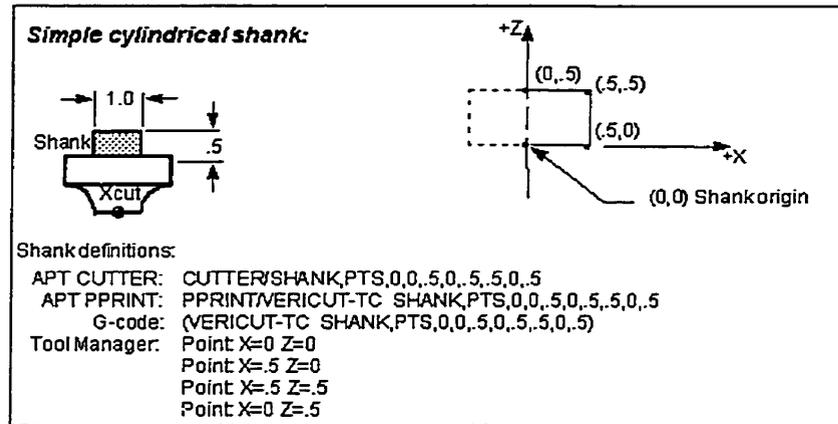


Figure 3.18 Exemple de détermination d'un corps d'outil.

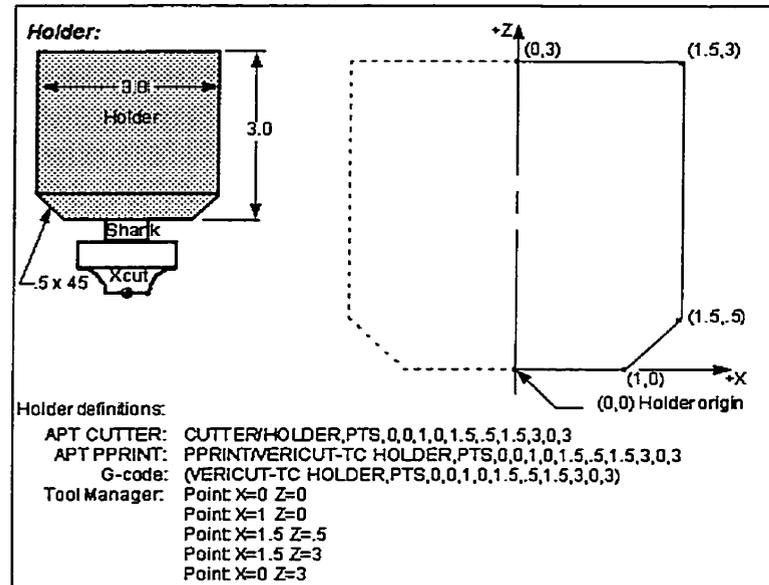


Figure 3.19 Exemple de détermination d'un 'porte-outil'

Le profil “porte-outil” décrit la partie qui maintient l’ensemble outil de coupe/corps d’outil pour une éventuelle utilisation sur la machine. Il est utilisé pour détecter les collisions du “porte-outil” avec la pièce brute ou les fixations.

3.8 Utilisation des tables

La spécification des mouvements de la machine pendant les changements d’outils, le retour au point de référence et le positionnement initial sont déterminés par le stockage des données du registre des décalages (offset) de la commande de la machine CN, et la description des relations entre les systèmes de coordonnées utilisés par les différents styles de programmation des machines CN.

- Offset du travail de base (base work offset) : Cette table spécifie le lieu de base des offsets de travail, elle est utilisée quand le fichier de chemin d’outil prend comme référence le système des coordonnées du travail de base (ex : G52), à partir duquel tous les autres offsets de travail sont basés.
- Compensation de l’outil de coupe : Elle stocke les valeurs de décalage des compensations des diamètres d’outils de coupe (CDC). Son utilisation est liée à la simulation de la compensation de diamètre de l’outil de coupe (ex : G41-G42).
- Offset de la longueur de référence (Gage offset) : Les valeurs des longueurs d’outils sont enregistrées, et permettent la conduite d’un point de contrôle autre que le bout de l’outil suivant les stipulations de la méthode de programmation du chemin de l’outil utilisée.
- Localisation initiale de la machine : Elle spécifie la localisation initiale des axes de la machine. Après une simulation ou après l’avoir annulée, le retour du chemin de l’outil positionne la machine à cette localisation.

- L'entrée zéro du programme : L'entrée zéro du programme (input program zero) spécifie la localisation zéro programmée des données de la CN (figure 3.20). Elle est requise quand l'opérateur de la machine établit une localisation zéro différente du zéro de la machine.

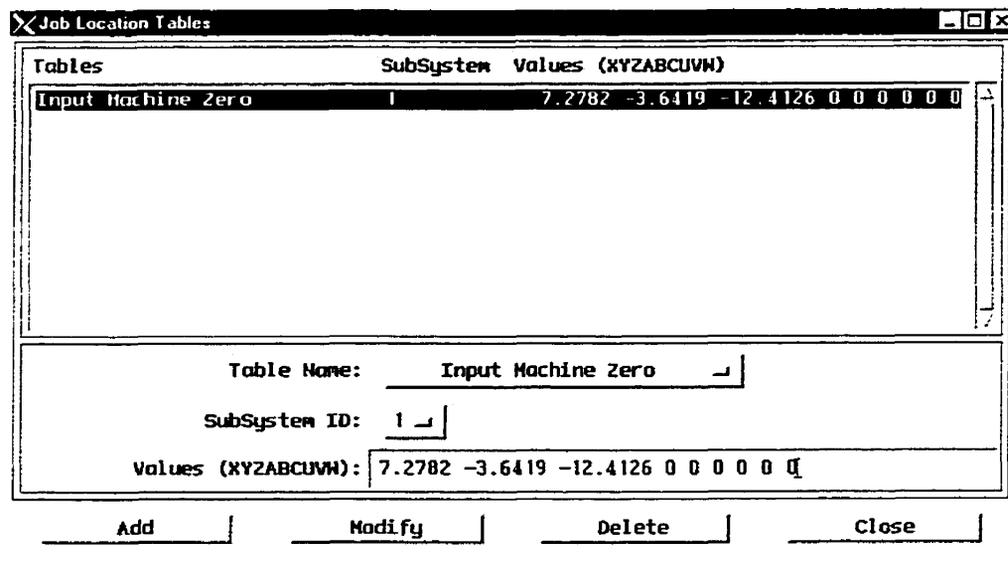


Figure 3.20 L'entrée zéro du programme

- Localisation de référence de la machine : La localisation de la référence est le lieu où les axes de la machine suivent un mouvement jusqu'au point où la commande exige un retour au point de référence (ex : G28).
- Transformation de la pièce : Elle envoie l'information de la transformation de la pièce brute vers le module VERICUT. Utilisant cette table et les macros de la transformation de la pièce, il devient possible de tourner, bouger et inverser les vues du modèle VERICUT pendant le processus de simulation.

- Probe offset : Elle stocke les valeurs “probe tool offset” et ajuste les mouvements de l’outil pendant la simulation de la pièce (part probing).
- RTCP Pivot Offset : Elle spécifie le décalage entre l’origine de l’outil et l’inclinaison du point de pivot de la tête. Cette table est utilisée quand la composante en rotation située entre l’outil et la base a une entrée initiale de la localisation de la machine autre que zéro. Ceci indique que la localisation initiale de la composante en rotation oriente l’axe de l’outil suivant les exigences de la compensation RTCP.
- Localisation du changement d’outil (tool change location) : Cette table spécifie la localisation des axes de la machine quand l’outil est changé. Elle est utilisée quand les axes de la machine sont spécifiés pour le retrait pendant le changement d’outil via la méthode de retrait de changement de l’outil (Tool change retraction).
- Retrait de changement d’outil : Elle indique quels axes de la machine sont rétractés pendant le changement d’outil.
- Index des outils : L’index des outils sert à stocker les numéros d’outils des chemins d’outil (codes G) dans une librairie.
- Compensation de longueur d’outil : L’utilisation de cette table est optionnelle quand le chemin de l’outil programmé utilise cette méthode (ex. G43 Hn).
- Tool Nose Compensation : Elle spécifie les valeurs de décalage de la compensation du diamètre de l’outil (ex. G41 – G42 Dn).
- Offsets du travail: Elle stocke les valeurs de décalages du système de coordonnées (fixture offset). Cette table est utilisée quand la programmation de la simulation se fait avec le work offset (ex. G53 – G59).

3.9 Édition du fichier du chemin de l'outil

L'ouverture du fichier du programme du chemin de l'outil se fait à l'aide de "Toolpath File" contenu dans "Edit" du menu "File". Un éditeur de texte permet d'afficher le fichier du chemin d'outil courant. Ce même fichier peut être visualisé dans la fenêtre " toolpath" du menu "Info". Durant la simulation, cette fenêtre permet de voir les différentes étapes du déroulement du programme.

3.10 Définition de la tâche

La définition de la tâche (figure 3.21) est l'élément central du simulateur. Elle permet le contrôle du chemin de l'outil, la méthode de programmation, la définition du fichier de sortie, ainsi que le fichier log.

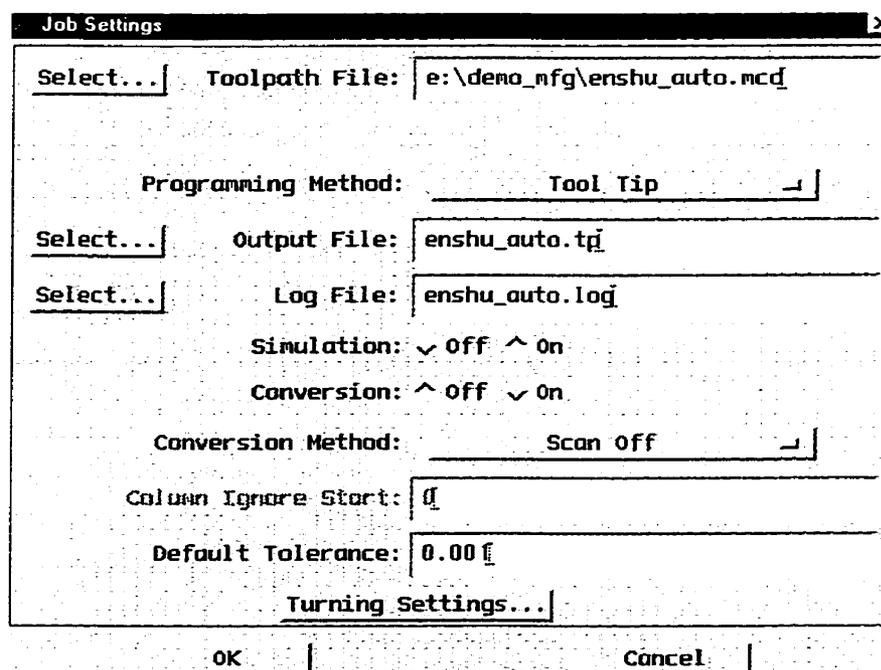


Figure 3.21 Fenêtre de définition de la tâche.

3.10.1 Fichier du chemin de l'outil

Avec une extension (.mcd), ce fichier est connu comme le fichier d'entrée (input file) contenant les blocs de données du code machine à être traités.

Typiquement, les codes G contiennent les paires de mots ou mot/valeur conformes au standard EIA, de format RS – 274. VERICUT traite trois types de codes G, de base, intermédiaires et complexes. Les codes G de base sont habituellement faciles à traiter car ils accèdent seulement aux formes de base du contrôle CN. Les codes G intermédiaires par contre accèdent aux formes de contrôle CN qui affectent le mouvement et la simulation comme la compensation du diamètre et de la longueur de l'outil, les décalages de la fixation, la programmation en longueur de référence, etc. Les codes G complexes utilisent des contrôles avancés tel que les variables, looping, macros, les sous-programmes, etc.

3.10.2 Méthode de programmation

Elle décrit la relation entre les outils de coupe et ce qui est en train de s'accomplir par les commandes de mouvements dans le fichier du chemin de l'outil. La sélection de la méthode de programmation doit refléter la relation entre les outils de coupe et le point de contrôle conduit par le chemin de l'outil.

- La pointe de l'outil (Tool tip) : Cette méthode de programmation permet de conduire le centre de la pointe de l'outil.
- Gage length : Elle conduit le point de référence de la longueur de référence, où la ligne d'axe de l'outil croise la ligne de référence de la machine.

- Compensation de la longueur de l'outil : Elle conduit la pointe de l'outil quand la compensation de la longueur est active et conduit la référence de la longueur de référence "gage" quand la compensation de la longueur est annulée.

