



PROGRAMMATION AVANCEE DES MACHINES OUTILS :IMPLEMENTATION DE L'USINAGE TROCHOIDAL ET DU TREFLAGE DANS LA CHAINE NUMERIQUE STEP-NC

Raphael Laguionie

► **To cite this version:**

Raphael Laguionie. PROGRAMMATION AVANCEE DES MACHINES OUTILS :IMPLEMENTATION DE L'USINAGE TROCHOIDAL ET DU TREFLAGE DANS LA CHAINE NUMERIQUE STEP-NC. Génie des procédés. Ecole Centrale de Nantes (ECN) (ECN) (ECN) (ECN); Université de Nantes, 2007. Français. <tel-00366065>

HAL Id: tel-00366065

<https://tel.archives-ouvertes.fr/tel-00366065>

Submitted on 5 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**MASTER SCIENCES MECANQUES APPLIQUEES
SPECIALITE GENIE MECANIQUE**

Année 2006/2007

Thèse de Master SMA

*Diplôme cohabilité par
L'École Centrale de Nantes et l'Université de Nantes*

TITRE :

**PROGRAMMATION AVANCEE DES MACHINES OUTILS :
IMPLEMENTATION DE L'USINAGE TROCHOIDAL ET DU TREPLAGE
DANS LA CHAINE NUMERIQUE STEP_NC**

Présentée et soutenue par :

RAPHAËL LAGUIONIE

le 01/10/2007

à l'école Centrale de Nantes

Directeur de Thèse : Jean-Yves HASCOËT

Laboratoire : IRCCyN

Résumé :

L'augmentation de la productivité sur machine à commande numérique passe, entre autre, par l'amélioration de la programmation de celles-ci. Le langage sur lequel se base actuellement cette programmation date du début des années 80 avec la norme 6983 qui définit les principes du code G. Ce dernier présente certains inconvénients aux vues des nouvelles stratégies d'usinage et crée une rupture de la chaîne numérique au niveau de la fabrication. Un nouveau format émergent, le Step_NC, voit le jour actuellement dans le but de palier aux manques du code G et d'intégrer complètement la chaîne CAO-FAO-CN.

Nos travaux ont pour objectif d'implanter deux nouvelles stratégies d'usinage, le tréflage et l'usinage trochoïdal, au sein du format Step_NC. Pour ceci, nous nous basons sur une description des trajectoires à l'aide de courbes à motif. La construction de ces courbes se fait par répétition d'un motif le long d'une courbe guide. Pour mettre en avant la faisabilité de l'implantation de ces stratégies à motifs dans le format Step_NC, nous avons développé une chaîne numérique complète permettant, à partir du modèle CAO, de créer le fichier Step_NC, de générer les trajectoires et d'usiner la pièce sur machine à commande numérique. Une partie de l'intégration de l'usinage trochoïdal et du tréflage dans le format Step_NC a été réalisé par implémentation dans le logiciel PosSFP, en collaboration avec le laboratoire de l'université de Postech en Corée. Plusieurs pièces tests ont ensuite été usinées pour valider l'ensemble de la chaîne numérique Step_NC complète dans le cas de l'usinage trochoïdal.

Mots clefs : Step_NC, programmation, machines outils, code G, chaîne numérique, tréflage, usinage trochoïdal, stratégies à motif, courbe guide, ISO 14649, AP 238, PosSFP.

Abstract :

Productivity on machine tools can be increased by improvement of programming them. Nowadays, the language used for programming is the G-code which dates from the beginning of the eighties with the norm ISO 6983. With the new technologies and possibilities offered, the G-code seems to be exceeded and presents drawbacks that create a rupture in the numerical chain at the manufacturing step. A new standard, the Step_NC, is developed with the aim of landing in the lacks of G-code and to totally integer the numerical chain CAD-CAM-CNC.

The objectives of our works here are to implement two new milling strategies in the Step_NC standard: trochoïdal and plunge milling. For that, we proposed to build toolpaths by using a pattern that is repeated all along a guide curve. To show the feasibility of the implementation of milling patterned strategies in Step_NC, we built a complete numerical chain that is able, from a CAD model, to create a Step_NC file, to generate toolpaths and to manufacture parts on the machine tool. To generate the Step_NC file, we implemented trochoïdal and plunge milling in PosSFP, the Korean software for shop floor programming. To validate all the tools we built in the Step_NC numerical chain, we manufactured tests parts in trochoïdal milling.

Key words : Step_NC, programation, machine tool, G-code, numerical chain, plunge milling, trochoïdal milling, patterned strategies, guide curve, ISO 14649, AP 238, PosSFP.

Remerciements

Le bon déroulement de ces travaux de Master n'a pu être possible que grâce à l'accueil qui m'a été fait au sein de l'IRCCyN, et en particulier par mon encadrant de stage, Jean-Yves HASCOET. Ses conseils et orientations précises m'ont été d'une aide précieuse. Je tiens donc à le remercier tout particulièrement.

Je tiens aussi à remercier les membres du jury pour leurs appréciations et retours sur le travail que j'ai effectué tant sur le fond que sur la forme.

Un grand merci également à Matthieu RAUCH avec qui j'ai été amené à travailler dans le cadre de l'étude proposée. Toujours là pour répondre à mes questions, il a été très présent au jour le jour pour que nous puissions échanger nos idées sur le sujet et a largement contribué à ma bonne intégration dans le laboratoire.

Enfin, je remercie l'ensemble de l'équipe MO2P pour son accueil et sa bonne ambiance.

Table des matières

Introduction: situation de l'étude.....	7
1. Programmation des MOCN.....	10
1.1. La programmation actuelle des machines outils	10
1.1.1. La norme Iso 6983.....	10
1.1.2. Les post-processeurs	11
1.1.3. Les lacunes du code G.....	11
1.2. Evolution de la programmation des MO	12
1.2.1. Du coté des constructeurs.....	12
1.2.2. Nouvelle vision de la maquette numérique	13
1.2.3. Le Step compliant NC	14
1.2.4. Structure d'un programme Step_NC	15
1.2.5. Les apports du Step_NC.....	17
1.2.6. Définitions et normes sur le Step_NC.....	18
1.3. Les groupes de travail sur le Step_NC :	19
1.3.1. IMS (Intelligent Manufacturing System Projet) Step_NC projet.....	19
1.3.2. Projet Européen Esprit Step_NC.....	20
1.3.3. Projet « Super Model » aux USA.....	20
1.3.4. Le projet SMS : Step Manufacturing Suite	21
1.3.5. Autres projets concernant le Step_NC	21
1.4. Les travaux de recherche effectués sur le Step_NC en fraisage.....	21
1.4.1. Travaux de recherche en Allemagne	21
1.4.2. Travaux de recherche au Royaume-Uni	24
1.4.3. Travaux de recherche aux USA	26
1.4.4. Travaux de recherche en Nouvelle Zélande	28
1.4.5. Travaux de recherche en Corée	28
1.4.6. Travaux de recherche en France	32
1.5. Conclusion.....	33
2. Optimisation des trajectoires en fraisage	34
2.1. L'UGV : une technologie en perpétuel essor.....	34
2.1.1. L'UGV : définitions et frontières	34
2.1.2. Avantages et inconvénients de l'UGV	35
2.2. Optimisation du processus d'usinage.....	37
2.2.1. Généralités.....	37
2.2.2. Le simulateur Ircsyn.....	38
2.2.3. Une nouvelle approche utilisant les données CN.....	39
2.3. Trajectoires en tréflage et usinage trochoïdal	41
2.4. Conclusion.....	43

3.	Partie stage de recherche	44
3.1.	Les normes AP238 et 14649	45
3.2.	La chaine numérique coréenne : PosSFP – TPG – TPV	48
3.2.1.	PosMMI (Men Machine Interface)	49
3.2.2.	PosSFP (Shop-floor Programing).....	50
3.2.3.	PosTPG (Tool Path Generator) / PosTPV (Tool Path Viewer).....	52
3.2.4.	PosCNC.....	53
4.	Implémentation de l’usinage trochoïdal et du tréflage dans la chaine numérique Step_NC	54
5.	Etape 1 : Génération du fichier Step_NC.....	56
5.1.	Structure d'un programme Step_NC et de PosSFP	56
5.2.	Sélection des paramètres nécessaires à la définition d'une trochoïde et du tréflage	58
5.3.	Implantation des stratégies dans le fichier Step_NC.....	60
5.4.	Mise en place des stratégies à motifs dans PosSFP	60
5.5.	Possibilités d'amélioration.....	62
5.6.	Conclusion de la première étape	63
6.	Etape 2 : Du fichier Step_NC à l’usinage	64
6.1.	Lecture du fichier Step_NC.....	65
6.2.	Génération des trajectoires pour l’usinage trochoïdal.....	66
6.3.	Conclusion et bilan de la génération des trajectoires	71
7.	Etape 3 : Mise en œuvre de la chaîne numérique Step_NC et validation par l’usinage de pièces.....	73
7.1.	Construction des pièces test et traitement dans la chaine numérique Step_NC.....	73
7.2.	Usinage sur les moyens à disposition : nécessité d’un interpréteur	74
7.2.1.	Machine parallèle Verne et Siemens 840D	74
7.2.2.	L’interpréteur	75
7.2.3.	Usinage par interpolation de type Spline	76
7.3.	Validation expérimentale	Erreur ! Signet non défini.
7.4.	Conclusion.....	79
8.	Conclusion, intérêt des travaux réalisés et perspectives	80
8.1.	Les travaux réalisés	80
8.2.	Intérêts de ces travaux dans la recherche sur le format Step_NC	80
8.3.	Les perspectives	81
	Références.....	83

Programmation avancée des machines outil : application aux machines parallèles

La recherche actuelle a pour objectif d'améliorer la production sur machine outil (M.O.) au travers du développement de nouveaux formats simplifiant le flux des données tout au long de la chaîne numérique jusqu'à la fabrication, de la génération de trajectoires d'usinage optimisées, de l'intégration en temps réel des paramètres influençant la coupe et la trajectoire, de l'utilisation de machines aux structures nouvelles... L'usinage grande vitesse permet aujourd'hui d'accroître la productivité au travers des gains de temps et de qualité qui ne sont plus à démontrer. Ce mode d'usinage récent met en jeu des phénomènes de coupe et dynamiques particuliers qui le différencient de l'usinage dit « conventionnel ». Ces phénomènes ont une influence sur la trajectoire de l'outil et beaucoup d'efforts de recherche ont lieu dans ce domaine pour pouvoir adapter au mieux ces trajectoires. De même, l'amélioration des technologies machines, tant au niveau de la multiplication des axes et des possibilités que de la rigidité et de la dynamique permettent d'exploiter au mieux de nouveaux types d'usinage comme l'usinage trochoïdal ou le tréflage par exemple. De nouvelles structures machine ont vu le jour au travers des machines à structure parallèle. Leurs bonnes performances dynamiques ainsi que leurs inerties faibles ouvrent un nouveau créneau à l'usinage par enlèvement de matière que la recherche se doit de mettre en avant. Pour finir, la programmation actuelle des machines outils à commande numérique s'effectue actuellement, et ce depuis le début des années 80, à l'aide du code G. Cette programmation, dont on voit réellement aujourd'hui les limites face aux évolutions citées précédemment, s'avère désuète et ne répond plus aux attentes de la technologie. De nouveaux travaux de recherche sont menés depuis quelques années pour faire évoluer ce langage de programmation vers un langage universel qui se doit de pallier aux manques du code G. C'est ainsi que le Step_NC a été développé et demande aujourd'hui à être mis en pratique et amélioré. Ce changement de vision dans la programmation des MOCN pourra alors permettre une meilleure intégration de la fabrication dans la chaîne numérique.

Nous allons donc nous intéresser à détailler les travaux qui ont été menés dans ce domaine.

Dans le cas d'une pièce destinée à être usinée sur machine à commande numérique et dans le cadre de notre étude, nous pourrions simplifier notre vision de la chaîne numérique à un schéma plus classique, réduit et pragmatique :

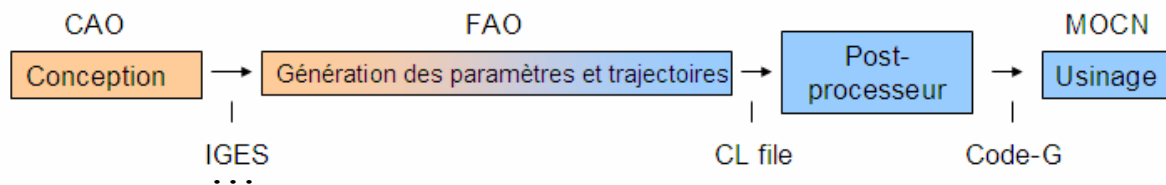


Fig. 2 : La chaîne numérique simplifiée

Le flux de données s'effectue de la conception jusqu'à l'usinage. La maquette CAO est envoyée en FAO sous la forme d'un format standard. Celui-ci peut être IGES, comme sur la Fig. 2, ou Step (Standard of Exchange of product) qui est basé sur une représentation B-rep, ou autre. La pièce est ensuite traitée par un logiciel de FAO dont le but est de préparer l'usinage par la génération de trajectoires, d'outils, de paramètres de coupe. Une fois cette phase terminée, le fichier CL (Cutter location) ainsi obtenu doit être post traité pour pouvoir être lu par la commande numérique. Cette dernière ne lit que le code G qui comporte souvent des fonctions spécifiques à chaque machine.

Ainsi, la problématique générale de l'usinage d'une pièce pourrait être :

'Comment, à partir d'un modèle CAO, obtenir une pièce qui soit : - la plus fidèle possible, - usinable par toutes les machines adaptées (adaptabilité), - pour un coût (i.e. temps d'usinage, de mise au point, ...) minimum'

L'objet de cette étude concernera plus spécifiquement le dernier 'maillon' de la chaîne numérique simplifiée, au niveau de la Fao puis au travers du contrôle de la génération des surfaces sur la machine outil.

Nous nous intéresserons en particulier dans ce mémoire aux 2 aspects suivants :

- les formats d'échange entre la CAO et la FAO, l'utilisation de Post-processeurs, l'emploi du code de programmation G sont autant d'étapes qui introduisent des approximations et erreurs se répercutant directement sur la pièce finie. De plus, l'impossibilité d'un retour expérience de la CN en direction de la FAO et de la CAO pose de gros problèmes et empêche en partie le bouclage de la chaîne numérique. Nous nous intéresserons ainsi au principe d'un nouveau format, le Step_NC, dont l'objectif est la meilleure intégration de la boucle CAO-FAO-CN.
- Dans le cadre du développement du format Step_NC, nous nous intéresserons à la programmation de trajectoires en UGV, et spécialement dans le cas de l'usinage trochoïdal et en tréflage

1. Programmation des MOCN

1.1. La programmation actuelle des machines outils

La programmation actuelle des machines outils se fait par l'intermédiaire du langage G ou G-code dont les principes sont regroupés dans la norme Iso 6983 et annexes.

1.1.1. La norme Iso 6983

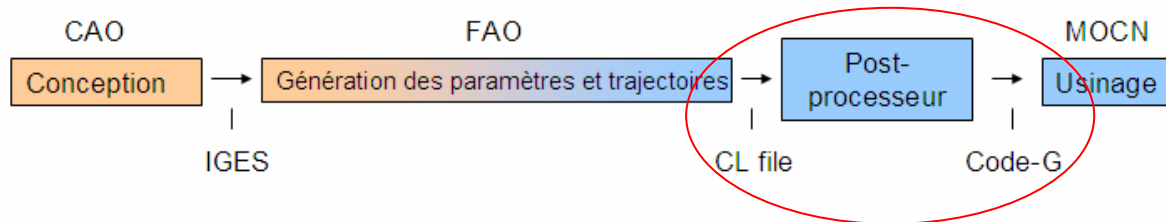
A l'origine, le code G est basé sur un principe de programmation qui remonte à la période des cartes perforées, au début des années 60. Il a tout d'abord été développé par l'EIA (Electronic Industries Alliances) et a été normalisé sous la référence RS274D ou ISO 6983 en février 1980.

Cette programmation s'appuie sur des fonctions préparatoires de type G. Elle est complétée par des fonctions auxiliaires (de type M) et technologiques (F, S, etc...). Par l'intermédiaire de ce code, l'utilisateur communique à la machine un ensemble d'instructions explicites. Un programme en G-code est donc la traduction d'une suite d'actions et de déplacements élémentaires (ligne droite ou arc de cercle par exemple) qui permettent de générer des conditions d'usinage ainsi qu'une trajectoire plus ou moins approximée.

Ce qu'il en est aujourd'hui du respect de la norme ISO dans la programmation des MOCN : Depuis l'établissement de la norme ISO 6983, et avec l'évolution rapide des technologies, de nombreuses extensions ont été ajoutées pour tenir compte des nouveautés et des nouvelles capacités des machines-outils. Ces extensions, bien que souvent utiles chez des constructeurs différents de Directeur de Commande Numérique, n'entrent pas dans la norme et compliquent la tâche des logiciels de FAO, qui doivent créer les lignes de ce langage pour un DCN particulier. À côté de l'ISO, de nouveaux codes sont apparus, différents selon le Directeur de Commande Numérique. Ils intègrent de plus en plus, en plus de ISO, des langages propres aux constructeurs de DCN (symbolique, C, etc.) ainsi que des interfaces de programmation conversationnelle destinées à simplifier la programmation (CN Mazak, Heidenhain, Selca, Siemens, etc.).

1.1.2. Les post-processeurs

Si l'on reprend le schéma simplifié de la chaîne numérique en usinage, on observe que les post-processeurs ont pour rôle de transformer un fichier CL (cutter location file) OU APT en un fichier adapté pour être lu par la commande numérique, c'est-à-dire en code G :



Le post processeur est donc un élément indispensable de la chaîne numérique actuelle dans la mesure où il permet d'adapter le programme aux spécificités d'une machine en particulier. Il existe donc des post-processeurs différents pour chaque constructeur et chaque modèle de machine. De la même façon, chaque logiciel de FAO a lui aussi ses propriétés qui font que le post-processeur d'un logiciel A ne sera pas le même qu'un post-processeur d'un logiciel B pour la même machine outil ! Il existe donc aujourd'hui presque autant de post processeurs que de combinaisons entre logiciels de FAO et machines : on estime leur nombre à presque 4500 variantes principales [Hardwick 01].

Outre le fait qu'ils constituent une étape supplémentaire entre la FAO et l'usinage (donc pertes possibles d'informations par exemple par approximation des trajectoires), la lecture du code G et la visualisation des trajectoires d'usinage qui en résulte s'avèrent difficile. De plus, il apparaît très clairement que l'utilisation de post-processeurs ne permet pas de répercuter sur la CAO-FAO les modifications qui ont pu être apportées au pied de la machine sur le code G (pas de « feed-back » possible).

1.1.3. Les lacunes du code G

Ainsi, d'une manière générale, le code G s'intéresse à programmer une trajectoire en respectant les mouvements des axes machine plutôt que de se concentrer sur les besoins de l'usinage en respectant la pièce. Nous pouvons en particulier noter les principales lacunes qui posent problème lors de la programmation et de manière plus générale dans l'intégration de l'usinage dans la chaîne CAO-FAO-CN :

- La sémantique peut parfois s'avérer ambiguë
- Les constructeurs rajoutent parfois des extensions au langage pour combler les manques et s'adapter à l'évolution des technologies. La 'portabilité' d'un programme s'avère alors impossible entre les différents fabricants.

- Le flux de l'information est unidirectionnel : l'absence de 'feedback' possible de la production à la conception entraîne des difficultés de communication et de correction. De même, la préservation et la capitalisation des expériences se révèlent compliquées.
- L'utilisation du code G rend les modifications au pied de la machine, dans la CN, difficiles et laborieuses ainsi qu'un contrôle limité de l'exécution du programme.
- L'utilisation de post-processeurs rajoute une étape et une perte d'information entre le modèle CAO et l'usinage de la pièce. La vérification de la conformité de la pièce qui doit être usinée avec le modèle CAO est alors complexe voire impossible.
- Le G-code n'est pas bien adapté pour la programmation des courbes complexes. Des pertes d'informations peuvent être engendrées lors de discrétisations et d'approximations.

Pour conclure, nous pouvons voir que le G-code, qui est basé sur des principes qui commencent à dater, ne répond plus aux exigences de programmation des usinages nouveaux et que son emploi constitue une rupture dans la chaîne numérique.

1.2. Evolution de la programmation des MO

Suite à ce que nous venons d'évoquer, le besoin de changer et de simplifier le mode de programmation se fait ressentir, tant au niveau des interfaces proposées par les constructeurs qu'au niveau de la recherche, avec le développement de nouveaux formats de programmation visant à une meilleure intégration de la chaîne numérique.

1.2.1. Du coté des constructeurs

Des efforts sont mis en oeuvre pour créer des interfaces Homme/Commande Numérique. plus accueillantes, plus simples, plus pratiques et fonctionnelles.

Quelques machines CNC utilisent des programmes "Interactifs", qui permettent de programmer avec des Agents en cachant partiellement ou totalement le G-code. Quelques exemples populaires sont Mazak Mazatrol, Hurco Ultimax et le langage interactif Mori Seiki's CAPS [Mazak] [Hurco] [Mori].

Dans le cas de Mazak, par exemple, une interface graphique permet de visionner le déroulement de la simulation d'usinage en 3D. De plus, la programmation sur la machine ne se fait plus en intervenant directement sur le programme en code G mais par intermédiaire d'une interface plus simple où les paramètres à modifier sont organisés de manière plus lisible.



Fig. 4 : DCN Mazak MAZATROL



Fig. 3 : Interface MAZATROLE

On voit donc apparaître le besoin de se détacher d'une programmation utilisant un G-code désuet, dont la syntaxe n'est ni intuitive ni explicite et dont la visualisation et les modifications au pied de la machine sont loin d'être simples. Ces nouvelles initiatives de la part des constructeurs sont cependant souvent basées sur l'utilisation du G-code qui, même s'il n'est plus directement visible par l'utilisateur, est toujours présent, avec tous les inconvénients que nous avons décrits précédemment.

1.2.2. Nouvelle vision de la maquette numérique

La recherche s'attache à développer actuellement un nouveau format pour la maquette numérique basé sur le format Step : le Step_NC. Celui-ci a pour objectif de palier aux inconvénients du G-code. La réorganisation de la chaîne numérique ne se restreint pas à la seule programmation des commandes numériques mais vise à l'implémentation d'un seul fichier lors des différentes étapes de la conception jusqu'à la fabrication. Ainsi, sur un noyau central de la maquette numérique au format Step (Standard of Exchange of Products), on construit l'ensemble des éléments nécessaires pour la réalisation du produit. Ainsi, la maquette numérique issue de CAO est enrichie des diverses simulations, est complétée en FAO par le choix de stratégies et de trajectoires outils et peut être directement lue par la machine pour être réalisée. En retour, l'expérience et les changements effectués sur la CN, en FAO, en CAO ou lors des simulations se retrouvent directement dans le fichier Step_NC,

permettant ainsi un flux bidirectionnel entre la maquette numérique et les outils logiciels utilisés, sans perte d'information.

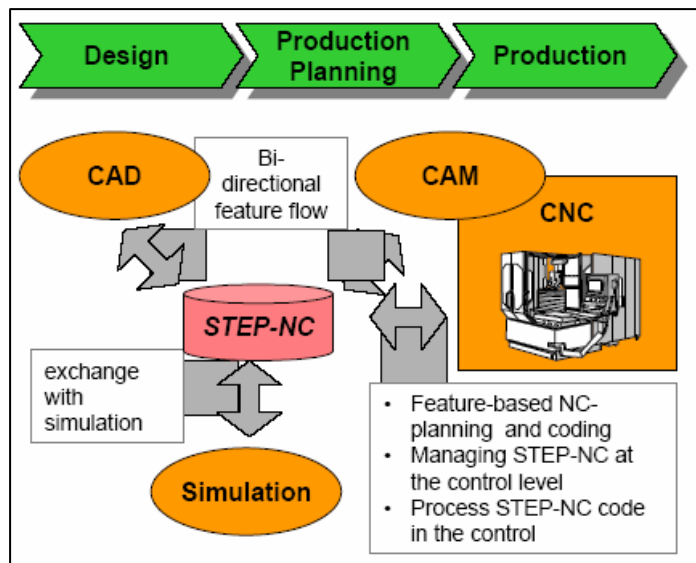


Fig. 5 : La maquette numérique basée sur le Step_NC

Ainsi, cette nouvelle approche permet une meilleure intégration de la chaîne CAO-FAO-CN, en diminuant les pertes d'informations lors des conversions de formats de fichier entre les différents logiciels, en permettant un retour expérience (« feed-back ») et l'implémentation d'un unique fichier (les flux de données sont donc simplifiés et les versions du fichier toujours à jour).

1.2.3. Le Step compliant NC

Dans le cadre de notre étude, nous nous intéresserons plus particulièrement aux évolutions apportées par le Step_NC au niveau de la FAO et de la commande numérique. Le Step compliant NC est, comme son nom l'indique, « conforme » à l'utilisation directe par la commande numérique. Alors que le G-code nécessitait, pour être généré, un post-processeur, le Step_NC est directement lu et interprété par la commande numérique.

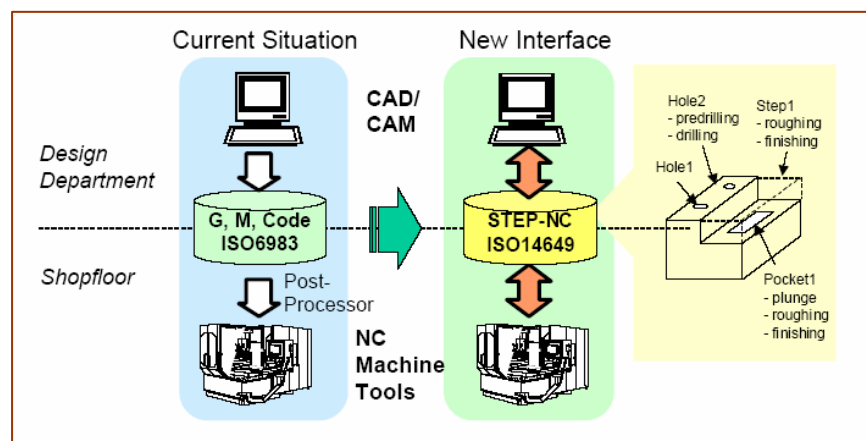


Fig. 6 : Comparaison G-code et Step_NC

Alors que le G-code s'intéressait à décrire « comment faire » et quels étaient les mouvements à effectuer au niveau des axes, le Step_NC décrit « quoi faire » par l'intermédiaire d'une liste d'entités à réaliser. Le fichier Step_NC peut alors être lu et exécuté par n'importe quelle machine, quelque soit son modèle. Cette adaptabilité s'oppose au G-code dont le programme était spécifique à une seule machine.

1.2.4. Structure d'un programme Step_NC :

Le principe fondamental sur lequel repose le modèle Step_NC est que la programmation est basée sur l'objet à réaliser, en terme d'entités d'usinage. Il n'y a plus, comme avec le G-code, de codage direct de séquences de mouvements sur les axes.

Le format du programme est le même que celui d'un fichier au format Step comme décrit dans la norme ISO 10303 :

- La première section est l'entête signalée par le mot-clef « HEADER ». C'est dans cette partie que l'on classe les informations d'ordre général comme le nom du fichier, de l'auteur, la date ainsi que les commentaires.
- La seconde section, signalée par « DATA », est la partie principale. Elle contient l'ensemble des informations concernant la géométrie et les paramètres d'usinage.