3.10.3 Fichiers de sortie

Avec une extension typique (.tp), il constitue le premier fichier de sortie qui stocke les données résultant de la conversion du programme (codes G). Prêt à être utilisé par VERICUT, ce fichier est de type APT ASCII et peut être visualisé ou édité en utilisant la commande "output". Le deuxième fichier (.log) comporte les messages d'erreurs, les avertissements et les informations concernant le traitement du programme.

3.10.4 Détection des collisions

En activant la fonction de détection des collisions (figure 3.22), les composantes entrant en collision prennent des couleurs caractéristiques (typiquement rouge) signalant l'erreur. Cette dernière est enregistrée dans le fichier log pour identifier le block du programme et les composantes mises en cause. Des tolérances peuvent être spécifiées. Il est aisé de considérer que les collisions entre la pièce brute et l'outil sont ignorées.

Components	Tool	broche	Z	y	Stock	Fixture	X
Base	0.1	0.1	0.1	0.1	0.1	0.1	0.1
X	0.1	0.1	0.1	0.1	0.1	0.1	
Fixture	0.1	0.1	0.1	0.1	0.1		
Stock	0.1	0.1	0.1	0.1			
y	0.1	0.1	0.1				
Z	0.1	0.1					
broche	0.1						

Default Near Miss: 0.1 Set All

OK Cancel

Figure 3.22 Fenêtre des tolérances de détection des collisions

3.11 Analyse de la simulation

Le menu "Analyse" permet l'inspection du modèle et de la machine, aussi bien que la géométrie du modèle nominal relativement au modèle en cours d'usinage. Il comporte trois fonctions principales :

- Historique : Cette fonction relate l'historique du programme, communément utilisée pour déterminer la source de l'erreur détectée par VERICUT. Elle comporte les informations de l'usinage incluant le nom, les nombres séquentiels et l'enregistrement du texte du programme, le sous-système et l'identité de l'outil responsable de l'enlèvement de matière, ainsi que le type de mouvement simulé.
- X-Caliper : Elle assure l'exécution des données de mesure et autres informations mathématiques. Ces capacités incluent la largeur du matériel, les angles et les distances entre les formes et les figures des éléments.
- Auto- Diff : Comme son nom l'indique, cette fonction sert à comparer les données du modèle nominal avec le modèle en cours d'usinage. Elle détecte les excès et les manques de matière.

3.12 Optipath

Inclus dans le menu "Applications" du module de base, "Optipath" permet de créer un chemin d'outil optimisé en recalculant les vitesses d'avance et les vitesses de la broche. Cette optimisation ne changera pas la trajectoire de l'outil, mais assurera des vitesses optimales afin de produire une pièce finie de bonne qualité tout en diminuant le temps de coupe. Il existe des codes optimisables et d'autres non optimisables. Le tableau 7 à l'annexe C montre les spécifications de ces codes.

Les étapes de l'opération d'optimisation sont comme suit :

- Spécification du programme d'usinage ".mcd".
- Définition des caractéristiques du matériau de la pièce brute, de la machine et la description des outils dans le menu "Applications".
- Définition des paramètres d'optimisation (volume de coupe, vitesse, etc).
- La mise en route de la simulation avec "Optipath" déclenché, fournissant ainsi un fichier de sortie (.tpo) représentant le programme optimisé.
- Lecture des résultats concernant le temps d'usinage, les changements d'outils et les erreurs dans le fichier log ou dans la fenêtre "status".
- Installation du programme optimisé (.tpo) dans le menu "Toolpath".
- Re-simulation et vérification des différences.

3.13 Le simulateur fin prêt

Suivant la définition des paramètres et la méthodologie de construction citées dans ce chapitre, la machine ainsi développée s'est basée sur deux éléments essentiels :

- L'assemblage des composantes, où la fraiseuse Enshu VMC-40 est physiquement conçue et fidèlement représentée.
- L'introduction des fichiers de commandes respectant le contrôleur FANUC OM-A dotant la machine, avec la spécification des codes, macros et de tous les mots et groupes précités.

Le chapitre suivant permettra l'introduction des autres éléments, tel que les outils et les programmes (codes G) dans des applications et des exemples, ainsi qu'une brève comparaison entre les simulations avec PRO/ENGINEER et VERICUT.

CHAPITRE 4

VALIDATION DU SIMULATEUR

Ce chapitre démontre l'utilisation des outils présentés dans les chapitres précédents. Le simulateur est validé par une multitude d'essais, allant des applications théoriques aux modèles pratiques usinés sur la machine. Le choix des simulations est basé sur la variété des séquences et des types de programmation afin d'illustrer la précision, la robustesse et la fiabilité du simulateur.

Parmi les fichiers constituant le simulateur décrit au chapitre précédent, on distingue deux types de données principales :

1. Les données fixes applicables à tous les modèles. Ils incluent les fichiers suivants :
 - Le fichier (.mch) : Une partie de ce fichier, incluant les composantes de la machine "base, colonne, table, porte-broche et broche", représente des éléments faisant partie des données permanentes de la machine et par conséquent applicables à tous les modèles.
 - Le fichier (.ctl) : Il représente les données de commande de la machine respectant le contrôleur FANUC OM-A. Quelques distinctions sont à signaler entre les modèles quant à l'affinement du contrôle dans le menu "modals" (faisant partie de ce fichier) relativement aux programmes (codes G).

2. Les données variables propres à chaque modèle. On peut citer :

- Le fichier (.tls) représentant les outils.
- Le fichier (.mcd) représentant le programme (codes G).
- Les fichiers de sortie (.tp) et (.log).
- Les composantes dans les fichiers (.mch) représentant la pièce brute et les éléments de montage ainsi que leurs localisations géométriques sur la machine pour chaque modèle.

4.1 Description sommaire des essais

La sélection des essais est basée sur la variété des types d'usinage. Il englobe les différents chemins d'outil pouvant être ou ayant été rencontrés lors des processus de fabrication réels. Les tableaux 8 et 10 à l'annexe D illustrent les spécifications techniques des modèles et l'identification des fichiers ayant fait l'objet d'essais dans VERICUT. La description des essais appliqués au simulateur est comme suit :

Essai 1 (Attache) :

Cet essai présente un aperçu de la simulation avec PRO/ENGINEER et sa mise en application dans VERICUT avec la définition des différents paramètres, du montage et de la simulation. Des hors courses et des collisions sont provoquées afin de tester la capacité du simulateur à les détecter. La comparaison du modèle nominal avec le produit final et l'optimisation du programme sont également présentées.

Essai 2 (Moule) :

Cet exemple met l'accent sur la démarche imagée de définition du brut et des outils dans VERICUT, avec les résultats de la simulation.

Essai 3 (Maquette d'automobile) :

Généré par PRO/ENGINEER, ce moule est en forme de voiture. L'usinage comporte une séquence de dégrossissage et une séquence de finition. Il met en relief la précision du simulateur et sa capacité de respecter les chemins d'outils. Les résultats de l'optimisation du programme d'usinage sont intéressants.

Essai 4 (Plaque) :

L'essai comporte plusieurs séquences d'usinage d'un couvercle. Développée manuellement et destinée à être usinée sur la machine Enshu, cette pièce appuie la théorie réaliste du simulateur. Le programme est revu et corrigé suite à la détection de codes non supportés par la machine et de défauts sur la surface du produit fini.

Essai 5 (Poulie) :

Cette pièce a été développée avec PRO/ENGINEER et usinée sur la machine Enshu dans le cadre d'un projet étudiant. L'essai d'une partie de l'usinage démontre la robustesse et l'exactitude des différents éléments du simulateur.

4.2 Essai 1 : Usinage d'une attache

L'exécution de ce modèle se compose de plusieurs séquences d'opérations, soit le profilage du contour, la coupe pochetée et le perçage de trous.

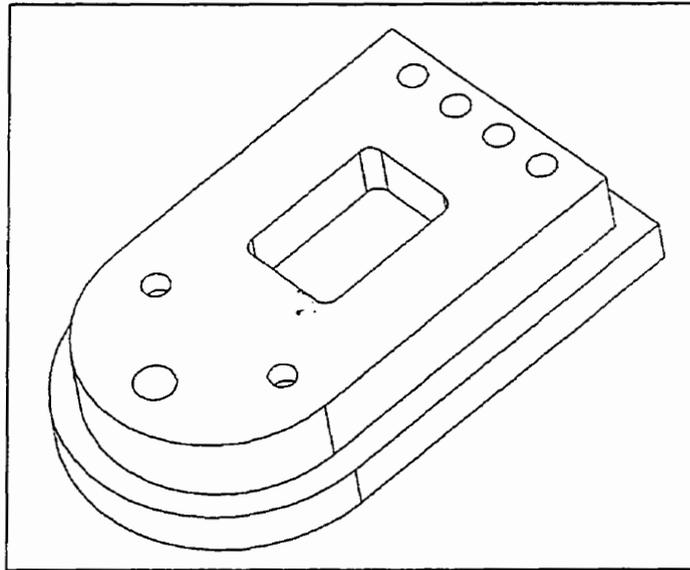


Figure 4.1 Attache

Le système de CAO/FAO PRO/ENGINEER permet de générer les trajectoires d'outils selon différentes stratégies compatibles aux opérations à effectuer. Les trajectoires peuvent être visualisées en mode filaire ou en mode ombragé tel qu'illustré à la figure 4.2. Même si le logiciel est en mesure de générer un fichier en code G compatible à la machine par le biais du post-processeur, la simulation de l'enlèvement de la matière est effectuée à partir de l'information produite par le système, soit les trajectoires compensées de type CL (position du centre de l'outil).

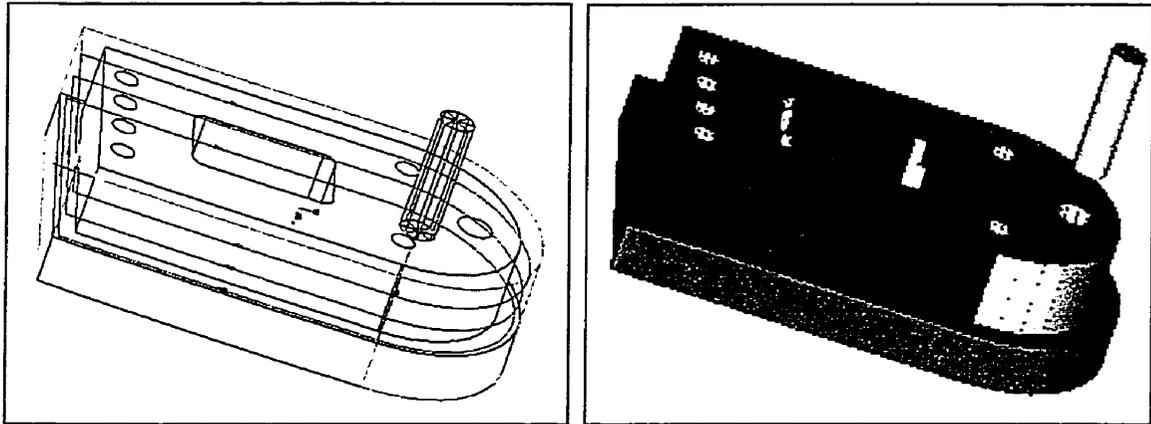


Figure 4.2 Le modèle fini usiné par PRO/MANUFACTURING.

4.2.1 Définition de l'usinage et des paramètres de coupe dans VERICUT

Tel que déjà mentionné, une fois que les composantes de la machine et le modèle de la pièce sont modélisés, la fonction "export" de PRO/ENGINEER permet de les transformer en format STL. Ce format exige des liaisons parfaites entre les surfaces des modèles 3D. Il est composé de facettes triangulaires avec leurs normales associées avec la capacité de décrire approximativement n'importe quelle forme.

Une fois la première étape terminée, c'est à dire la machine construite et la pièce brute fixée sur le montage, on procède à la définition de la tâche. Celle-ci comporte la spécification dans la boîte de dialogue du programme d'entrée (attache.mcd) et des fichiers de sortie (attache.tp et attache.log) où Vericut peut insérer les informations relatives à l'usinage. Le programme est de type pointe d'outil "tool type" et est également identifié dans cette boîte de dialogue. La figure 4.3 montre une partie du programme.

```

%
O0001;
N1 G20;
N3 G90 G40 G49 G80;
N5 G91 G28 Z0.0;
N7 G28 X0.0 Y0.0;
N9 T01;
N11 M06;
N13 G90 S2546 M03;
N15 G00 X5.5118 Y0.0984;
N17 Z0.7874;
N19 Z0.0394;
N21 G94 G01 X5.5118 Y0.0984 Z-0.3937 F68.19;
.....

N221 M05;
N223 G91 G28 Z0.0;
N225 G28 X0.0 Y0.0;
N227 T00;
N229 M06;
N231 M30;
%
```

Figure 4.3 Aperçu du programme d'usinage

L'usinage de ce modèle nécessite trois types d'outils. Ils sont définis suivant leurs fonctionnalités et leurs caractéristiques dans le menu (outils) en se basant sur les spécifications déjà déterminées dans Pro/Manufacturing. L'ajout des éléments de soutien des outils comme le "porte-outil" et le "corps d'outil" peut se faire à l'aide de la librairie contenue dans VERICUT ou introduit suivant les modèles réels et spécifiques existants. La longueur de référence de l'ensemble de l'outil spécifiée dans le menu "outil" est la somme des hauteurs du porte-outil, du corps d'outil et de la partie apparente de l'outil de coupe. Elle est automatiquement prise en considération par VERICUT et déduite de la distance séparant le zéro de la machine du zéro du programme représentant le point de départ de l'usinage sur la pièce brute.