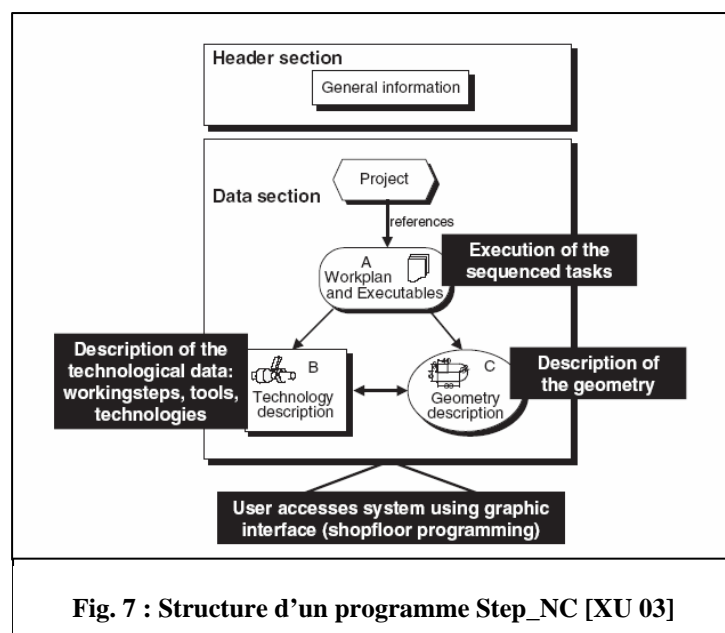


Fig. 7 : Structure d'un programme Step_NC [XU 03]

La section « DATA » contient tout d'abord une première entité « Project » qui sert de point de départ à l'exécution du programme. Elle est ensuite divisée en 3 parties :

- La partie A, « Workplan and Executables », recense les informations concernant l'ordre et le séquençement des opérations à établir. Ainsi que les exécutable associés.

- La partie B, « Technology description », permet de décrire les entités d'usinage, les opérations, les outils, etc.
- La partie C, « geometry description », renvoie à la géométrie de la pièce.

Nous retrouvons ces différentes parties dans l'exemple suivant extrait d'un programme Step_NC :

<pre> ISO-10303-21; HEADER; FILE DESCRIPTION(('A STEP-NC testing file'),'1'); FILE NAME('sample part1.stp',\$,('AUMS'),('', 'Prototype Mill','')); FILE SCHEMA(('STEP-NC milling schema')); ENDSEC; DATA; // Project and workplan #1=PROJECT('Contour',#2,(#3)); #2=WORKPLAN('Work plan',(#4),\$,#5); #3=WORKPIECE('Workpiece',#6,0.01,\$,\$,#8,()); // Working steps #4=MACHINING WORKINGSTEP('Rough Contour',#13,#16,#17); #5=SETUP('main setup',#44,#48,(#51)); #6=MATERIAL('ST-50','Steel',(#7)); #7=PROPERTY PARAMETER ('E=200000 N/mm^2'); #8=BLOCK('Block',#9,260.000,210.000,110.000); // Geometric data #9=AXIS2 PLACEMENT 3D ('BLOCK',#10,#11,#12); yyyy // Manufacturing features #16=GENERAL OUTSIDE PROFILE ('Profile',#3,(#17),#18,#22,\$,\$,\$,#23,\$,\$); </pre>	<pre> // Operation data #17=SIDE ROUGH MILLING(\$,\$,'Contour profile',#38,10.000,#39,#40,#43,\$,\$,\$,20.000,5.000,0.000); #18=AXIS2 PLACEMENT 3D('Position of contour', #19,#20,#21); #19=CARTESIAN POINT('Position of contour', (40.000,90.000,100.000)); #20=DIRECTION('',(0.0,0.0,1.0)); #21=DIRECTION('',(1.0,0.0,0.0)); #22=TOLERANCED LENGTH MEASURE(20.000,\$,\$,\$); #23=COMPOSITE CURVE('Contour Profile', (#24,#25,#56),.F.); yyyy // Tool data #40=CUTTING TOOL('Endmill 10 mm',#41,(),(50.000),50.000); #41=TAPERED ENDMILL(#42,3,.RIGHT..F.,\$,\$); #42=TOOL DIMENSION(10.000,\$,\$,\$,\$,\$); // Machining technology #43=MILLING TECHNOLOGY(\$,.TCP.,\$,3.3333,\$, 0.10,.T.,.F.,.F.); #44=AXIS2 PLACEMENT 3D('Reference point to Machine zero,#45,#46,#47); #45=CARTESIAN POINT('',(20.000,30.000,10.000)); yyyy #56=COMPOSITE CURVE SEGMENT (.CONTINUOUS,.T.,#57); #57=POLYLINE('Second cut of the contour', (#29,#30,#31,#32,#33,#27)); ENDSEC; END-ISO-10303-21; </pre>
--	---

Fig. 8 : Exemple de programme Step_NC

Ainsi, toutes les informations d'ordre géométrique, technologiques et concernant le process sont disponibles dans le fichier Step_NC :

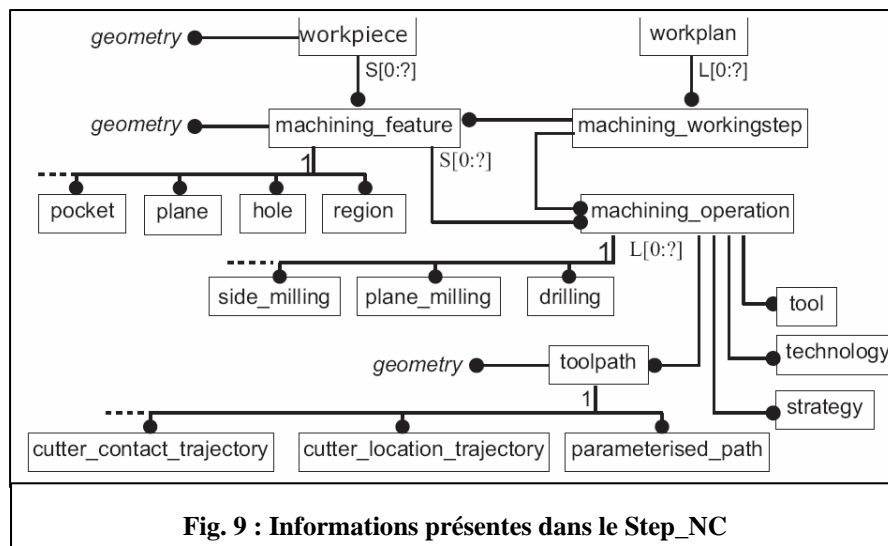


Fig. 9 : Informations présentes dans le Step_NC

Pour prendre le seul exemple des outils, les données qui peuvent être présentes dans le fichier Step_NC sont le type d'outil, sa géométrie, sa durée de vie attendue, etc. Le fichier est donc vraiment complet et dispose de l'ensemble des données permettant d'être directement utilisé par la commande numérique. Cela permet d'éviter les conversions de formats qui sont source de pertes d'information. De plus, la séparation existante entre les données de la géométrie, des opérations et du process permet une simplification du flux et du stockage des données. Il permet aussi de rendre possible l'échange avec des modules situés en amont de la chaîne numérique (par exemple avec la CAO).

1.2.5. Les apports du Step_NC

Le Step_NC offre donc des nouvelles possibilités, non seulement en terme d'usinage, mais aussi au niveau du remaniement qu'il entraîne au sein de la chaîne numérique. Nous pouvons ainsi déjà citer les principales améliorations qu'il serait susceptible d'amener :

- Réduction significative de temps en CAO et dans la préparation des données avec l'existence d'un unique fichier.
- Réduction du nombre de fichiers envoyés de la CAO à la FAO.
- Réduction des temps d'usinage pour les petites et moyennes séries par optimisation intelligente au sein de la commande numérique.
- Eradication des post-processeurs.
- Adaptation de façon indépendante de la machine.
- Plus de pertes d'informations entre les différentes étapes du process : le Step_NC fournit un modèle complet et structuré des données.
- Possibilité de sauvegarde et de 'Feed-back' des modifications liées à la production en direction de la conception.
- Possibilités de transfert Web des informations améliorées.
- Permet de ne garder qu'un seul fichier qui prend en compte toutes les modifications effectuées.

Pour que le Step_NC puisse prendre place au sein de l'industrie et devienne une nouvelle façon de programmer accessible et rentable pour tous les industriels, il reste cependant de gros travaux d'amélioration, de démonstration et de démocratisation à effectuer dans ce sens. Nous allons nous intéresser maintenant aux travaux de recherche qui ont été menés dans cette optique.

1.2.6. Définitions et normes sur le Step_NC

Tout d'abord, le format STEP (Standard of Exchange of Products) porte sur la représentation et l'échange de données de produits et a pour objectif d'intégrer les processus de conception, de développement, de fabrication et de maintenance de ces derniers. Il a été normalisé sous le numéro ISO 10303. Il est intéressant dans la mesure où il couvre un très vaste domaine de connaissances correspondant à l'ensemble des catégories de produits et avec le point de vue de tous les métiers et concernant toutes les phases de vie du produit (conception, fabrication, maintenance...).

Concernant le standard Step_NC, il existe 2 comités ISO différents avec des centres d'intérêts distincts. Le groupe ISO TC 184/SC1 travaille sur l'ISO 14649 : c'est le modèle ARM (Application Reference Model). Quant au groupe ISO TC 184/SC4, il travaille sur le STEP AP238 : c'est le modèle AIM (Application Interpreted Model). La principale différence entre ces 2 modèles est le degré auquel ils font appel quant aux méthodes de représentation et à l'architecture STEP [Feeney & all 03]. Quelques critères de comparaison sont fournis dans le tableau suivant :

Critères de comparaison	Modèle ISO 14649 (ARM)	Modèle 10303-238 (AIM)
Volume de stockage	10 fois moins que AIM	10 fois plus que AIM
Programmation	facile	plus complexe
Relisible par l'homme	difficile	presque impossible
Compatibilité avec le STEP	en partie	totalemment
Données	les informations originales du design sont abandonnées	les informations originales du design sont conservées

ISO 14649 et ISO 10303-238 peuvent être vues comme 2 différentes méthodes d'implémentation du standard Step_NC :

- ISO 14649 est plus à même d'être utilisée dans un environnement dans lequel les systèmes de FAO possèdent des informations exactes concernant le moyen de production et son environnement. Ce standard n'a pas de mécanisme spécifique pour intégrer les autres types de données au format STEP. Ainsi, le flux bi-directionnel des données entre la conception et la fabrication est rendu plus difficile.
- STEP AP-238 est plus adaptée à une intégration complète de la conception et de la fabrication. Ce standard permet d'intégrer toutes les informations du fichier STEP ainsi que certaines construites à partir de l'ISO 14649. L'échange bi-directionnel des données est ainsi possible.

Cependant, l'un des problèmes de l'AP-238 est qu'il ne donne qu'une vue informationnelle des données contrairement au standard ARM qui en fournit une vue fonctionnelle. En outre, dans le premier cas, les données peuvent s'avérer plus difficiles à déchiffrer, sa structure est plus complexe et la taille des fichiers nécessite plus d'espace de stockage.

1.3. Les groupes de travail sur le Step_NC :

De plus en plus de laboratoires et de grands industriels se regroupent aujourd'hui à travers certains projets visant à développer le Step_NC. De plus, le groupe TC 184 de l'ISO a pour mission le développement de ce protocole Step_NC ainsi que sa normalisation. Nous allons nous intéresser à rendre compte de l'état actuel de la recherche dans ce domaine ainsi que des perspectives à venir.

X.W.XU fait un récapitulatif des travaux de recherche ainsi que des projets majeurs qui ont été menés [XU05]. Ces projets ont eut lieu à plusieurs échelles : international, au sein d'un regroupement de quelques pays d'une même région ou sur un plan national. Les principaux d'entre eux sont les suivants.

1.3.1. IMS (Intelligent Manufacturing System Projet) Step_NC projet

Le projet IMS Step_NC, approuvé en Novembre 2001, regroupe un ensemble d'actions de recherche au niveau international avec des laboratoires d'Europe, de Suisse, de Corée et des USA. Font partie de ce projet, des utilisateurs, des fabricant de systèmes CAO, FAO, de machines outils et CN, etc. Les coordinateurs régionaux sont Siemens pour l'Europe, CADCAMation pour la Suisse, Step Tools pour les USA et ERC-ACI pour la Corée. Siemens a aussi pour rôle la coordination internationale de ce groupe. Ce projet représente un point culminant quant au nombre de sous-projets qu'il représente au travers des différents groupes

Region	EU	Switzerland	Korea	USA
Technologies covered	Milling, Turning, Wood/glass cutting, Stone machining, Inspection	Wire/Sink EDM	Milling, Turning, Rapid Prototyping, XML-formatted STEP-NC data structure	AIM for Milling & Turning (STIX)
End user	<ul style="list-style-type: none"> • Daimler-Chrysler • Volvo • Franci (Italy) • Progetti 	<ul style="list-style-type: none"> • Derendinger • Wyss 	<ul style="list-style-type: none"> • Samsung 	<ul style="list-style-type: none"> • IRB, including • Boeing • Lockheed Martin • General Electric • GDLS • General Motors...
Machine-tool manufacturer	<ul style="list-style-type: none"> • CMS (Italy) 	<ul style="list-style-type: none"> • AGIE • Starrag 		
Control manufacturer	<ul style="list-style-type: none"> • Siemens* • OSAI (Italy) • Fidia • Open Mind • Dassault 			
CAM manufacturer		<ul style="list-style-type: none"> • CADCAMation* 	<ul style="list-style-type: none"> • Cubictek 	<ul style="list-style-type: none"> • STEP Tools* • Gibbs & Associate • BA Solutions • Numerical Control Services • Louisiana Center for Manufacturing Sciences • Lawrence Livermore National Laboratories
Research institute	<ul style="list-style-type: none"> • WZL (RWTH AACHEN) • ISW (University of Stuttgart) 	<ul style="list-style-type: none"> • EPFL • EIG I-tech 	<ul style="list-style-type: none"> • ERC-ACI* • KIST 	
Association	<ul style="list-style-type: none"> • KTH • CECIMO (Belgium) 	<ul style="list-style-type: none"> • AMT 	<ul style="list-style-type: none"> • NRL-SNT 	<ul style="list-style-type: none"> • NIST • Department of Energy • Army's National Automotive Center (NAC)

Fig. 10 : Tableau récapitulatif des participants aux projets de l'IMS Step_NC [XU & HE 03]

de recherche. Il a notamment donné lieu au projet Européen Esprit Step_NC, au projet Super Model et au projet SMS et RAMP que nous allons détailler par la suite.