Ayant les paramètres définis et le brut ajusté entre les deux parties de la fixation, comme illustré à la figure 4.4, on procède à la simulation de l'usinage (figure 4.5).

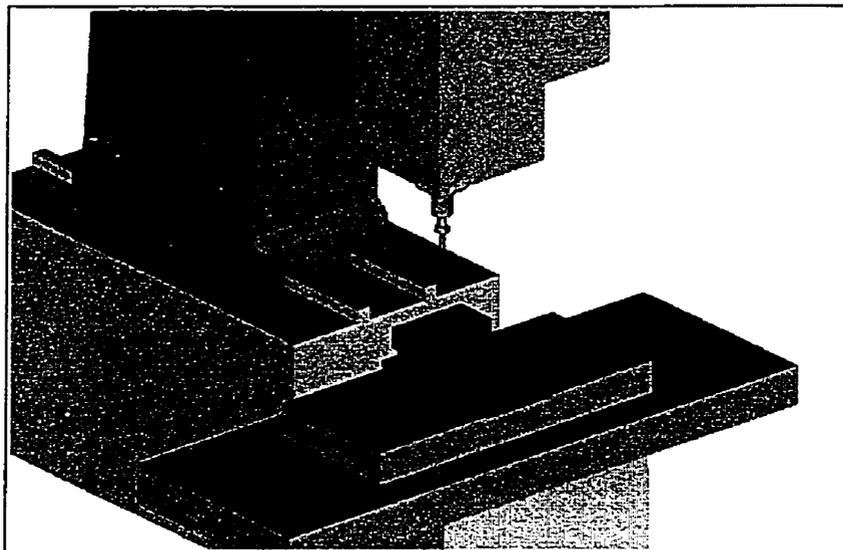


Figure 4.4 Le montage fin prêt pour la simulation de l'usinage

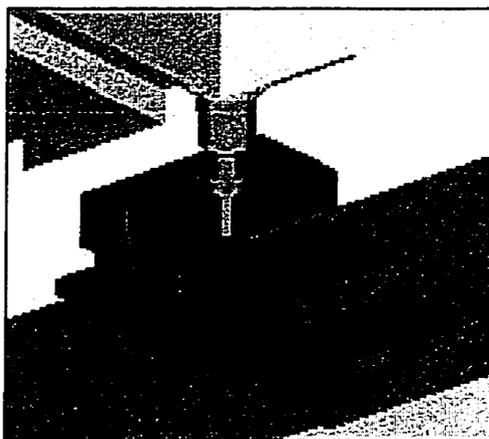


Figure 4.5 La simulation pendant la séquence de profilage.

Étant rassuré de la conformité et de la bonne marche de la simulation (pas de hors courses ou de collisions), la visualisation de la coupe de la pièce peut s'effectuer à l'aide du module de base de Vericut. Il s'agit de créer un fichier de format (.usr) en se basant sur le fichier du logiciel (Vericut.usr) dans le module de base de VERICUT, ainsi que

de définir les paramètres, le programme et la pièce brute. La position relative du point de départ du chemin d'outil par rapport au zéro de la machine ou à l'origine du modèle doit être également spécifiée. Dans le module de base de VERICUT, on a l'avantage de pouvoir visualiser la simulation sur deux fenêtres distinctes, L'une représentant la simulation de l'ensemble de la machine et l'autre représentant seulement le brut tout en montrant l'enlèvement de matière avec tous les détails pertinents. Cette dernière donne une idée précise du produit final résultant de l'usinage.

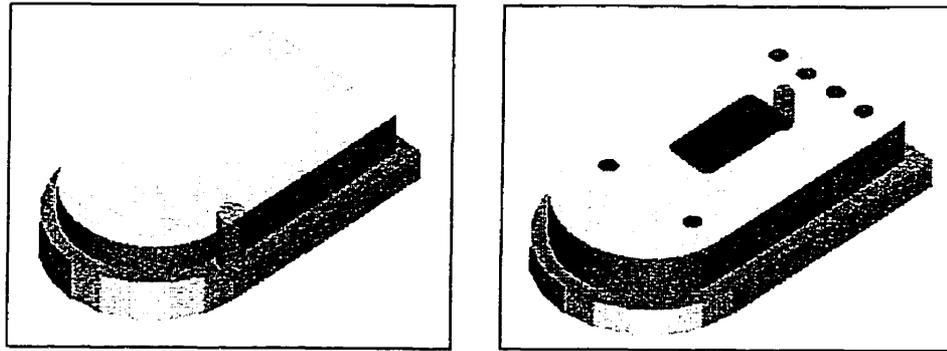


Figure 4.6 Étapes d'usinage visualisées dans le module VERICUT

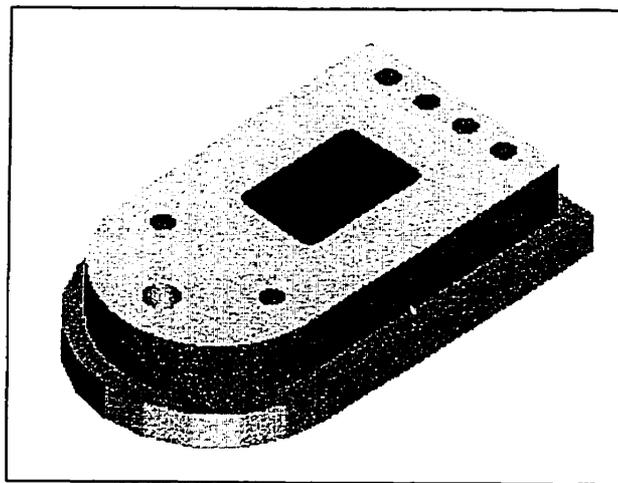


Figure 4.7 Pièce finie visualisée dans le module Vericut

4.2.2 Simulation des hors courses et collisions

Pour les besoins de l'étude, on procède à la provocation des avaries pouvant survenir réellement sur une machine dans un atelier d'usinage. Pour ce faire, un changement dans le programme (codes G) ou un décalage du montage de l'ensemble brut-fixation peut engendrer des défaillances et par conséquent des messages d'erreurs.

Par exemple, en changeant la valeur 5.5118 de X par 20.0 à la ligne N21 G94 G01 X5.5118 Y0.0984 Z-0.3937 F68.19; du programme de la figure 4.3, un chiffre dépassant la limite de course de la machine sur cet axe, on observe, une fois la simulation arrivée à ce stade, une ouverture automatique de la fenêtre des messages d'erreurs, nous indiquant l'endroit précis du dépassement de la course et/ou de la collision. Dans le cas d'une collision, les éléments entrant en contact prennent la couleur rouge assignée à cet effet. Pour cette avarie simulée, la figure 4.8 montre le changement de couleur de l'élément concerné par le dépassement de la course et la figure 4.9 montre le message d'erreur.

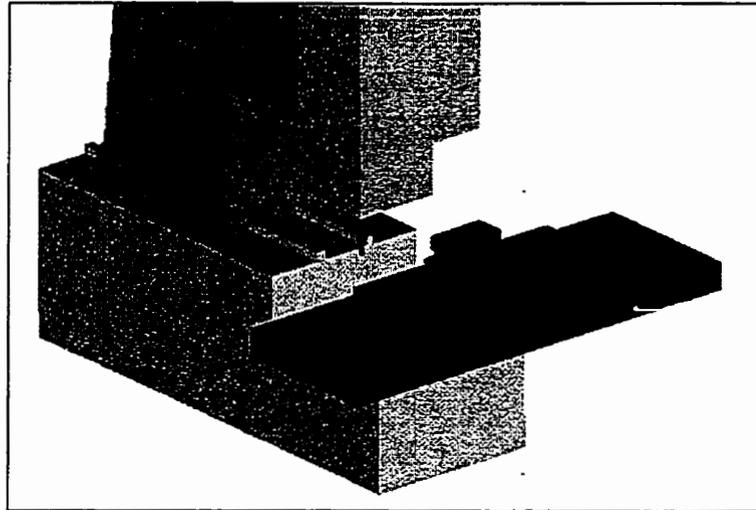


Figure 4.8 Dépassement de la course suivant l'axe X.

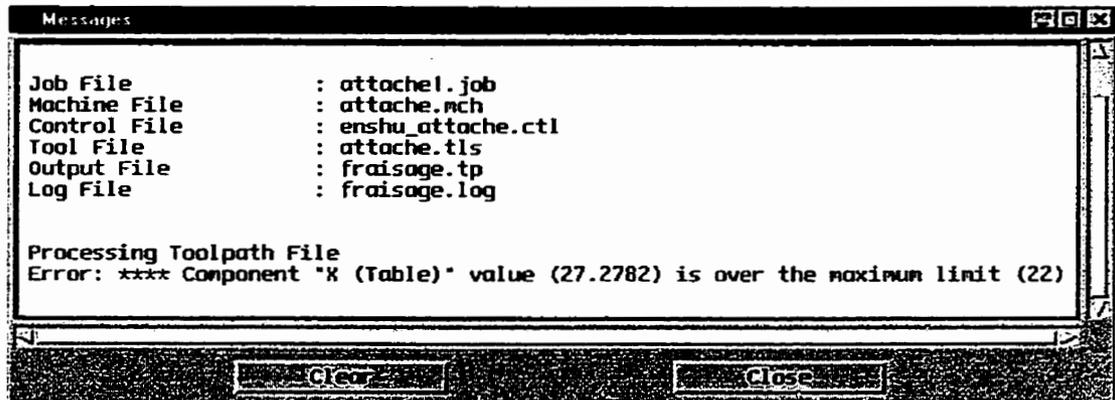


Figure 4.9 Fenêtre des messages indiquant une hors course.

Pour la simulation d'une collision, le changement de la valeur de (Z-0.3937) sur la même ligne du programme par (Z-2.0) donne une série de collisions entre l'outil de coupe et la fixation, tel que montré sur les figures 4.10 et 4.11.

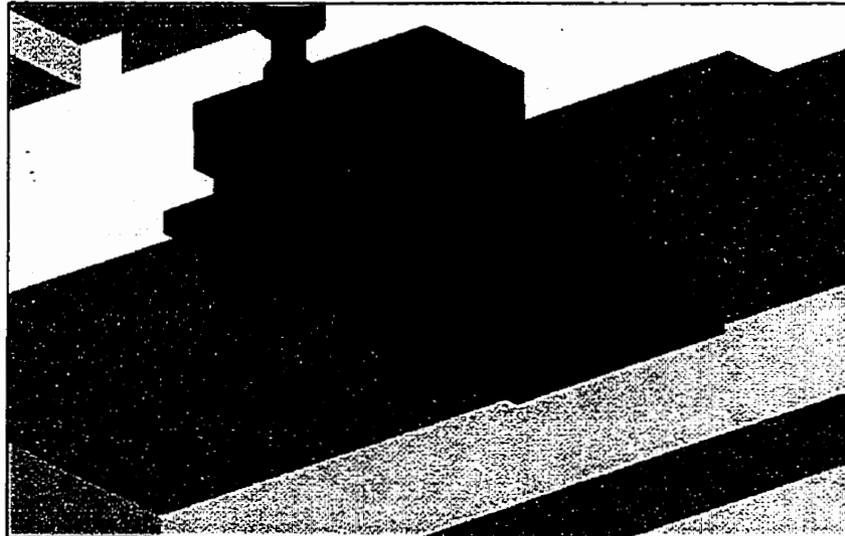


Figure 4.10 Collision entre l'outil de coupe et la fixation.

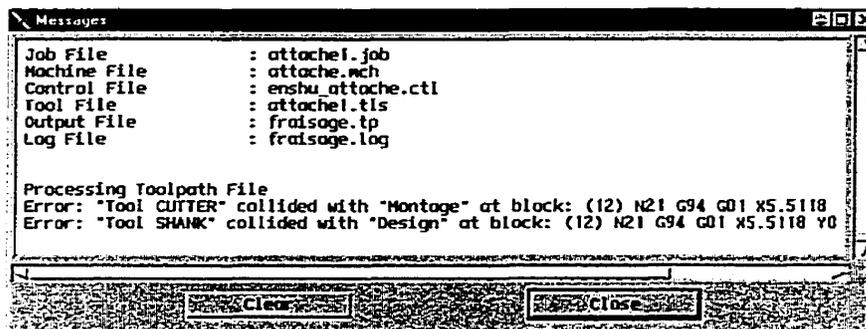


Figure 4.11 Fenêtre des messages indiquant des collisions.

4.2.3 Comparaison du modèle simulé et du modèle nominal

La comparaison des modèles simulé et nominal se fait à l'aide du module de comparaison "Autodiff". En superposant le modèle nominal de la pièce et le brut, la fonction "Autodiff" (figure 4.12) permet de détecter les excès et les manques de matière dans la pièce finie, ces derniers ayant été provoqués pour les besoins de l'étude.

Les résultats de la comparaison sont présentés à la figure 4.13. La figure (a) illustre les excès de matière, alors que la figure (b) illustre les manques de matière détectés par le module de comparaison.

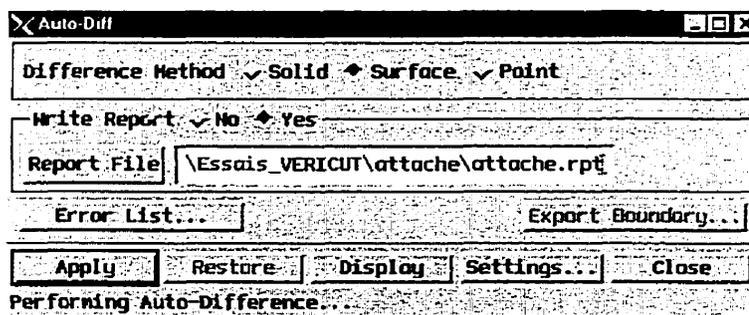


Figure 4.12 Boite de dialogue de la fonction "Autodiff"

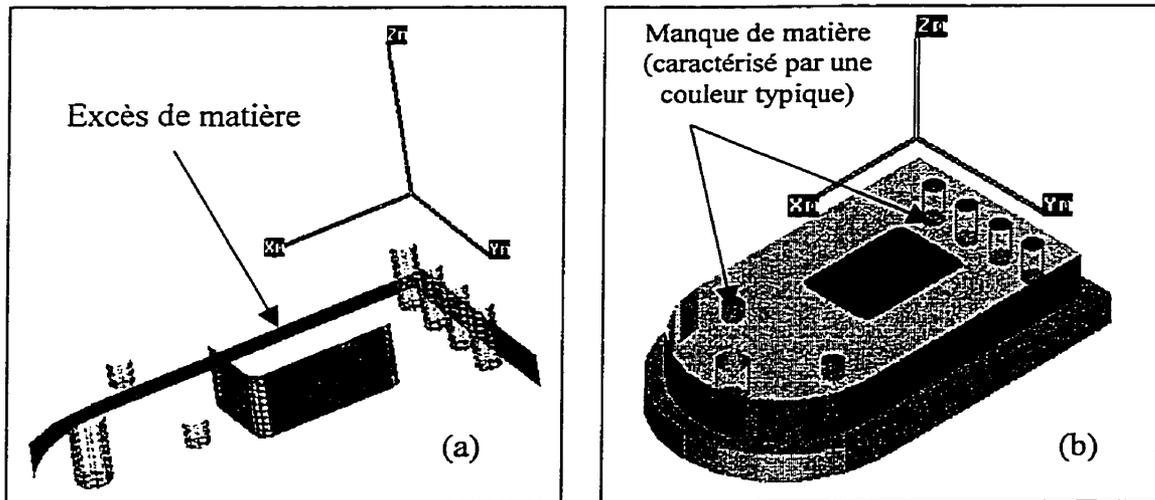


Figure 4.13 Résultats de la comparaison avec "Autodiff"

En plus du résultat graphique, la fonction de comparaison permet de ressortir la liste des déviations et des imperfections et également de préciser les blocs défaillants du programme (figure 4.14). Un rapport plus détaillé de ces erreurs (attache.rpt) est fourni à la fin de la simulation.

Type	Record Number	Deviation
Gouge	13	-0.0991
Excess	15	0.0967
Excess	16	0.1249
Gouge	18	-0.0991
Excess	19	0.0977
Excess	20	0.0967
Excess	21	0.1249
Gouge	22	-1.1811
Gouge	23	-1.1811
Excess	24	0.2429
Excess	25	0.1545
Gouge	26	-1.1811
Gouge	35	-0.1009
Excess	37	0.0997
Gouge	37	-0.1009
Gouge	39	-1.0060
Gouge	44	-1.0099
Excess	47	0.9841

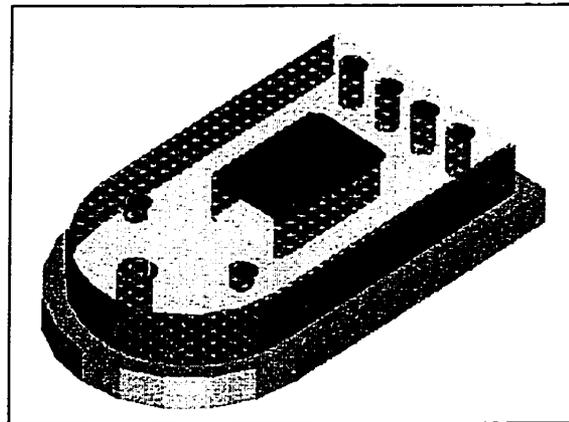


Figure 4.14 Les excès et les manques de matière

4.2.4 Optimisation du chemin d'outil

Après avoir suivi la procédure d'optimisation du chemin d'outil telle qu'expliquée dans le chapitre précédent et illustrée aux figures 4.15 et 4.16, le résultat est représenté par le nouveau programme attache.tpo.

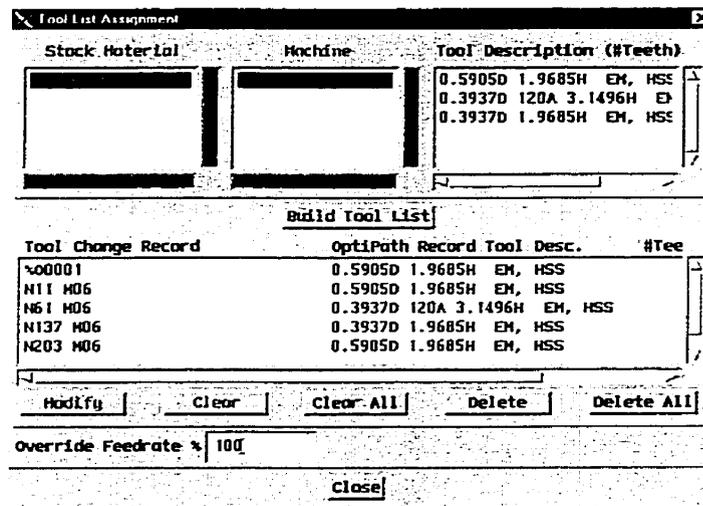


Figure 4.15 Assignation des outils et autres caractéristiques.

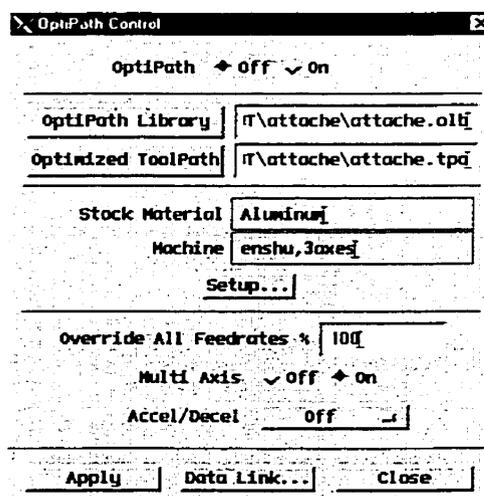


Figure 4.16 Fenêtre de contrôle du module "optipath"

Les figures 4.17 (a) et (b) montrent le fichier log décrivant les résultats de la simulation du programme original et du programme optimisé respectivement. L'optimisation a permis une réduction de temps d'usinage d'environ une minute, soit 28% du temps original. Ce résultat est purement théorique et n'offre aucune certitude quant au respect des tolérances.

TOOLPATH ERROR REPORT (a)					
Error for line 74 N145 Z-0.9449					
Fast feed rate removed 0.112049 units of material at record sequence number 74					
Current Tool : Seq# 4, Record # 70, Record : N137 M06					

Seq#	Tool Change Record	Tool Summary			
		Line#	Library ID	Height	Time
1	%O0001	1	1	1.9685	0
2	N11 M06	7	1	1.9685	1.56161
3	N61 M06	32	2	3.1496	0.51856
4	N137 M06	71	3	1.9685	1.51315
5	N203 M06	104	1	1.9685	0.306396
6	N229 M06	104		0	0
Total Time					3.89972

Tool Summary (b)					
Seq#	Tool Change Record	Line#	Library ID	Height	Time
1	N12 M06	7	1	1.9685	1.04079
2	N66 M06	34	2	3.1496	0.493446
3	N142 M06	72	3	1.9685	0.974198
4	N406 M06	204	1	1.9685	0.306396
5	N432 M06	204		0	0
Total Time					2.81483

Figure 4.17 Résultats de l'optimisation

La fenêtre "status" permet selon sa configuration la visualisation en temps réel du déroulement des séquences, du temps d'usinage, de la vitesse d'avance, des changements d'outils et de nombreuses autres caractéristiques d'usinage. Les figures 4.18 et 4.19 illustrent les résultats à la fin de la simulation des deux programmes.

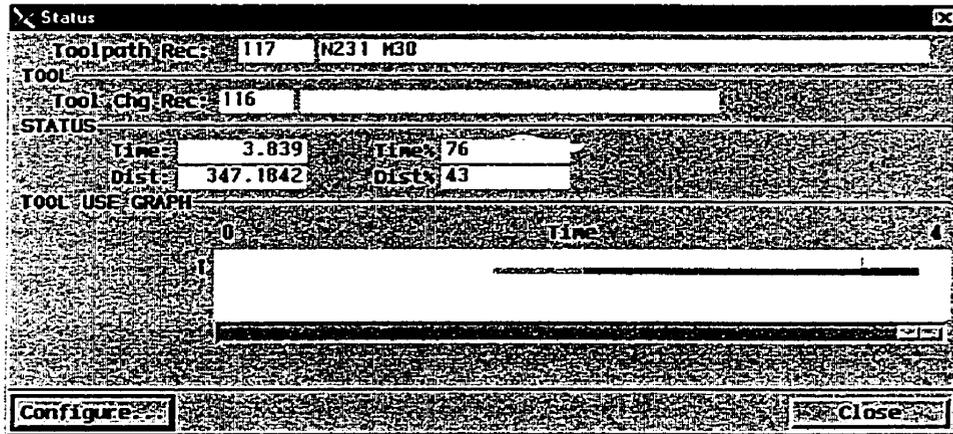


Figure 4.18 Résultat de la simulation du programme original

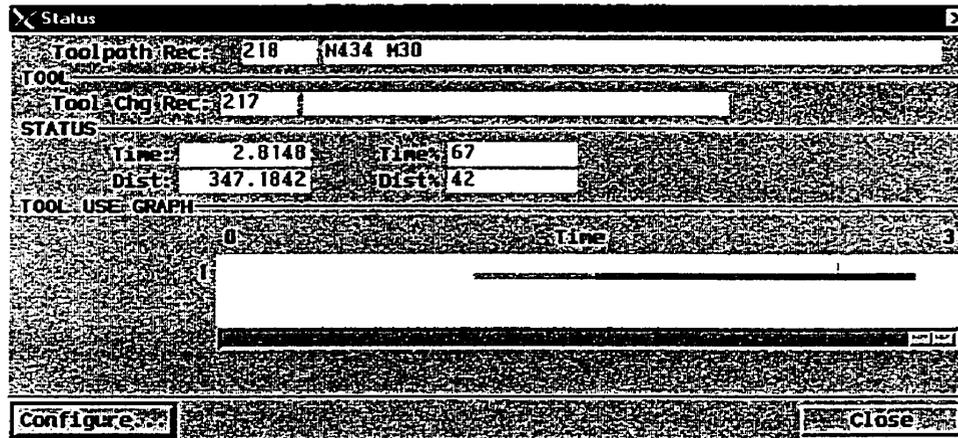


Figure 4.19 Résultat de la simulation du programme optimisé

4.3 Essai 2 : Usinage d'un moule

Le deuxième modèle étudié est un moule (figure 4.20). Les démarches du montage, la définition des outils et du programme sont les mêmes que pour l'essai 1.