1.3.2. Projet Européen Esprit Step_NC

Né d'un ensemble de groupes de recherche créé au début des années 90 et ayant mené à la formation du consortium Européen Step_NC, le projet Européen Esprit Step_NC débute en janvier 1999 pour s'achever en décembre 2001. Dans le cadre de ce projet, un prototype pour le fraisage a été mis au point et présenté en Octobre 2001, à Aachen en Allemagne. Les modules CAO d'Open Mind (hyperfact) et de Dassault Système (Catia V5) ont permis de générer le programme Step_NC. Ce programme a ensuite pu être exécuté par une CN Sinumeric 840D qui avait été adaptée dans ce but. Plusieurs autres travaux concernant le prototypage rapide ont de même été menés.

En 2002, le projet Step_NC Européen est passé dans une seconde phase. En complément des travaux de la phase 1 à poursuivre, un axe de recherche concernant les mécanismes de retour (ou feedback) de l'information a été développé et a pour objectif de surveiller la précision lors de l'usinage. Dans ce but, des travaux sur l'inspection sont menés. En parallèle, des travaux d'évaluation et de vérification du modèle de données ont fait l'objet d'investigations.

1.3.3. Projet « Super Model » aux USA

Ce projet a été fondé par le NIST (National Institute of Standards and Technology) et attribué en octobre 1999 à Step Tools inc., en association avec d'autres entreprises collaboratrices. Le but de ce projet était de construire une base de données qui pourrait contenir l'ensemble des informations nécessaires à la fabrication de pièces. Les plus grandes firmes ont pris part au projet, comme par exemple Boeing, General Electrics, General Motors, Daimler-Chrysler, etc.

Une des clefs de la base de donnée du Super Model est l'utilisation d'interfaces au format XML, ce qui constitue un moyen pratique de lier les trajectoires, les stratégies d'usinage et les caractéristiques outils aux données de géométrie, d'entité et de phases d'usinage.

Un outil post-processeur permettant, à partir d'un fichier Step, de générer un fichier Step_NC, dont l'utilisation est indépendante de la machine, a été développé : ST-Plan. Il est basé sur une technologie développée par le Honeywell Federal Manufacturing & Technologies (FM&T) et permet d'agir sur des paramètres tels que les tolérances, les entités, les conditions d'outil, etc.

Une autre bibliothèque de code libre nommée STIX a aussi été développée dans le but de regrouper les avantages des versions AIM (Application Interpreted Model) et ARM (Application Reference Model) du Step_NC.

1.3.4. Le projet SMS : Step Manufacturing Suite

Ce projet est coordonné par le SCRA (South Carolina Research Authority) et est financé par l'US Army et sa branche TACOM (Tank-Automotive and Armament Command), ainsi que le National Automotive Center. Il s'appuie sur les concepts développés par les pays ayant participé au groupe de travail TC 184 SC4 de l'ISO. Il a pour but d'implémenter l'architecture dans le cadre de la production des pièces en vue d'applications commerciales ou de défense.

1.3.5. Autres projets concernant le Step_NC

Plusieurs autres projets sont ou ont été menés pour des applications du Step_NC :

- Nous pouvons citer le projet RAMP (Rapid Acquisition of Manufactured Parts) qui a mené au développement et à l'implémentation du standard Step AP224 (ISO Standard 10303-224). AP (Application Protocol) 224 permet d'obtenir des spécifications de données de produit indépendantes du fournisseur qui comportent l'ensemble des informations nécessaires pour fabriquer et assembler des pièces. Il a tout d'abord été développé aux USA puis repris en Grande-Bretagne. Il semblerait qu'il soit utilisé aujourd'hui dans l'industrie.
- Toujours en Grande Bretagne, un projet débuté en mai 2004 et mené par le AMST Centre de l'université de Loughborough vise à étendre les travaux de recherche européens en utilisant la commande numérique Siemens 840D à l'inspection et en particulier l'intégrations du standard Step_NC entre la MOCN et la machine à mesurer tridimensionnelle.

1.4. Les travaux de recherche effectués sur le Step_NC en fraisage

1.4.1. Travaux de recherche en Allemagne

Il existe 2 principaux laboratoires en Allemagne travaillant sur le Step_NC : le ISW à Stuttgart et la WZL RWTH à Aachen. Ces laboratoires travaillent beaucoup avec le constructeur de commandes numériques allemand Siemens.

L'ISW s'intéresse en particulier au tournage et a développé un prototype nommé STEPTurn [Storr & Heussinger 02]. Ce système permet d'agir entre la CAO et la FAO en lisant les données de géométrie à partir d'un fichier Step AP-203 Part 21 et en générant le fichier Step_NC par reconnaissance de formes puis des phases. De plus, il a été développé une méthodologie d'optimisation automatique de planification des phases en tournage [Storr & All 02].

Une autre partie de la recherche effectuée à l'ISW concerne le 'Feedback' de la commande numérique à la FAO [Denkena & all 02].

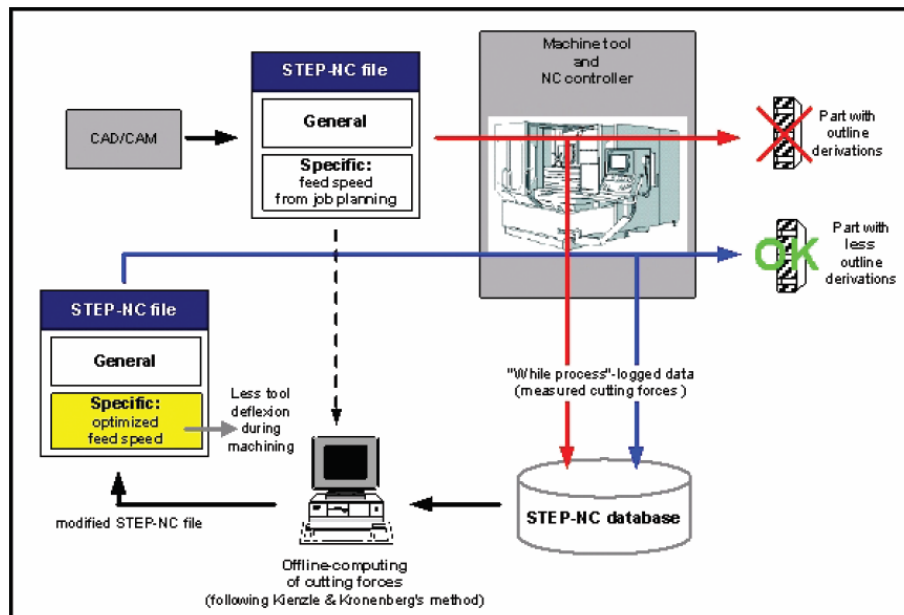


Fig. 11 : Feedback hors-ligne pour le Step_NC [Denkena & all 02]

Un algorithme de compensation off line a été développé et permet de modifier le programme Step_NC pour diminuer les effets des déformations de l'outil en UGV. Pour ceci, ils réinjectent les valeurs mesurées des efforts de coupe dans une base de donnée nommée STEP_NC database. A partir de modèles de déformation d'outil, il est alors possible de régénérer le programme Step_NC corrigé, dont les vitesses d'avance ont été modifiées et optimisées.

Le laboratoire WZL de machine outil et production s'intéresse plus particulièrement à la conformité et à la possibilité d'utiliser le format Step dans les commandes numériques. Il a été le lieu de développement du premier prototype de commande numérique industrielle conforme au format Step_NC. Celui-ci est basé sur une CN Siemens 840D [Weck & al. 01] [Weck & al. 03] [Wolf & al. 03]. Une interface graphique a été développée en utilisant Shop Mill (un outil de programmation machine) et Sinumerik 840D HMI (interface homme machine).

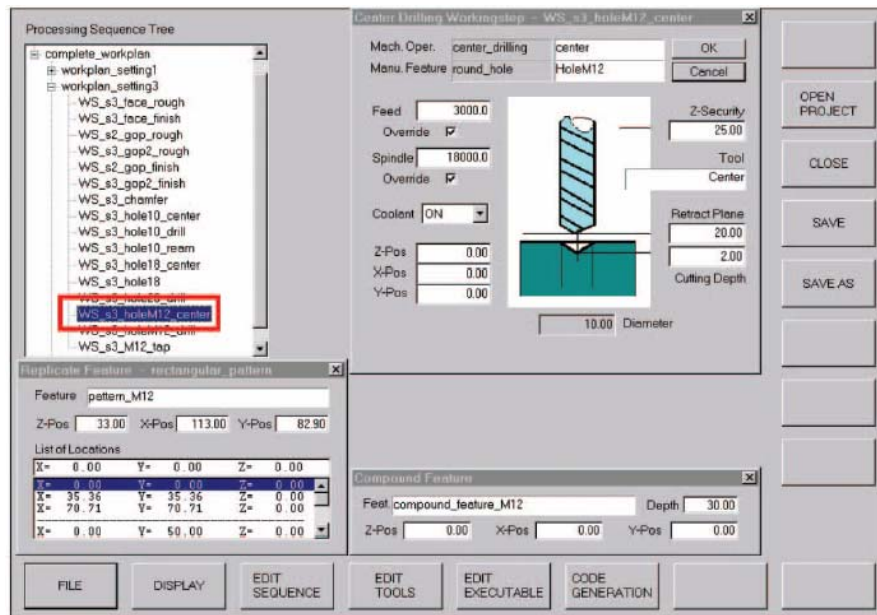


Fig. 12 : Interface graphique pour le Step_NC [Weck& Wolf 02]

Cette interface permet d'analyser les programmes Step_NC. Les opérations d'usinage sont exécutées à partir de cycles préexistants de la commande numérique. La figure suivante illustre l'architecture du contrôleur Step_NC Siemens :

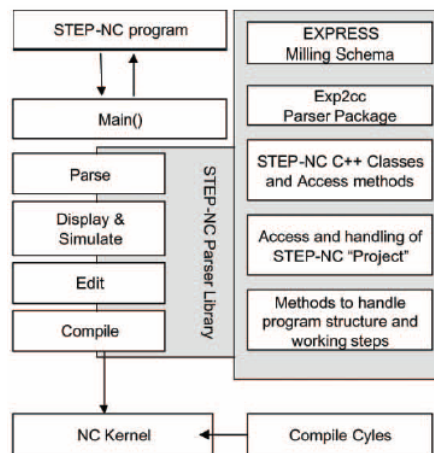


Fig. 13 : Architecture Siemens Step_NC [Weck& Wolf]

Il est aussi possible de compiler directement les opérations avec des commandes rapportées et de les traiter dans le but de contrôler des formats d'interpolation spécifiques.

Ces travaux de recherche ont donné lieu à une présentation en octobre 2001 dans le cadre du projet Européen Esprit où une chaîne Step_NC complète a pu être mise en œuvre, de la CAO jusqu'à la machine.

1.4.2. Travaux de recherche au Royaume-Uni :

Des travaux ont été menés, en dehors des projets liés à la collaboration dans le cadre du projet européen Step_NC, par 2 laboratoires :

L'AMST centre, de l'université de Loughborough, s'est intéressé au développement d'un système AB-CAM (Agent Based CAM) [Newman & al. 03] [Allen & al. 05]. AB CAM a pour objectif d'implémenter le modèle de données Step avec un système de données dont le support est Step compliant NC, c'est-à-dire le format Step conforme à l'utilisation sur la commande numérique. A partir d'un fichier Step généré en CAO/FAO qui permet de fournir la géométrie ainsi que des éléments concernant les paramètres de fabrication, le système AB-CAM permet l'extraction des formes, la génération de la gamme du processus ainsi que des programmes spécifiques au contrôleur

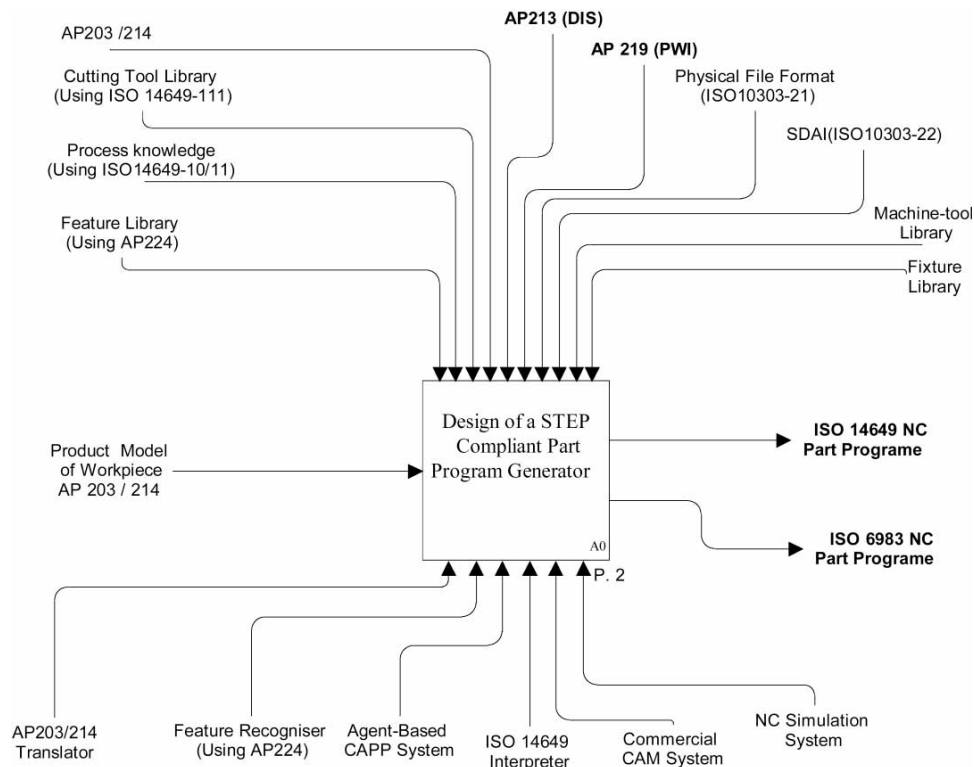


Fig. 14 : Structure du système AB-CAM [Newman & al. 03]

Il a été proposé de développer un traducteur AP-224 qui soit capable de capturer les informations de forme à partir du modèle CAO. Si ces informations ne sont pas disponibles, il est possible d'utiliser un système de reconnaissance automatique de forme. Concernant les gammes de fabrication, le générateur « Step compliant process plan » sélectionne un type de processus, une machine, des outils, des paramètres de coupe, des mises et maintiens en position,...pour convertir les entités en opérations dans le fichier Step_NC. Pour finir, AB-CAM renvoie un fichier Step_NC natif.

Ces travaux servent de base au nouveau groupe de recherche EPSRC de l'université de Loughborough.

Dans le second projet, RAMP, le modèle est converti en un fichier Step AP224 en utilisant un logiciel nommé STEPTrans pour s'affranchir des droits propriétaires. Le système LOCAM a été développé par le groupe LSC. Les fichiers Step AP-224 constituent le format « neutre » des informations nécessaires à la planification du processus d'usinage et ne sont pas prédéterminés et spécifiques à un processus de fabrication particulier. Une base de données contenant les règles et contraintes de fabrication spécifique à une entreprise est alors introduite dans la planification. Ceci permet donc de travailler sur n'importe quel fichier Step AP224, quelque soit l'endroit où le logiciel est employé pour créer le fichier CAO. Les données sont ainsi gardées génériques jusqu'à la FAO.

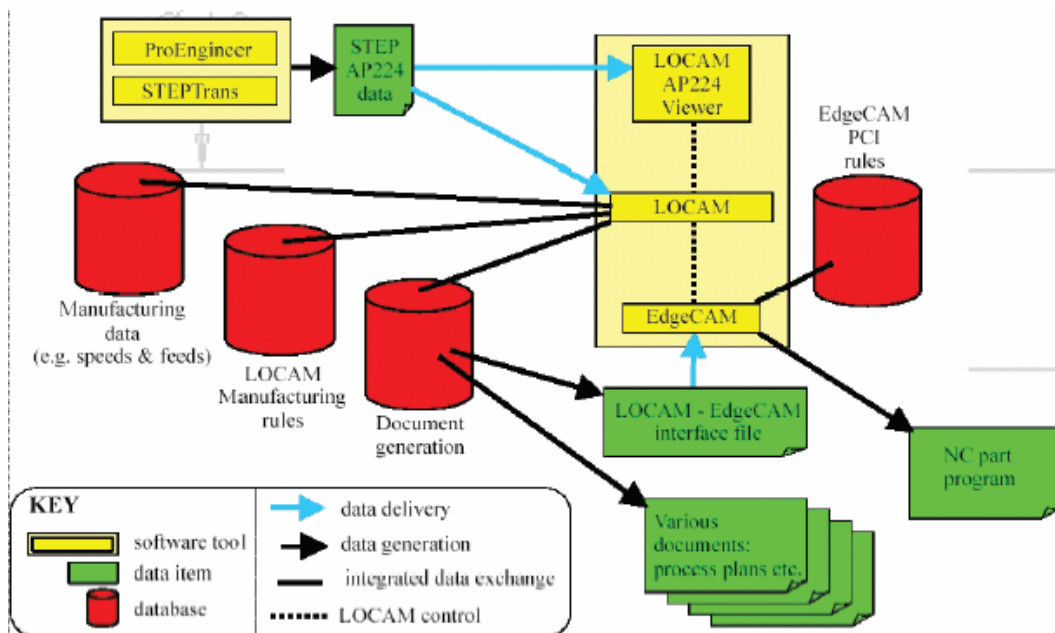


Fig. 15 : Flux des données dans RAMP

Des tests ont été menés dans le cadre d'un projet avec la UK Navy et a pu aboutir à des gains de temps de l'ordre de 3,5 par rapport à des méthodes conventionnelles pour le cas de nouvelles commandes. Ce facteur grimpe à 9 pour un système de production RAMP totalement intégré pour des commandes mixtes (nouvelles commandes et commandes renouvelées). Le format Step_NC est actuellement en cours d'implantation dans le projet RAMP, ce qui est relativement aisé dans la mesure où la définition des formes dans le Step_NC est principalement le même que dans le Step AP224.

1.4.3. Travaux de recherche aux USA :

La division ISD (Intelligent System Division) du NIST s'occupe, depuis le début des années 80, des problématiques de génération de code pour les commandes numériques, et en particulier à partir de données Step.

A partir de la fin des années 90, un système de contrôle et d'usinage basé sur les entités et formes a été développé : le FBICS (Feature Based Inspection and control System) [Kramer & all 01 04]. L'un de ses principaux objectifs est de tester les possibilités d'utilisation des méthodes et modèles Step. Les deux principales capacités du système FBICS sont de pouvoir générer automatiquement une planification au niveau du processus de contrôle et de l'exécuter pour la fabrication ou l'inspection de la pièce. Il contient des modèles d'outils de coupe, de gammes de fabrication, de forme des pièces avant et après usinage, de sélection des paramètres d'usinage, du bridage des pièces...

Les travaux de recherche concernant le Step_NC commencés par ailleurs ont menés au développement d'un interpréteur de fichier ISO 14649 (ARM). Son but est de lire le fichier Step et de générer les commandes machine. Les informations machine que l'interpréteur peut traiter sont le perçage, l'alésage, le fraisage ébauche et finition de surfaces planes et de poches rectangulaires. L'exemple en annexe F de l'ISO 14649 part 11 a pu être traité (pièce Fig. 16).

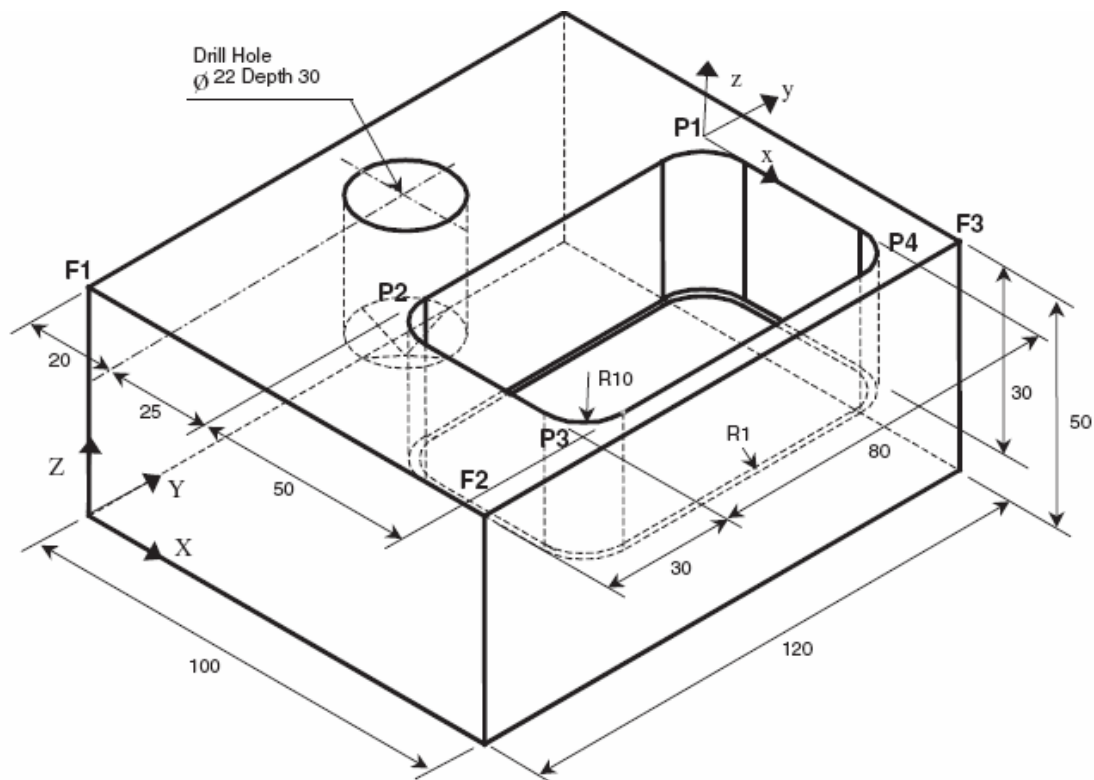


Fig. 16 : Pièce test de l'ISO 14649 Part 11

Un autre interpréteur concernant l'AP238 (AIM) est aussi en cours de développement : le premier objectif est de pouvoir l'exécuter, de la même manière que l'interpréteur ARM, sur l'exemple de la pièce test ci-dessus.

La recherche concernant le Step_NC aux USA se fait aussi par l'intermédiaire de STEP Tools Inc. : cette entreprise a développé ST-plan, un système Step compliant qui permet de créer des programmes CN indépendants du type de machine en se basant sur une approche dont l'entrée est un fichier Step et la sortie un fichier Step_NC. ST-plan comporte 2 modules principaux, FBMach (Feature-Based Machining) et FBTol (Feature-Based Tolerancing). FBMach permet la reconnaissance des entités d'usinage ainsi que la définition des opérations associées et FBTol concerne la définition des tolérances.

Le module FBMach a été développé par FM&T (Federal Manufacturing and Technologies de Honeywell) [Brooks & Greenway 95],[Albert 01]. Il contient des algorithmes de reconnaissance de forme ainsi qu'une librairie d'entités d'usinage. Le système peut créer à la fois des entités basées sur des surfaces et des volumes. Concernant la reconnaissance de forme, il en existe 3 types : - automatique (un algorithme cherche et crée les entités sans intervention de l'utilisateur), -manuel (l'utilisateur clique sur toutes les faces de entités et en définit le rôle), - hybride (l'utilisateur sélectionne une ou plusieurs faces pour guider le logiciel). Suite à cette reconnaissance de forme sont créés les opérations d'usinage. La gamme est alors complétée en organisant la séquence de ces opérations. Les informations sont sauveées sous la forme d'un fichier Step_NC (AP 238). Une des qualités de FBMach est qu'il est possible de voir la pièce avant et après chaque opération d'usinage, ce qui permet à l'utilisateur de voir la progression étape par étape.

Step Tool Inc. a par ailleurs développé un plug-in nommé ST-machine dont il existe une version adaptable pour le logiciel GibbsCAM et une pour MasterCAM. Ce plug-in permet d'utiliser du Step_NC sur une machine spécifique sans avoir à modifier sa structure ou sa commande. De plus, une librairie libre de droits appelée STIX a été développée pour pouvoir compiler des données au format AP-238 dans un système de FAO ou de commande numérique.

Ainsi le schéma complet de la chaîne numérique proposée par Step Tools Inc. est représenté Fig. 17 :

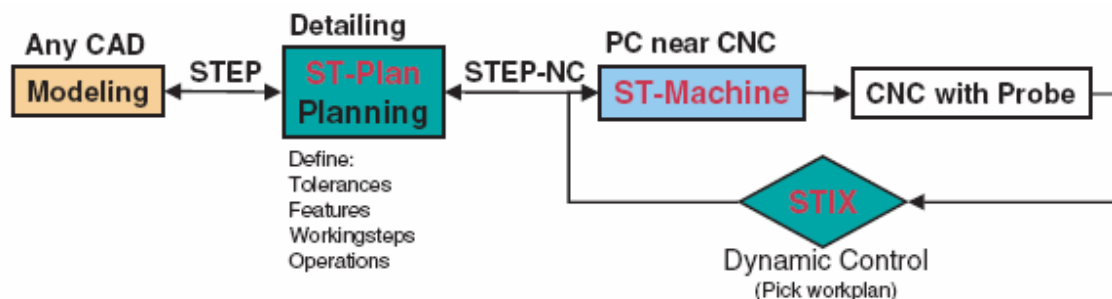


Fig. 17 : Chaîne numérique « Step-compliant CAD/CAPP/CAM/CNC » proposée par Step Tools Inc. (IMS Consortium 03)

Plusieurs démonstrations ont été menées sur des sites participant : en janvier 2003, un test utilisant le plug-in pour le logiciel GibbsCAM a été mis en œuvre dans un laboratoire lié à la NASA et a permis de générer pour la commande numérique en fraisage à partir d'un fichier d'entrée en Step AP-238 [Anonyme 03]. Les résultats ont permis de mettre en avant une réduction dans le temps de programmation de 105 mn à 12 mn. Une seconde démonstration a eu lieu en juin 2003 avec pour base le plug-in adapté au logiciel MasterCAM.

1.4.4. Travaux de recherche en Nouvelle Zélande :

Une première partie de la recherche en Nouvelle Zélande s'intéresse au développement d'un système permettant le travail collaboratif entre différents postes de CAO et FAO basé sur le Step et le Step_NC. Il utilise différentes technologies d'échange d'informations et de réseaux comme ASP, ActiveX, OpenGL ainsi que le XML pour les échanges via Internet.

Une seconde partie de la recherche concerne le développement d'un scénario d'usinage sans utilisation de code G [Xu 05b]. Dans un premier temps, une machine utilisant exclusivement le code G (commande de tournage HERCUS PC200) a été remise à jour par l'implantation d'un système de contrôle plus ouvert (Compumotor Motion Control System) qui permet de créer une interface avec la FAO par l'intermédiaire de langages comme le Virtual Basic, le Visual C++ et le Delphi. Ce système de contrôle utilise cependant son propre langage de contrôle des axes (le 6K Motion Control langage). Une librairie de fonctions 6K a été développée pour couvrir différentes opérations d'usinage.