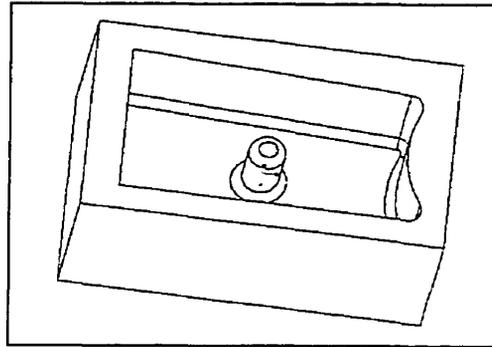


Figure 4.20 Moule

La définition du brut est faite à l'aide des primitives de VERICUT. Ses dimensions (10.25, 6.25, 5.125) sont spécifiées suivant X, Y et Z respectivement. Tel qu'illustré à la figure 4.19, le point de départ du chemin d'outil a comme référence les axes du modèle, et se situe à (6, 3.39, 20.92) de l'origine du modèle. Le montage du brut dans le simulateur est suivi par la définition du modèle dans la fenêtre de définition de la tâche.

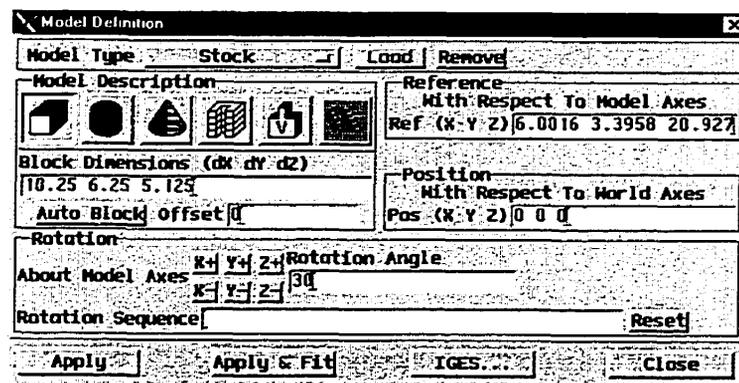


Figure 4.21 Définition du modèle

Dans le tool manager du menu outil, l'introduction des ensembles d'outils est spécifiée selon la figure 4.22. L'étape de l'introduction du programme complet ou des séquences est illustrée aux figures 4.23 et 4.24.

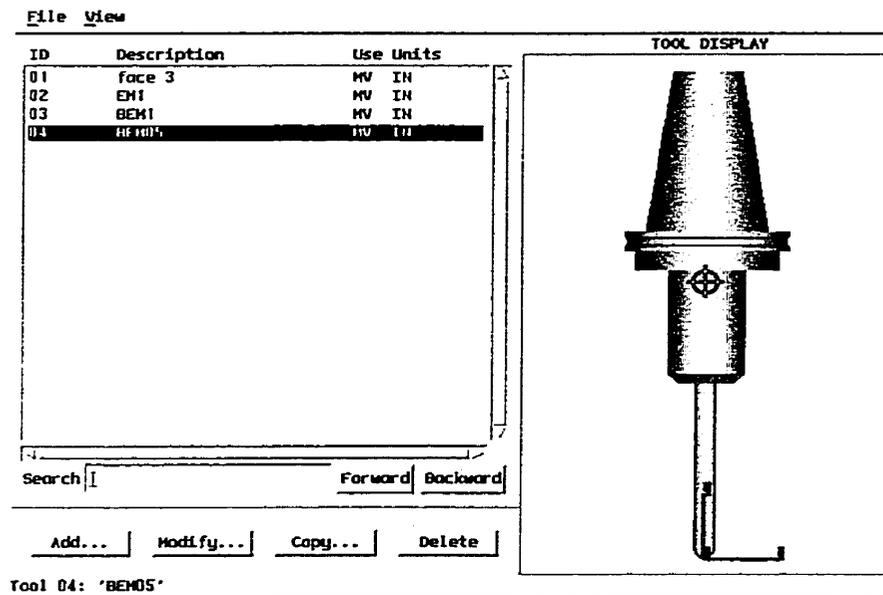


Figure 4.22 Définition des outils

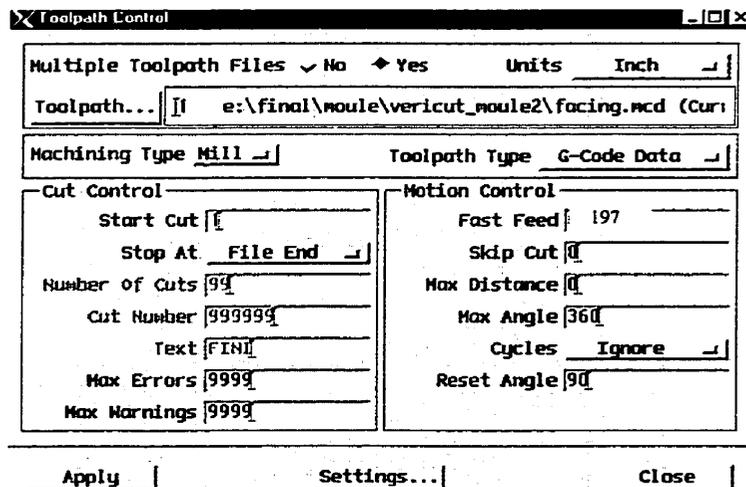


Figure 4.23 Fenêtre de chemin d'outil.

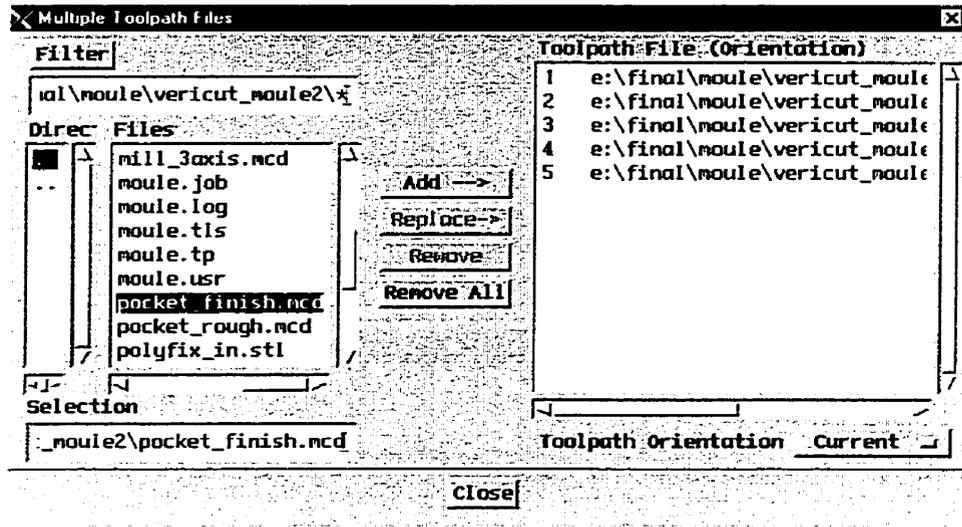


Figure 4.24 Fenêtre du settings avec "multiple toolpath files"

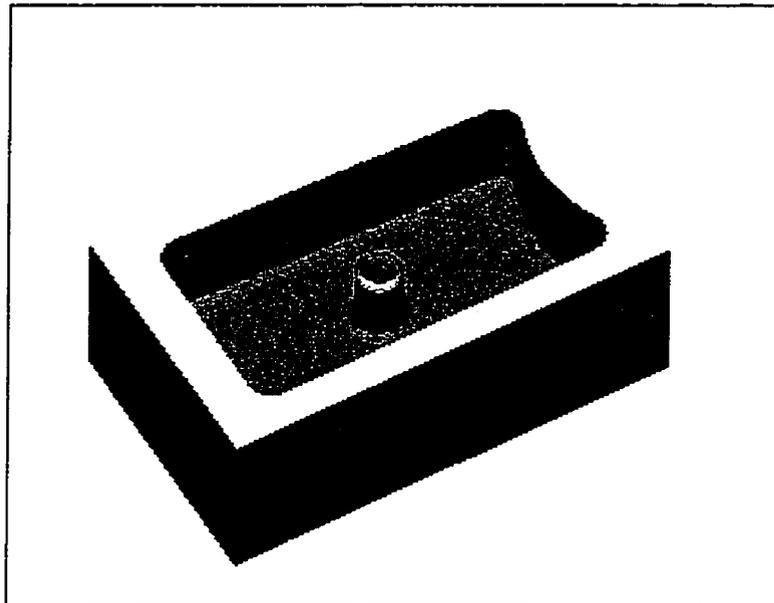


Figure 4.25 Le moule fini usiné avec le module de base de Vericut.

4.4 Essai 3 : Maquette d'automobile

L'essai 3 met en application un moule de petite voiture (figure 4.26). Suivant le même processus que les deux premiers essais, le modèle et le programme d'usinage sont issus de Pro/engineer. Les figures 4.27 et 4.28 représentent le graphique du chemin d'outil et les résultats de la simulation. Le déroulement du programme d'usinage avec VERICUT est visualisé à l'aide de la fenêtre "toolpath" contenue dans le menu "Info" (figure 4.29). La ligne du programme noircie représente la ligne de l'usinage en cours. Les résultats présentés à la figure 4.30 sont concluants.

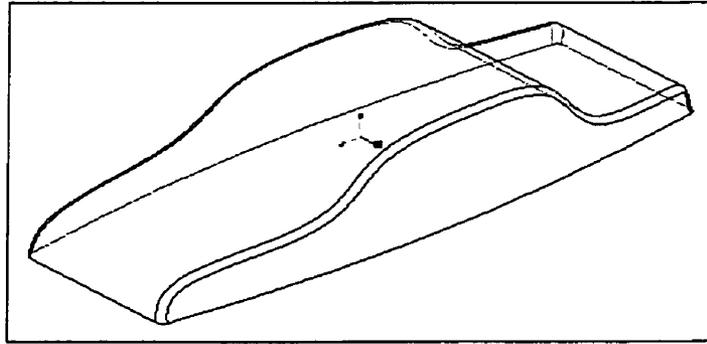


Figure 4.26 Le modèle "Auto"

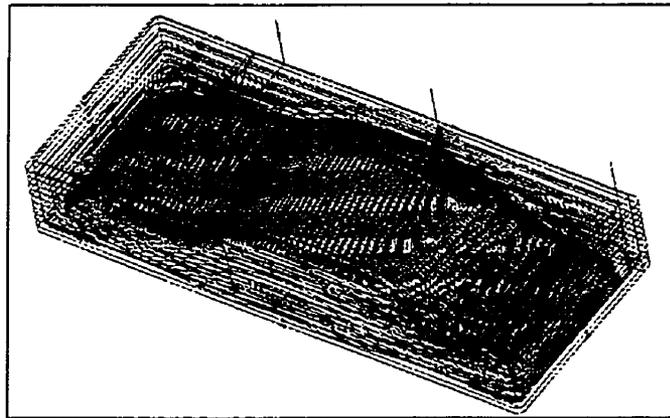


Figure 4.27 Le chemin d'outil généré avec Pro/Engineer

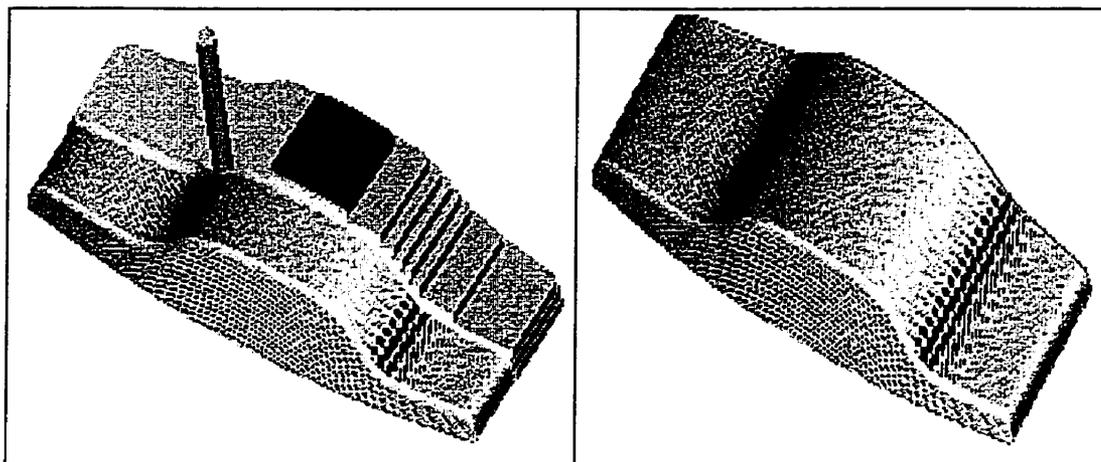


Figure 4.28 Les étapes de la simulation avec Pro/Engineer.

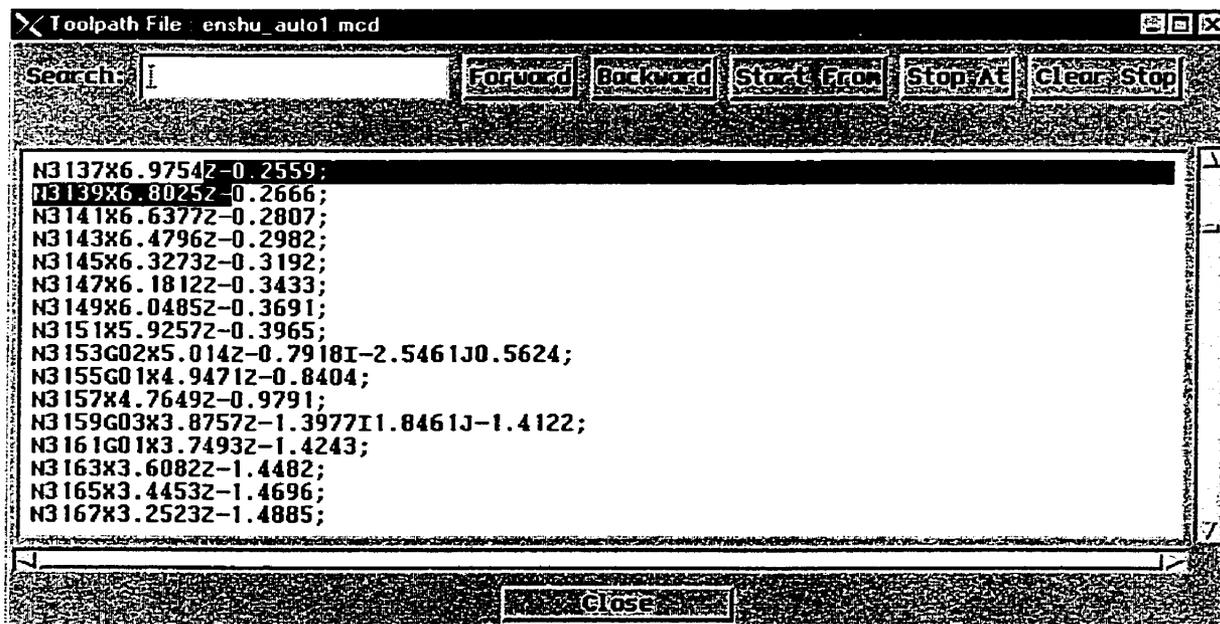


Figure 4.29 La fenêtre de déroulement du programme d'usinage avec VERICUT.

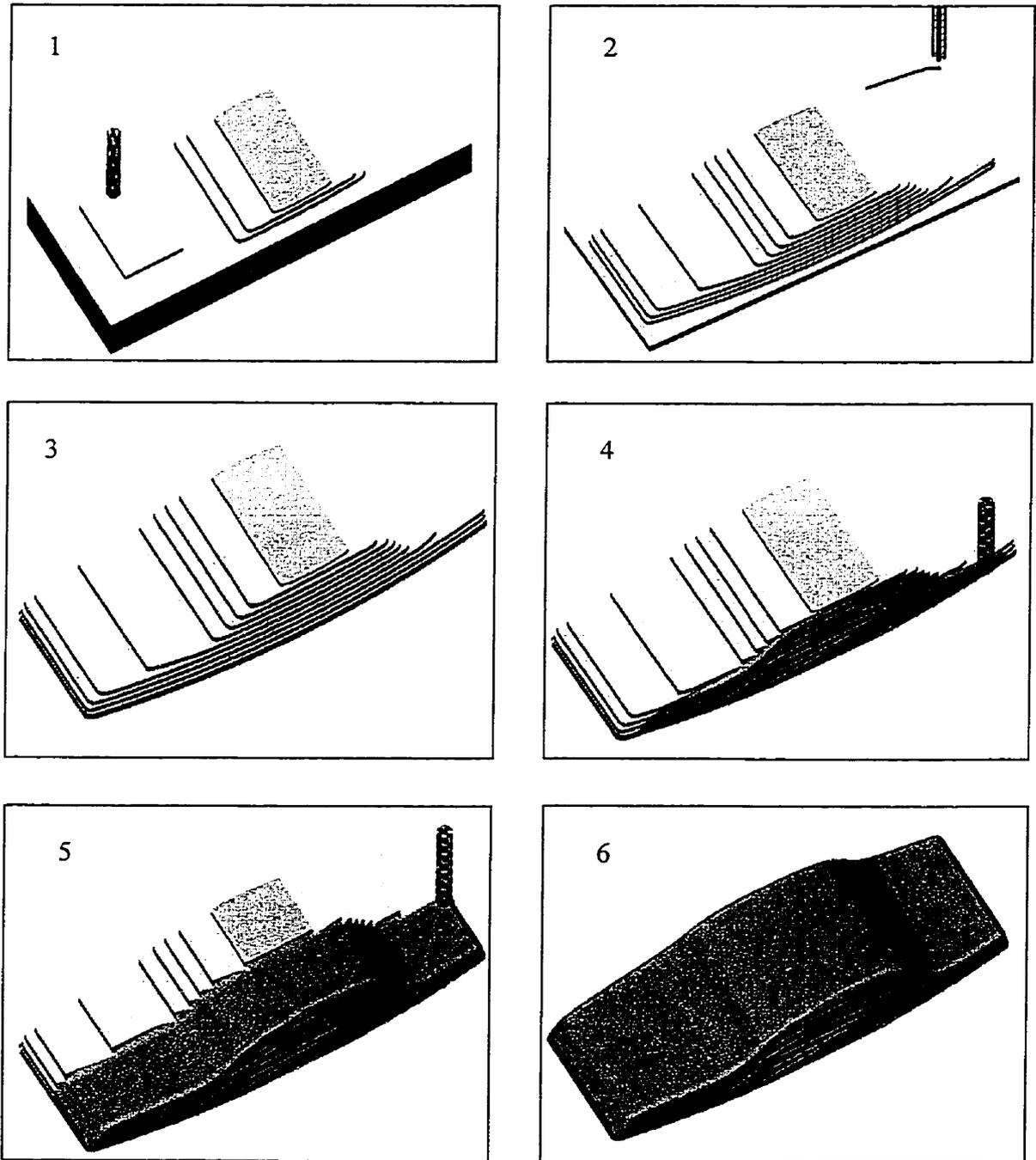


Figure 4.30 Étapes de la simulation de l'usinage du modèle "Auto" avec VERICUT

Vu la durée relativement longue du déroulement de son programme d'usinage, cet exemple est intéressant d'un point de vue optimisation. Le chemin d'outil est scindé en deux séquences (dégrossissage et finition). Les temps d'usinage originaux des deux opérations sont de 28.5 minutes et de 18.5 minutes respectivement. La figure 4.31 montre la fenêtre "Status" à la fin du cycle de la première séquence optimisée avec un temps de 13.83 minutes, et par conséquent une économie considérable épargnant plus de 51% du temps d'usinage.

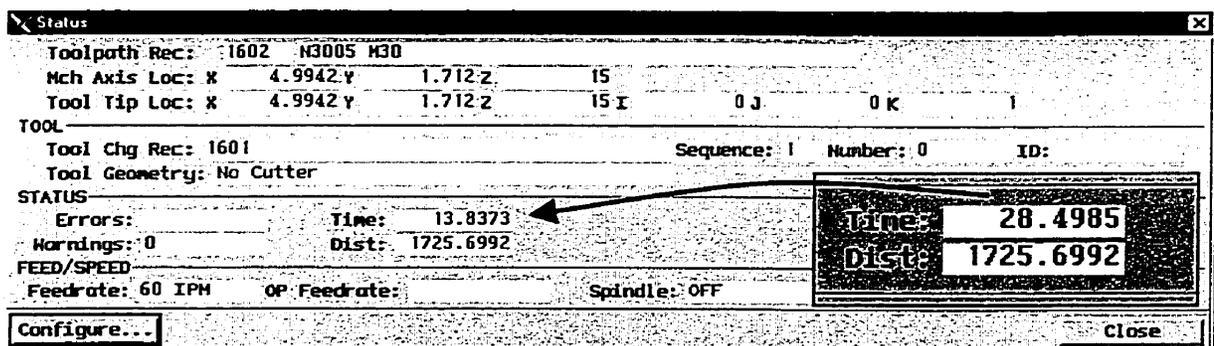


Figure 4.31 La séquence de dégrossissage optimisée

L'optimisation de la séquence de finition permet une économie de temps de 13%, pour un taux de réduction du temps d'usinage total de 36.5%.

Très importante et très utilisée en industrie, la base de données d'usinage IAM METCUT constitue la table de référence du module d'optimisation de VERICUT. La précision de certaines données empiriques telles que les effets des vibrations, les défauts de parcours relatifs aux déviations des outils liés à leurs fixations, ainsi que les données relatives à certains matériaux peuvent être inadéquats ou manquants. Par conséquent, l'optimisation du parcours d'outil ne représente qu'un bon indicateur approximatif des meilleures vitesses d'avance. Tel que mentionné dans l'essai 1, l'optimisation du chemin d'outil et l'économie du temps d'usinage réalisée est théorique.

4.5 Essai 4 : Plaque

Ce modèle réaliste représente une plaque servant de couvercle. Le programme est généré manuellement et contient des compensations de longueurs d'outils. Il englobe des séquences de chanfreinage, profilage et perçage. Initialement, pendant l'exécution du programme, des messages d'erreurs indiquaient la présence du code G54 non supporté par la machine. Aussi, quelques excès de matière sont observés sur la surface de la pièce, tels qu'illustrés à la figure 4.32. En corrigeant le chemin d'outil (figure 4.33), on peut voir le résultat de la simulation présenté à la figure 4.34 (d).

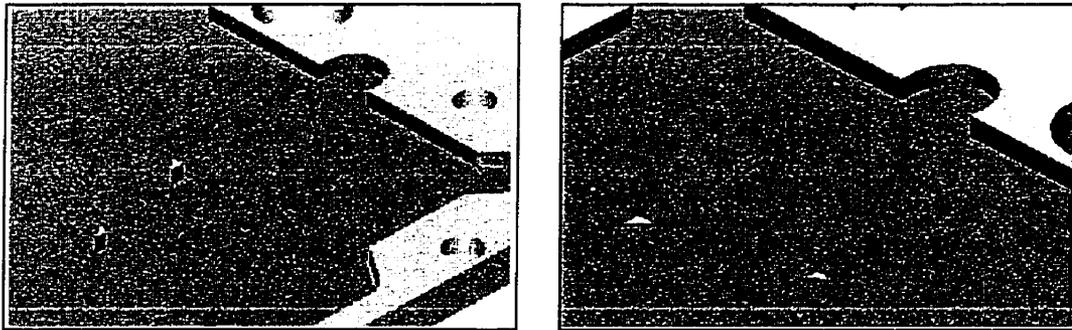


Figure 4.32 Les excès sur la surface de la plaque.

```

X-2.885; (AJOUT)
Y2.65;
X-1.86;
Y2.8;
X-3.15 ;
Y1.7428; (FIN DES AJOUTS)
.....
G02X-2.9Y-2.1331R3.6 ;
G01X-2.5942; (AJOUT)
Y-2.844;
X-2.8814;
Y-1.804; (FIN DES AJOUTS)