1.4.5. Travaux de recherche en Corée :

Le laboratoire NRL-SNT (National Research Laboratory for Step_NC Technology) de l'université de Pohang s'applique, depuis 1997, à développer un système complet basé sur le Step_NC. Dans ce but, il a été réalisé un système de commande numérique autonome basé sur le Step_NC et il a été construit un système de production virtuel à architecture ouverte [Suh & al. 02a] [Suh & al. 02b] [Suh & al.03].

Ainsi, ce nouveau contrôleur de commande numérique, nommé le système Coréen Step_NC, comporte un composant clef, le système SFP (Shop Floor Programming system). Ce dernier permet de passer de la CAO au Step_NC, en générant un programme Step_NC (ARM) à partir d'un fichier CAO Step AP 203. Il est aussi possible de prendre comme entrée un fichier ISO 14649 pour en régénérer une nouvelle version prenant en compte des mises à jour et les spécificités de la commande.

Les étapes de fonctionnement du système SFP sont les suivantes (voir figure ci dessous) : après avoir interprété (A1) le fichier Step AP203, un algorithme de reconnaissance de forme et entités est appliqué (A2) permettant ainsi la génération des opérations ainsi que

leur planification (A3). Le programme Step_NC (ISO 14649) peut alors être généré (A5) et visualisé (A6).

Si le fichier d'entrée est en Step_NC (ISO 14649), l'interpréteur permet d'extraire de celui-ci les entités de forme ainsi que les opérations. On rajoute ensuite, à partir d'une base de donnée, des informations natives au programme Step_NC original. Cela permet donc de convertir un programme Step_NC générique en programme natif.

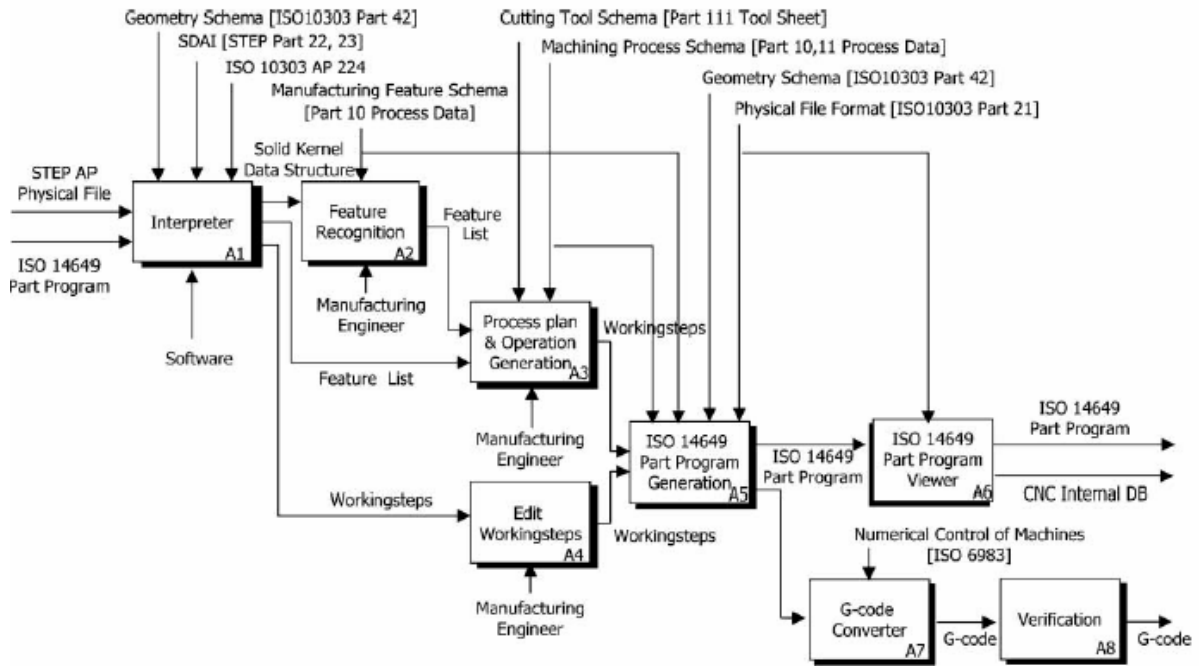


Fig. 18 : Diagramme SADT du système SFP [Suh & Cheon 02]

De la FAO au plan processus natif : Les informations provenant du fichier Step_NC sont converties dans le format interne de données, c'est-à-dire une séquence processus sous la forme d'un graphe de séquence (Process Sequence Graph : PSG) [Suh & all 02a]. Dans l'interpréteur, un compilateur EXPRESS convertit le fichier physique en PSG (Fig. 19) en se basant sur des informations comme la géométrie, la technologie de fabrication, les outils, etc.

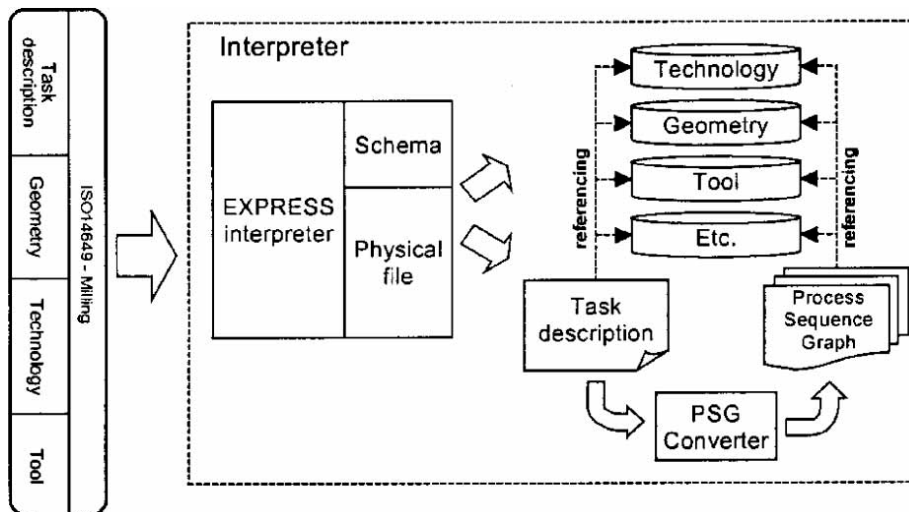


Fig. 19 : Détail de l'interpréteur [Suh & all 02a]

Le PSG représente l'enchaînement des étapes de travail, ces dernières étant décrites en terme d'entités et d'opérations d'usinage (Fig. 21) coordonnées entre elles par des relations ET/OU (Fig. 20).

WORKINGSTEP		
ID	FEATURE	OPERATION
1	planar_face	plane_rough_milling
2	closed_pocket	bottom_and_side_rough_milling
3	round_hole	drilling
4	round_hole	drilling
5	slot	bottom_and_side_rough_milling
6	slot	bottom_and_side_rough_milling
7	round_hole	drilling
8	slot	bottom_and_side_rough_milling

Fig. 21 : Liste des étapes de travail

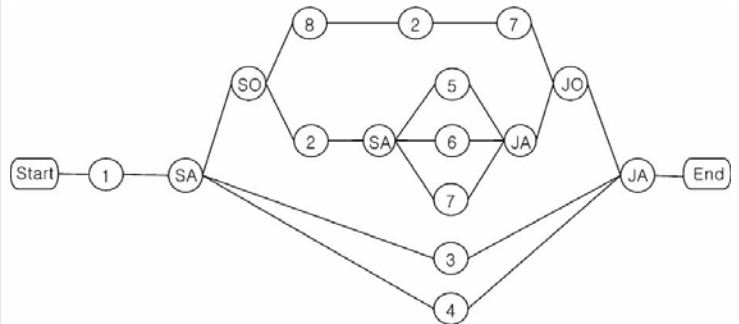


Fig. 20 : PSG : Process Sequence Graph

Ainsi, l'usinage peut se faire dans différents ordres, (exemple : 1-8-2-7-3-4 ou 1-3-4-2-5-6-7 ou 1-3-4-2-5-6-7). Cela permet de conférer plus d'autonomie, de flexibilité, d'intelligence dans l'exécution par la commande numérique. Deux approches sont alors envisageables suivant la machine et son directeur de commande numérique disponibles :

Dans le cas d'une commande numérique compatible avec le Step_NC, il est utilisé un générateur de trajectoires (PosTPG : Tool Path Generator), un simulateur de trajectoire (PosTPS : Tool Path Simulator) et une application pour la CN nommé NCK/PLC [Suh & all 02b]. PosTPG génère la trajectoire pour chaque opération en 4 parties : l'approche, l'usinage, le dégagement et le retour. Les trajectoires peuvent être visualisées grâce au simulateur PosTPS et les données de vérification sont stockées dans la base de données concernant les trajectoires. L'application NCK/PLC est utilisée pour convertir les données du model Step_NC en informations pour les mouvements de la machine. Ces possibilités sont prévues pour être étendues au traitement des interpolations NURBS, à l'usinage et au contrôle de la trajectoire par anticipation, etc.

Dans le cas d'une commande numérique conventionnelle n'acceptant que le code ISO 6983 (code G) [Suh & all 02a], une conversion est effectuée à partir du fichier ISO 14649 en G-code. Cela requiert une génération des trajectoires explicite, avec un emploi des mêmes outils que ceux spécifiés dans le programme ISO 14649. Suite à cela, les trajectoires sont vérifiées.

Mise en œuvre sur la machine à commande numérique : les trajectoires sont donc stockées dans une base de données. Lorsque l'exécution commence, une trajectoire est prise dans la base de données et convertie en code pour la commande suivant le schéma ci-dessous :

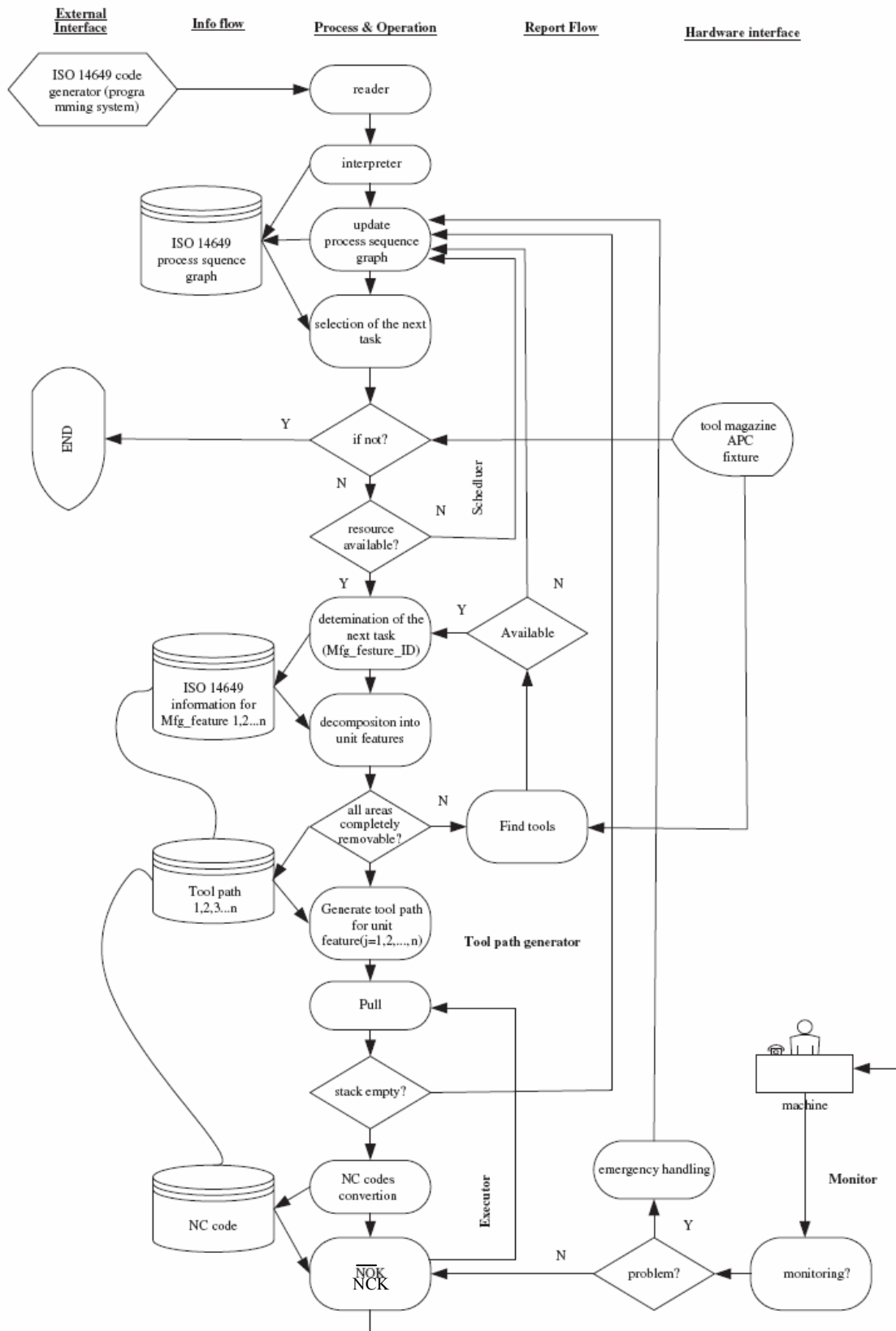


Fig. 22 : Logigramme de la boucle de fonctionnement STEP compliant NC [Suh & all 02a]

Les codes CN convertis sont stockés dans une base de données (NC code). L'algorithme NCK exécute le code CN et met en œuvre la machine en commandant les axes, la table, le magasin d'outil, la broche, la lubrification, etc. Lorsqu'un code CN a été exécuté entièrement, la trajectoire de l'entité suivante est alors extraite de la base de données des trajectoires et ainsi de suite. Cela continue ainsi jusqu'à ce que toutes les opérations aient été effectuées.

1.4.6. Travaux de recherche en France :

Les travaux de recherche en France ont principalement lieu au laboratoire Ircyn (Institut de Recherche en Communications et Cybernétique de Nantes). Les deux axes majeurs de recherche concernent pour l'instant le fraisage et le prototypage rapide [Hascoët & all 06a, b].