```

Figure 4.33 Correction du programme

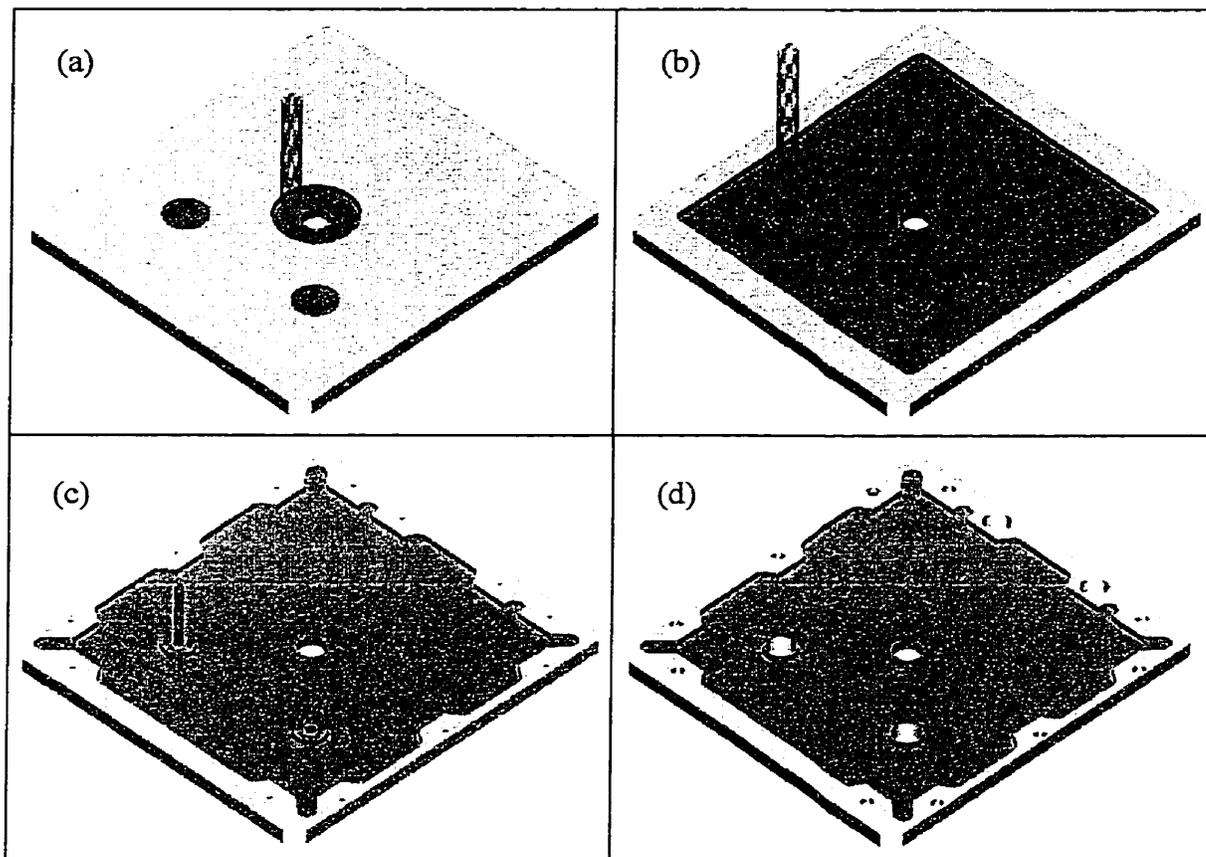


Figure 4.34 Les étapes de la simulation de la plaque.

Tel que mentionné, VERICUT permet la détection des hors courses et des collisions. La capacité de vérification des états de surfaces et la détection des erreurs émanant d'une programmation de chemin d'outil déficiente donnant une mauvaise qualité d'usinage est mise en évidence dans cet essai. Néanmoins, les erreurs incontournables des tolérances issues de la CAO (format STL très approximatif), les erreurs d'ordre pratique liées au choix de la hauteur de l'outil de coupe, les déviations et les vibrations non négligeables qui en résultent passent inaperçus.

4.6 Essai 5 : Poulie

Ce modèle utilisé dans le cadre du projet ‘‘Helios’’ a été usiné sur la machine ‘‘Enshu’’. Développé avec Pro/Engineer, il comporte plusieurs séquences d’usinage d’un côté de la poulie. Cet essai permet d’assurer le bon fonctionnement du simulateur et de démontrer sa robustesse dans des applications réelles.

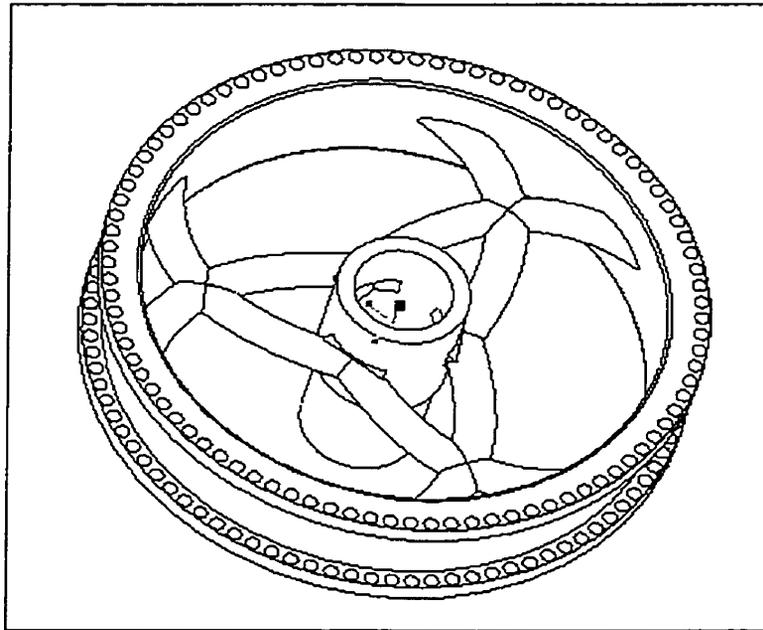


Figure 4.35 La poulie

Le résultat partiel obtenu avec cette simulation et illustré aux figures 4.36 et 4.37 montre des séquences d’usinage sans avaries, collisions ou dépassements. Le test de la partie disponible du programme (codes G) est satisfaisant.

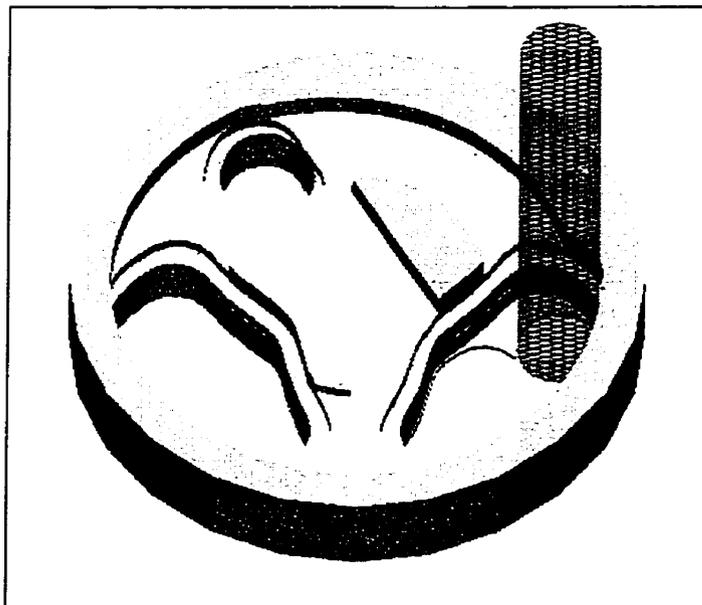


Figure 4.36 La poulie en cours de simulation avec Vericut

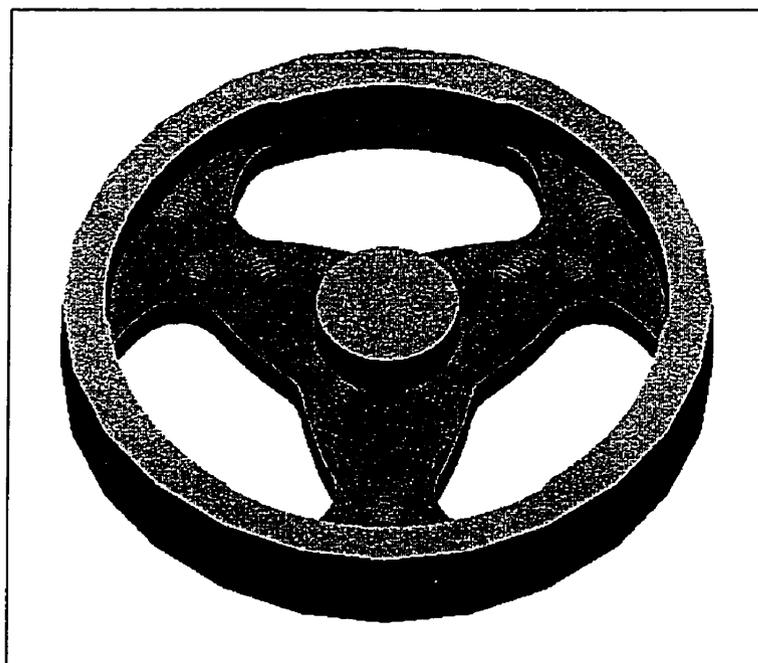


Figure 4.37 La poulie semi finie.

CONCLUSION

La technique de la simulation a révolutionné la vision des concepteurs et des manufacturiers. Elle permet d'anticiper la qualité du produit, le déroulement du processus de fabrication et par conséquent éviter toute situation coûteuse ou fâcheuse. La fraiseuse Enshu VMC-40 considérée dans ce projet est un exemple concret des prouesses possibles avec cette technologie.

Une fois définis, les éléments de la machine sont assemblés pour former la représentation virtuelle fidèle avec le respect de tous les aspects dimensionnels et fonctionnels de la fraiseuse.

Les différentes fonctions des menus du logiciel VERICUT, qui est à la base du simulateur, ont été largement élaborées et expliquées.

Enfin, quelques exemples d'application pour valider le bon fonctionnement du simulateur ont été présentés. Pour ce faire, des simulations de modèles simples, complexes, théoriques et réels ont été exécutées.

Les deux premiers essais ont comporté des simulations avec PRO/ENGINEER et VERICUT. On a pu noter la détection par ce dernier des avaries, tel que les dépassements et les collisions introduits dans les programmes. Le troisième essai a permis d'appliquer et de montrer quelques notions importantes telles que la définition des outils de coupe et l'optimisation des paramètres menant à un gain appréciable de productivité machine. Les derniers essais ont porté sur des modèles usinés avec la

fraiseuse dans le cadre de projets étudiants. Les résultats sont concluants et confirment le bon fonctionnement du simulateur.

Ce travail démontre l'utilité du simulateur dans la détection des programmes et des montages non conformes et peut servir les étudiants dans l'apprentissage des rudiments de l'usinage sans mettre en péril les installations ni même leurs propres personnes.

Il se veut aussi un document de référence pour des applications futures et une contribution dans l'effort de l'introduction des nouvelles techniques au niveau de l'ÉTS.

RECOMMANDATIONS

Le logiciel VERICUT constitue un outil de base robuste et très performant. L'aspect "simulation" est bien développé en offrant des fonctions et des fichiers aidant à la construction des machines-outils et à leurs simulations. Le côté modélisation est rudimentaire, en comparaison avec PRO/ENGINEER, alors que les assemblages des éléments, la mise en place et le montage du brut constituent un effort en soi. En améliorant l'aspect "modélisation- assemblage" et l'ajout de la possibilité de génération des programmes (codes G), VERICUT constituerait un outil encore plus performant.

L'utilisation du simulateur est avantageuse pour la détection des avaries importantes telles que les collisions entre les composantes de la machine. Par contre, la qualité des résultats de l'optimisation et de la comparaison est limitée. Cette dernière est sujette généralement à la problématique de la précision de la CAO aujourd'hui et spécifiquement à certains formats de modélisation utilisés par VERICUT. Le format STL employé est très approximatif et peut aboutir à des résultats de détection d'erreur des modèles simulés erronés. L'optimisation dépend de la qualité de la base de données IAMS METCUT. Par conséquent, une marge d'erreur doit être prise en considération pendant les processus de vérification.

Enfin, le manque d'un outil contribuant à la sécurité, ajouté à l'expertise variable des étudiants dans le domaine de l'usinage, à leur grand taux de roulement au fil des sessions, ainsi qu'aux séances de laboratoires courtes et intensives font que l'apprentissage est laborieux. Pour toutes ces raisons et afin de réduire les coûts de fonctionnement des MOCN, des rebuts et de l'usure des outils au niveau des laboratoires d'usinage de l'ÉTS, la généralisation de l'utilisation du simulateur est recommandée.

BIBLIOGRAPHIE

- [1] Roger Smith, *Encyclopedia of Computer Science*, 4th Ed, November 99, Macmillan References Press, New York..
- [2] Alkan Donmez, *Development of virtual machine tools*, 1996, NIST, Gaithersburg.
- [3] *FANUC OM-MODEL A, Manuel de l'utilisateur*, 1985, Fanuc LTD.
- [4] Manuel d'aide de VERICUT.
- [5] Paul Gely, *Commande numérique des machines outils*, GPA-421, 2ème Ed, Janvier 1990, École de technologie supérieure, Montréal.
- [6] <http://www.cgtech.com>

ANNEXE A

Format des registres associés au contrôleur de la machine

Tableau 1

Définition de tous les mots et caractères dans le fichier de commande du simulateur.