Pour le fraisage, les applications en lien avec le Step_NC concernent principalement la génération de trajectoires et la programmation intelligente avec pour objectif la programmation de trajectoires optimisées et corrigées en Step_NC. Dans des précédents travaux, un algorithme d'optimisation de la trajectoire a été mis au point en collaboration avec le laboratoire coréen de Pohang [Suh, Cho, Hascoët 96]. Sur cette base, de nouveaux travaux ont été menés dans le but de pouvoir modifier la trajectoire en fonction des informations directement relevées sur la machine (couples et puissances moteurs, avances réelles...). Ces informations sont alors traitées par une application hors ligne permettant de recalculer et d'optimiser la trajectoire, le but à terme étant de pouvoir modifier la trajectoire en temps réel.

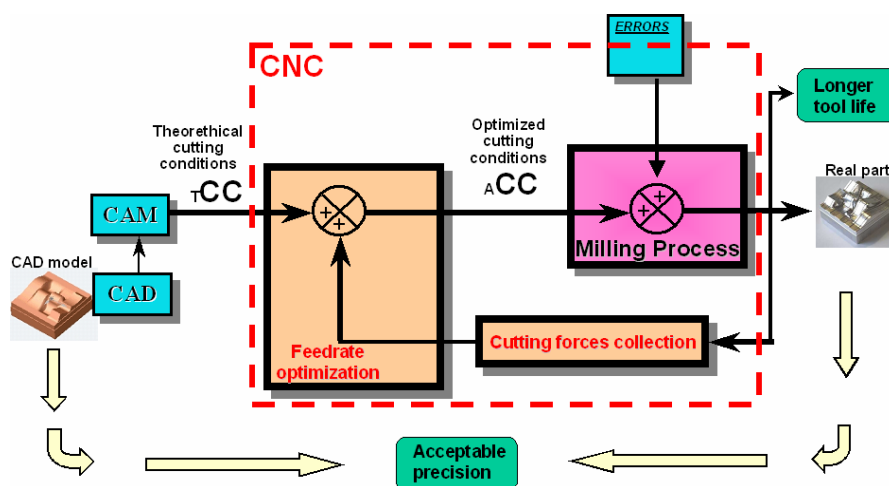


Fig. 23 : Programmation intelligente et Feed-back dans la CN [Hascoët & al. 06b]

En collaboration avec l'université de Pohang en Corée, les travaux actuels sur le Step_NC correspondent, pour ce standard, à la génération de trajectoires et en particulier en tréflage et

en fraisage trochoïdal [Hascoët, Rauch, Suh 07], l'objectif étant de pouvoir enrichir les bases de données concernant le Step_NC de nouvelles fonctionnalités en fraisage.

Ainsi, il reste beaucoup de travail, en particulier pour compléter les bases de données qui serviront à la génération des trajectoires par la CN. De plus, avec la démocratisation de l'ugv et l'arrivée de nouveaux modes d'usinage, l'étude et l'optimisation des trajectoires reste un domaine de recherche prometteur.

1.5. Conclusion

Comme nous venons de le voir dans cette partie, beaucoup de travaux ont été réalisés concernant le Step_NC. Il reste cependant un gros travail de recherche pour démocratiser et implémenter ce format. C'est en effet une nouvelle vision de la chaîne numérique qui s'offre à nous et qui doit entraîner des changements profonds tant au niveau matériel que dans l'approche de l'ensemble des intervenants à toutes les étapes de la chaîne numérique. Il incombe donc aux chercheurs de montrer la réelle faisabilité de cette démocratisation, en partenariat avec des entreprises motrices dans ce projet, et de développer et consolider cette approche. Dans cette optique, de grands laboratoires internationaux participent à des projets liés aux plus grandes entreprises. En France, la recherche dans ce domaine a pris du retard mais de plus en plus de laboratoires commencent à s'intéresser à cette problématique. Dans cette configuration, l'IRCCyN peut avoir une position motrice, tant dans le domaine du fraisage que du prototypage rapide.

2. Optimisation des trajectoires en fraisage

De nombreux travaux ont été réalisés dans le but de trouver de nouvelles trajectoires répondant au mieux aux particularités des nouvelles technologies. Nous allons présenter dans cette partie les spécificités de l'usinage grande vitesse (UGV) ainsi que les principales avancées en terme d'optimisation des trajectoires d'usinage ainsi que de prise en compte des déformations outils.

2.1. L'UGV : une technologie en perpétuel essor

2.1.1. L'UGV : définitions et frontières

Pour que nous puissions prendre la mesure des phénomènes mis en jeu dans l'UGV, il est intéressant d'en donner une définition générique que nous nuancerons par la suite :

« L'usinage grande vitesse regroupe l'ensemble des usinages par enlèvement de matière mettant en jeu des phénomènes de coupe adiabatique et/ou des vitesses d'avance élevées par rapport à l'usinage conventionnel»

En effet, l'UGV se distingue de l'usinage à vitesse conventionnelle (UVC) par les phénomènes mis en jeu lors de la formation du copeau. Komanduri voit en cette dernière deux phases : la première consiste en un aplatissement de la matière à l'avant de l'outil suivie par la seconde phase de cisaillement catastrophique du copeau [Komanduri & all 82]. Ces phénomènes apparaissent à partir d'une certaine vitesse qui dépend du matériau usiné. Pour citer un exemple, dans un acier 35 NiCrMo6 de dureté 325 HB, ce phénomène apparaît à partir d'une vitesse de coupe avoisinant les 275m/min. Le tableau suivant nous donne des ordres de grandeur des vitesses de coupe mises en jeu en UGV.

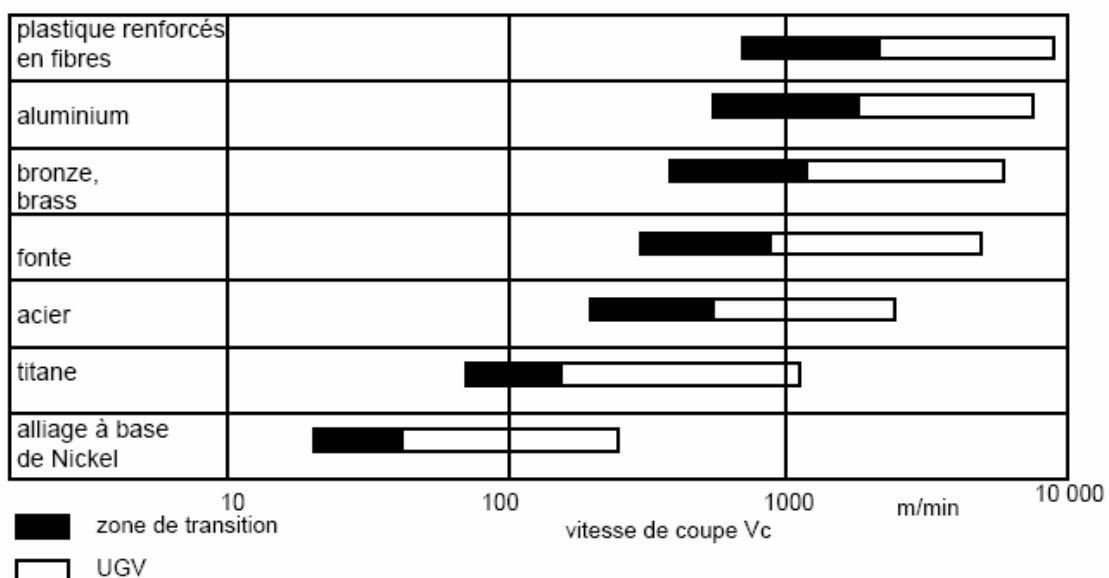


Fig. 24 : Vitesse de coupe pour différents matériaux [Tönshoff & all 98]

Nous pouvons remarquer que le passage de l'UGC à l'UGV n'est pas net et qu'une zone de transition assure la continuité des transformations.

Il arrive cependant que, dans le cas de l'usinage avec des fraises de petit diamètre par exemple, les vitesses de coupe n'atteignent pas les valeurs considérées de l'UGV malgré des fréquences de rotation de broches élevées, de l'ordre de 20 000 tours/min. Selon les points de vue, ce type d'usinage appartient ou non au domaine de l'usinage grande vitesse. En général, les vitesses d'avance élevées qui en découlent permettent de les différencier d'un usinage conventionnel et des usinages à fortes avances sont classées dans le domaine de l'UGV.

2.1.2. Avantages et inconvénients de l'UGV

Ce nouveau mode d'usinage dont la démocratisation a commencé au début des années 90, a été rendu possible par le développement des technologies machine. En particulier, l'amélioration des propriétés dynamique a été capitale. En UGV, la vitesse de coupe ainsi que l'avance sont multipliées par 5 à 10. Ceci a des conséquences très avantageuses sur la coupe et l'usinage :

- Un gain de temps très conséquent dans les phases de finition et en particulier lors de l'usinage de formes complexes qui nécessitent un grand nombre de passes.
- Diminution des efforts de coupe :
 - liée à l'utilisation d'une faible avance par dent et donc d'une section équivalente du copeau plus faible.
 - liée à une légère diminution (jusqu'à 20%) de la pression spécifique de coupe grâce aux phénomènes de coupe adiabatiques mis en jeu.Ceci permet entre autre la réalisation de voiles minces de quelques dixièmes de mm par exemple dans l'aluminium.
- Affectation thermique faible de la pièce et de l'outil : la dissipation de la chaleur se fait presque exclusivement dans le copeau.
- Gain de temps lors des trajets hors matière.
- Amélioration des surfaces finies qui permettent souvent de se passer de super finition (polissage, ébavurage, rectification...).
- A puissance de broche maxi, en ébauche, le débit copeau est amélioré dans la mesure où le coefficient de pression spécifique de coupe est plus faible.

Ces avancées dans le domaine de l'usinage entraînent cependant l'apparition de phénomènes liés à la dynamique et qui peuvent parfois s'avérer être des inconvénients :

- La présence de vibrations à certaines fréquences peut s'avérer problématique dans certains cas. La connaissance des lobes de stabilité permet de résoudre en partie ces problèmes (certains types d'usinages utilisent même les vibrations à leur avantage comme le perçage vibratoire).
- Bien que les accélérations machine soient importantes, les inerties de pièces en mouvement rapide créent parfois des problèmes de suivi de trajectoires et de respect des conditions de coupe. L'exemple (Fig. 25) montre les problèmes de suivi de la consigne d'avance dans le cas d'une pièce de forme complexe présentant des discontinuités d'ordre différents [Dugas, Lee, Hascoët 01] :

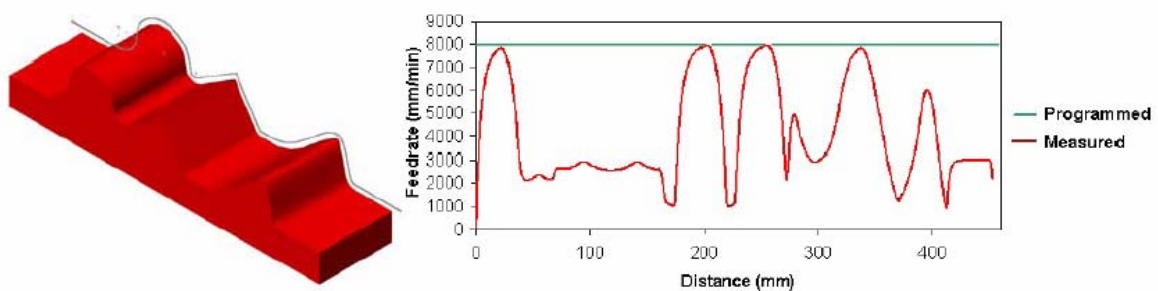


Fig. 25 : Comparaison entre l'avance programmée et mesurée [Dugas, Lee, Hascoët 01]

- Nous pouvons citer comme autre inconvénient de l'UGV l'utilisation de machines chères à l'achat et à l'entretien. C'est également le cas de la plupart des autres éléments de la cellule élémentaire d'usinage UGV, comme par exemple le porte outil qui nécessite un équilibrage statique et dynamique.

Ainsi beaucoup de publications ont été réalisées à ce jour pour mettre en avant les travaux réalisés concernant l'UGV. Ainsi, comme nous l'avons vu précédemment, l'UGV se distingue de l'UGC par la mise en œuvre de machines aux propriétés dynamiques plus importantes. Ces propriétés changent le comportement de la pièce, de l'outil et de la machine lors de l'usinage. Il est donc nécessaire, pour optimiser l'usinage, d'intégrer ces spécificités dans la génération des trajectoires, des paramètres d'usinage et dans le suivi et l'anticipation du comportement de la cellule d'usinage.

2.2. Optimisation du processus d'usinage

Le point de vue que nous allons adopter par la suite concernant l'optimisation du processus d'usinage est celui de la programmation et de la génération de trajectoires. D'autres points de vue comme par exemple la surveillance d'usinage permettent l'optimisation du processus d'usinage mais ne correspondent pas, à proprement dit, à nos thèmes de recherche privilégiés. Les principaux travaux présentés sont généralement ceux de l'IRCCyN.

2.2.1. Généralités

Outre les avancées technologiques réalisées au niveau de l'ensemble des éléments constituant la cellule élémentaire d'usinage, des progrès considérables dans le domaine des commandes numériques ont été réalisés.

Les systèmes actuels de commande des machines ne prennent en compte que les données de position et cinématiques pour décrire les trajectoires souhaitées. D'autres paramètres influencent cependant la trajectoire, comme les efforts de coupe. Les nouvelles CN permettent d'acquérir des données en temps réel comme les couples et puissances délivrées par les moteurs au niveau des axes et de la broche. Ces données, bien que disponibles, sont encore très peu exploitées.

Bien des travaux ont été réalisés pour développer des simulateurs permettant de prendre en compte les phénomènes de flexion d'outil, de vibration, etc.