Name	Type	Sub-type	Inch Format	Metric Format	Multiplier
#	SPECIAL	VARIABLE TAG			
\$\$	SPECIAL	SKIP			
%	SPECIAL	SKIP			
(SPECIAL	BEGIN COMMENT			
)	SPECIAL	END COMMENT			
*	MATH	MULTIPLY			
+	MATH	ADD			
,	SPECIAL	SEPARATOR			
-	MATH	SUBTRACT			
/	CONDITION	AtanCondDivide	DECIMAL	DECIMAL	
:	MACRO		DECIMAL:4.0	DECIMAL:4.0	
;	SPECIAL	END OF BLOCK			
=	MATH	ASSIGNMENT			
[MATH	LEFT PRECEDENCE			
]	MATH	RIGHT PRECEDENCE			
^	MATH	POWER			
A	MACRO		DECIMAL:5.3	DECIMAL:5.3	
ABS	FUNCTION	abs			
AND	LOGICAL	AND			
ATAN	FUNCTION	atan2_d			
C	MACRO		DECIMAL:5.3	DECIMAL:5.3	
COS	FUNCTION	cos_d			
D	MACRO		DECIMAL:2.0	DECIMAL:0	
DO	MACRO		DECIMAL	DECIMAL	
END	MACRO		DECIMAL	DECIMAL	
EQ	LOGICAL	EQ			
F	MACRO		DECIMAL:3.4	DECIMAL:4.3	
FIX	FUNCTION	floor			
FUP	FUNCTION	ceil			
G	MACRO		DECIMAL:2.0	DECIMAL:2.0	
GE	LOGICAL	GE			
GOTO	MACRO		DECIMAL	DECIMAL	
GT	LOGICAL	GT			
H	MACRO		DECIMAL:2.0	DECIMAL:2.0	
I	MACRO		DECIMAL:3.4	DECIMAL:4.3	1
IF	MACRO		DECIMAL	DECIMAL	
J	MACRO		DECIMAL:3.4	DECIMAL:4.3	1
K	MACRO		DECIMAL:3.4	DECIMAL:4.3	
L	MACRO		DECIMAL:4.0	DECIMAL:4.0	
LE	LOGICAL	LE			
LT	LOGICAL	LT			
M	MACRO		DECIMAL:2.0	DECIMAL:0	
N	MACRO		DECIMAL:4.0	DECIMAL:4.0	
NE	LOGICAL	NE			
O	MACRO		DECIMAL:4.0	DECIMAL:4.0	
OR	LOGICAL	OR			
P	MACRO		DECIMAL:4.0	DECIMAL:4.0	
Q	MACRO		DECIMAL:3.4	DECIMAL:4.3	
R	MACRO		DECIMAL:3.4	DECIMAL:4.3	
RND	FUNCTION	Round			
ROUND	FUNCTION	Round			
S	MACRO		DECIMAL:4.0	DECIMAL:0	
SIN	FUNCTION	sin_d			
SQRT	FUNCTION	sqrt			
T	MACRO		DECIMAL:2.0	DECIMAL:0	
TAN	FUNCTION	tan_d			
THEN	SPECIAL	IGNORE			
U	MACRO		DECIMAL	DECIMAL	
V	MACRO		DECIMAL	DECIMAL	
W	MACRO		DECIMAL	DECIMAL	
WHILE	MACRO		DECIMAL	DECIMAL	
X	MACRO		DECIMAL:3.4	DECIMAL:4.3	1
Y	MACRO		DECIMAL:3.4	DECIMAL:4.3	1
Z	MACRO		DECIMAL:3.4	DECIMAL:4.3	1

ANNEXE B

Tableaux des codes et autres commandes

Tableau 2
Les codes G des états de la machine

Codes	Signification
G00	Positionnement à avance rapide sur un, deux ou trois axes simultanés. Les avances sont de 18m/min sur X et Y et 12 m/min sur l'axe Z.
G01	Interpolation linéaire à avance programmée.
G02	Interpolation circulaire dans le sens CW. (Préalable : G17, G18, G19).
G03	Interpolation circulaire (CCW).
G17	Déclaration du mode interpolation circulaire en XY.
G18	Déclaration du mode interpolation circulaire en ZX.
G19	Déclaration du mode interpolation circulaire en YZ
G20	Programmation en pouces.
G21	Programmation en mm.
G90	Déclaration du mode absolu de programmation.
G91	Déclaration du mode incrémental de programmation.

Tableau 3
Les codes G des cycles de la machine

Codes	Signification
G73	Cycle perçage par étapes à percussions simples, afin de briser le copeau.
G74	Cycle de taraudage à gauche.
G76	Cycle d'alésage fin
G80	Annulation de tout cycle de perçage (G73, G74, G76, G81, jusqu'à G89).
G81 à G89	Différents cycles de perçage.
G98	Mode déclarant la fin des cycles de perçage au point initial.
G99	Mode déclarant la fin des cycles de perçage au niveau R.

Tableau 4

Liste des codes préparatoires de la machine.

Codes	Signification
G04	Arrêt temporisé ou Dwell.
G10	Établissement de la valeur de correction.
G28	Retour automatique au point de changement d'outil.
G40	Annulation de la compensation sur le rayon.
G41	Déclaration de la compensation à gauche.
G42	Déclaration de la compensation à droite.
G43	Compensation de la longueur du nouvel outil installé.
G44	Autre mode de Compensation de la longueur d'outil.
G49	Annulation de la Compensation de la longueur d'outil.
G65	Appel d'une fonction.
G92	Déclaration d'un système d'axes par rapport à la position de l'outil.
G94	Déclaration des avances en unité/minute.
G95	Déclaration des avances en unité/tour d'outil.

Tableau 5

Commandes variées inscrites dans le registre de la machine.

Codes	Signification
X,Y,Z,A	Quatre mots précédents les valeurs numériques des déplacements suivant ces axes.
I,J,K	Valeurs des coordonnées du centre d'arc, chanfreinage
U,V,W	Co-linéaires.
D	Compensation du rayon de l'outil.
F	Avance programmée.
H	Registre des longueurs des outils.
L	Loop count.
P	Durée de l'arrêt temporisé (dwell).
Q	Valeur de retrait latéral, valeur incrémentale des descentes de coupe (perçage).
R	Rayon de l'arc.
S	Vitesse de rotation de la broche en tour/min. (6000tr/min maximum)
T	Sélection de l'outil dans le magasin en vue d'un futur changement d'outil.

Tableau 6

Les codes M de la machine

Codes	Signification
M00	Stop, arrêt obligatoire du programme.
M01	Stop optionnel.
M03	Départ de la broche dans le sens horaire (CW).
M04	Départ de la broche dans le sens inverse horaire (CCW).
M05	Arrêt de la broche.
M06	Changement d'outil automatique.
M08	Mise en marche de la pompe du lubrifiant liquide.
M09	Arrêt du lubrifiant
M19	Arrêt de la broche, toujours à la même position angulaire pour permettre le changement d'outil (équivalent à M05 avec indexation).
M30	Fin du programme.
M98	Sous-routines, activation.
M99	Retour au programme original.

ANNEXE C

Tableau et exemples des codes optimisables.

Tableau 7

Les codes optimisables et non optimisables

Les codes optimisables	Les codes non optimisables
<ul style="list-style-type: none"> - N, - G01, interpolation linéaire - G02, interpolation circulaire CW - G03, interpolation circulaire CCW - F, - X Y Z A B C (en mode G01), - X Y I J (en mode G02 ou G03 et en plan XY), - X Z I K (en mode G02 ou G03 et en plan ZX), - Y Z J K (en mode G02 ou G03 et en plan YZ), - BSPLINE P SD K PL R PW X Y Z (en mode NURBS). 	<ul style="list-style-type: none"> - G00, positionnement, avance rapide. - M03, broche CW - Mouvements programmés dans des états autre que : G01, G02, G03 ou NURBS. - Mouvements des cycles canés (G81-89) - Mouvements programmés en mode vitesse de surface constante (CSS) - Mouvements programmés en mode feed/revolution (IPR/MMPR) - Mouvements programmés du cercle en mode (inverse time, IVT). - Mouvements programmés en utilisant la compensation du diamètre de l'outil (CDC), G41-42

Exemples de codes optimisables et non optimisables :

G01 X1.0 Y2.0 F20.0 ⇒ optimisable

X18.0 Y.2 A180 ⇒ optimisable

G01 X12.0 Z1.5 M03 ⇒ non optimisable : M03

G00 Y.5 Z3.0 ⇒ non optimisable : mode G00

G41 X4.0 Y1.2 F30 D01 ⇒ non optimisable : G41/D01

ANNEXE D

Listes des fichiers de conception et des essais de validation

Tableau 8

Liste des éléments de la machine et des essais avec PRO/ENGINEER

Éléments de la machine modélisés avec PRO/ENGINEER		Base.prt Base.drw Colonne.prt Colonne.drw Porte_broche.prt Porte_outil.prt Table.prt Table.drw Polyfix.prt Barre.prt Assemblage.asm
Liste des essais avec PRO/MFG	Attache	<ul style="list-style-type: none"> • Attache.mfg - Degrossissage.ncl, Profil.ncl, Perçage.ncl - Fraisage.ncl (Il englobe les séquences de dégrossissage, profilage et perçage) - Attache.tph - Attache.prt (Le modèle) - Wp_attache.prt (Le brut) - Attache.asm (L'assemblage du brut et du modèle).
	Moule	<ul style="list-style-type: none"> • Cavité.mfg - Cavité.asm - Cavite.tph - Mill_3axes.ncl (facing, pocket_rough, profil) - Cavity.prt - Cavity_wp.prt
	Auto	<ul style="list-style-type: none"> • Auto.mfg - Auto.asm - Auto.tph - Auto.ncl (Dégrossissage, finition) - Auto.prt - Auto_wp.prt
	Poulie	<ul style="list-style-type: none"> • Poulie.mfg - Poulie.asm - Poulie.tph - Poulie.ncl (Dégrossissage, Finition) - Poulie.prt - Stock.prt

Tableau 9

Liste des fichiers de simulation accomplis avec VERICUT

Essais	Fichiers Principaux	Fichiers secondaires
Attache	Attache.job	Fichier d'utilisation sur le module « machine simulation » de Vericut.
	Attache.usr	Fichier d'utilisation sur le module de base de Vericut.
	Attache.mch	- Base.stl, Colonne.stl, table.stl, porte-broche.stl, spindle.stl - La fixation - polyfix_in.stl, - barre.stl). - wp_attache.stl.
	Attache.tls	3 types d'outils.
	Enshu.ctl	
	Fraisage.mcd	Programme (codes G) de type « tool type ». (avec fichiers de sortie : faisage.tp et fraisage.log).
	Attache.tpo	Chemin d'outil optimisé.
	Attache.lbo	Fichier des caractéristiques d'optimisation du chemin d'outil.
	Attache.rpt	Fichier détaillée des excès et manque de matière.
Moule	Moule.job	Fichier d'utilisation sur le module « machine simulation » de Vericut.
	Moule.usr	Fichier d'utilisation sur le module de base de Vericut.
	Moule.mch	- Base.stl, Colonne.stl, table.stl, porte-broche.stl, spindle.stl - La fixation (polyfix_in.stl, barre.stl). - Brut moule.stl
	Moule.tls	4 types d'outils.
	Enshu.ctl	
	*.mcd	Programme (codes G) de type « tool type » élaboré en 5 séquences: - Facing.mcd - Profile.mcd - Pocket_rough.mcd - Pocket_finish.mcd - Seq005.mcd
Maquette "Auto"	Auto.job	Fichier d'utilisation sur le module « machine simulation » de Vericut.
	Auto.usr	Fichier d'utilisation sur le module de base de Vericut.
	Auto.mch	- Base.stl, Colonne.stl, table.stl, porte-broche.stl, spindle.stl - La fixation (polyfix_in.stl, barre.stl). - Brut auto.stl.
	Auto.tls	2 types d'outils .
	Enshu.ctl	
	*.mcd	Programme (codes G) de type « tool type » élaboré en 2 séquences: - Volume milling.mcd - Finish_convntl.mcd
	Auto.lbo	
	Auto1.tpo	
Auto2.tpo		

Tableau 9 (suite)

Essais	Fichiers Principaux	Fichiers secondaires
Plaque	Plaque.job	Fichier d'utilisation sur le module « machine simulation » de Vericut.
	Plaque.usr	Fichier d'utilisation sur le module de base de Vericut.
	Plaque.mch	- Base.stl, Colonne.stl, table.stl, porte-broche.stl, spindle.stl - La fixation (polyfix_in.stl, barre.stl). - Brut Plaque.stl
	Plaque.tls	6 types d'outils.
	Enshu.ctl	
	Plaque.mcd	- Programme (codes G) de type « Tool length compensation ».
Poulie	Poulie.job	Fichier d'utilisation sur le module « machine simulation » de Vericut.
	Poulie.usr	Fichier d'utilisation sur le module de base de Vericut.
	Poulie.mch	- Base.stl, Colonne.stl, table.stl, porte-broche.stl, spindle.stl - La fixation (polyfix_in.stl, barre.stl). - Brut Plaque.stl
	Poulie.tls	4 types d'outils.
	Enshu.ctl	
	Poulie.mcd	- Programme (codes G) de type « Tool type ».