Nous pouvons citer par exemple le simulateur CUTPRO développé par le laboratoire MAL (Manufacturing Automation Laboratory) de l'université de Colombie britannique au Canada [Altintas 00]. Celui-ci est relativement complet et permet de simuler plusieurs types de phénomènes comme par exemple la réponse dynamique de l'ensemble outil/ broche / porte outil ou l'analyse modale d'un axe machine par l'intermédiaire de différentes 'boîtes à outils' [Malinc].

Un autre simulateur, BALLMILL, a été développé en Grèce au Laboratory for Machine Tool and Manufacturing Engineering [Bouzaki & all 03]. Il est restreint au cas des fraises boules mais permet de déterminer les conditions de coupe optimum pour respecter un critère d'état de surface pour un temps d'usinage minimum.

Les simulateurs précédents permettent, entre autres, de déterminer les efforts de coupe mais ne prennent pas en compte la vitesse d'avance réelle qui diffère souvent, comme nous l'avons vu, de la vitesse d'avance programmée.

2.2.2. Le simulateur Ircsyn

Ce simulateur a été développé dans le cadre du projet « Saut technologique UGV » [Hascöet & all 01a] et de la thèse d'Arnaud Dugas [Dugas 02a] à l'IRCCyN (Institut de Recherche en Communication et Cybernétique de Nantes).

Le principe général de fonctionnement est schématisé sur la figure suivante :

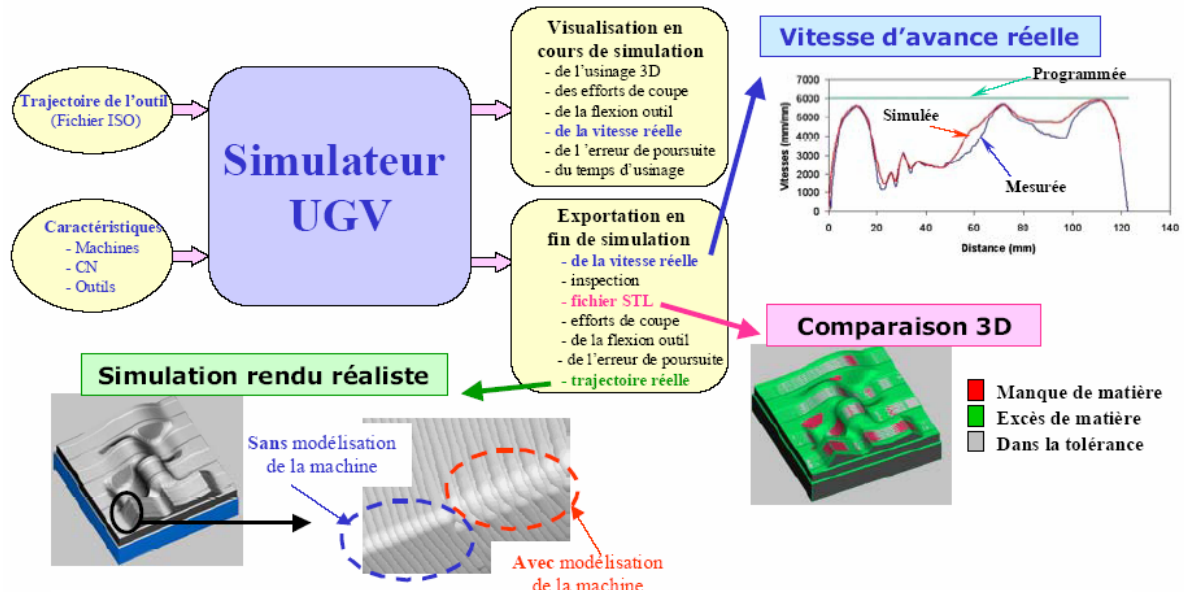


Fig. 26 : Schématisation du fonctionnement du simulateur IRCCyN [Dugas 02d]

Ce simulateur permet de visualiser, à chaque instant de la simulation, le déroulement de l'usinage, les efforts de coupe et les répercussions sur la flexion de l'outil, la vitesse d'avance réelle ainsi que l'erreur de poursuite et le temps d'usinage. Ainsi, ce qui le différencie des simulateurs cités précédemment, est le fait qu'il puisse simuler et tenir compte de la vitesse d'avance réelle, ce qui est primordial pour le calcul de l'engagement radial et tout ce qui en découle (erreur de poursuite, efforts de coupe, etc.) [Dugas 02a] [Dugas 02b].

Il est ensuite possible d'exporter la pièce en fin de simulation pour pouvoir en faire une comparaison 3D avec le modèle souhaité, et ainsi déterminer les zones d'excès et de manque de matière. Il est aussi possible de réaliser une simulation « rendu réaliste » qui permet de comparer la simulation avec ou sans modélisation dynamique de la machine et évaluation de l'erreur dynamique. Concernant la vitesse d'avance réelle, qui diffère de la vitesse d'avance programmée, l'exemple ci-dessus montre que la simulation donne une image proche de la vitesse d'avance mesurée lors de l'usinage. Cette vitesse d'avance simulée est calculée en utilisant un look ahead statique (seules les informations des blocs suivants sont nécessaires) et un look ahead dynamique (informations des blocs précédents et suivants nécessaires). Le profil réel de la machine peut donc être déterminé en connaissant les limites de la machine et du contrôleur.

Des travaux complémentaires ont permis d'étendre son utilisation aux machines à structure parallèle [Terrier 05]. Ceci a nécessité un travail d'intégration des modèles géométriques des

machines parallèles. A ce jour, en plus de l'ensemble des machines sérielles, le simulateur est disponible pour les machines parallèles VERNE et SPACE développées par Fatronik.

2.2.3. Une nouvelle approche utilisant les données CN

Nous avons vu précédemment que les nouvelles commandes numériques permettent d'acquérir des données lors de l'usinage et en particulier les couples et la puissance utilisés sur les axes lors de l'usinage d'une pièce. Il est ainsi possible, en collectant les informations disponibles, de pouvoir déterminer les efforts de coupe et ainsi modifier la trajectoire et les paramètres de coupe pour tenir compte de ces perturbations [Hascoët, Rauch 06c]. Le but de cette étude est de pouvoir, à terme, intégrer dans la CN ces données temps réel pour optimiser les conditions de coupe en tenant compte des efforts de coupe (Fig. 27):

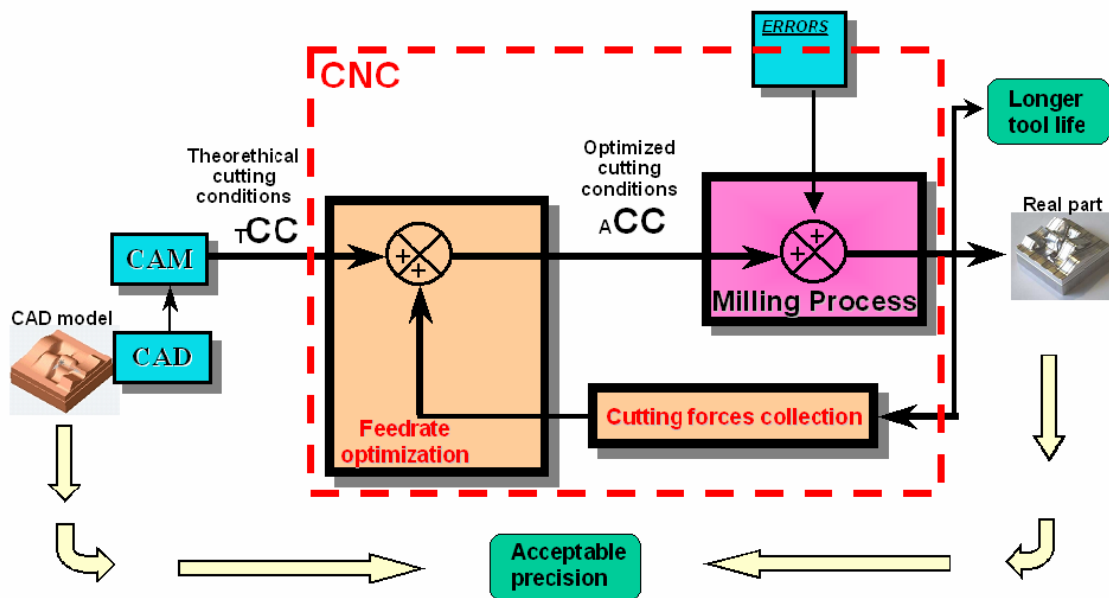


Fig. 27 : Schéma de la méthode proposée de la CAO à la pièce réelle [Hascoët, Rauch 06]

Beaucoup de méthodes se basent, pour la détermination des efforts de coupe, sur des composants rapportés au sein de la cellule élémentaire d'usinage comme par exemple des tables dynamométriques 3D. Celles-ci permettent de relever une image précise des efforts de coupe mais ont plusieurs inconvénients : elles représentent tout d'abord un investissement non négligeable. De plus, elles modifient directement le comportement de la cellule d'usinage au niveau vibratoire mais aussi en terme d'encombrement. Les données récoltées doivent être traitées sur un PC séparé et tous les modes de lubrification ne sont pas tolérés. La méthode proposée à l'IRCCyN permet de s'affranchir de ce genre de matériel en traitant directement les mesures temps réel fournies à travers la CN.

L'acquisition et le calcul des efforts de coupe se fait en 4 temps :

- Le premier temps consiste à acquérir les couples sur les axes en exécutant l'usinage sans la pièce. Cela permet de prendre en compte la dynamique et les inerties liées à la machine.
- Le second temps consiste à acquérir les couples sur les axes lors de l'usinage de la pièce.
- On soustrait les résultats de la première étape à la seconde : cela permet de recueillir les couples nécessaires pour vaincre exclusivement les efforts de coupe.
- La dernière étape consiste à traiter les données obtenues pour en déduire directement les efforts de coupe.

Des mesures ont été effectuées pour valider expérimentalement cette nouvelle méthode d'acquisition des efforts de coupe. Des tests ont été menés sur une pièce test en utilisant la machine parallèle VERNE :

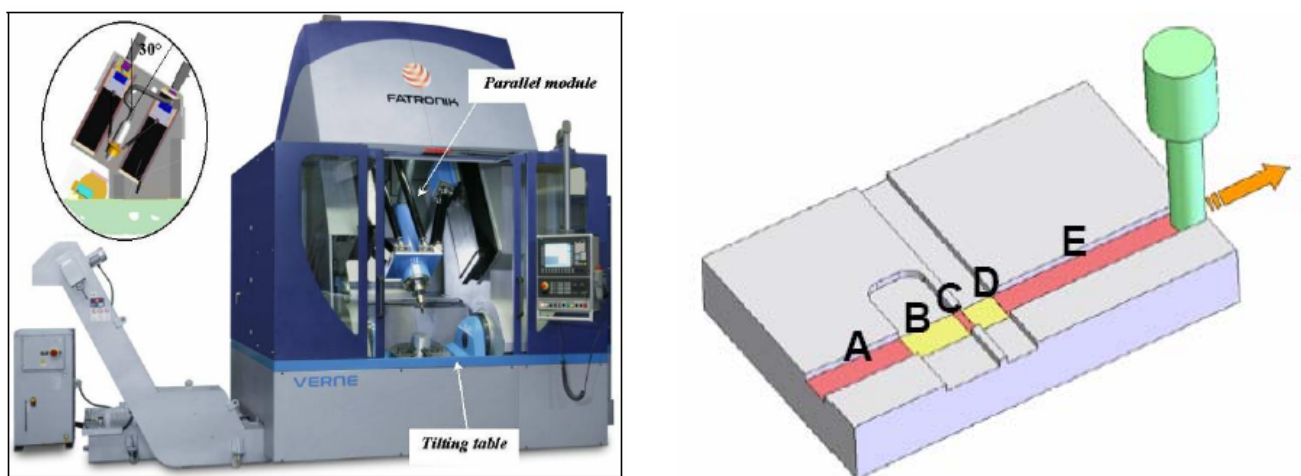


Fig. 28 : Machine parallèle VERNE et pièce Test [Hascoet, Rauch 06a]

Les résultats pour cette pièce test ont été comparés à ceux obtenus par l'emploi d'une table dynamométrique :

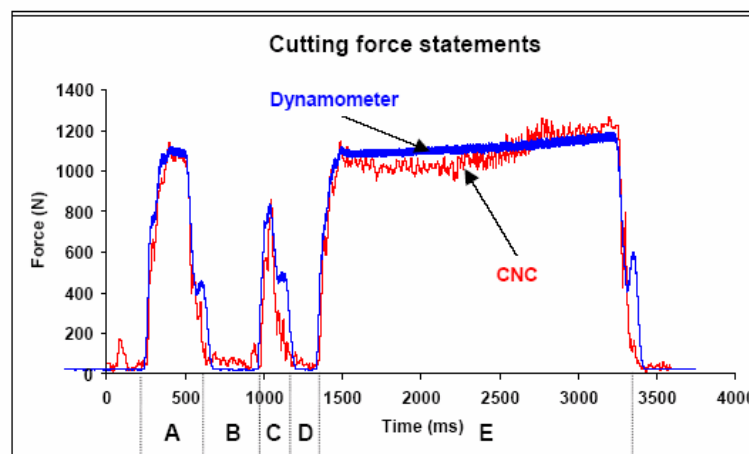


Fig. 29 : Comparaison des efforts de coupe acquis par la CN et par la table dynamométrique

Les résultats montrent que cette méthode est prometteuse dans la mesure où elle ne nécessite pas d'investissement en matériel de mesure, où elle est simple à mettre en œuvre et où les résultats sont proches de ceux trouvés avec une table dynamométrique. Cela ne permet

cependant pas de remplacer la table dynamométrique dans le cas d'applications où la précision demandée est grande, mais pourrait convenir dans une optique de retour des grandeurs des efforts au niveau de la commande numérique.

2.3. Trajectoires en tréflage et usinage Trochoïdal

Nous avons vu précédemment que le fraisage de finition nécessitait une optimisation des trajectoires pour que la surface usinée finie soit conforme aux tolérances souhaitées. Dans le cas du fraisage d'ébauche, les critères d'optimisation sont différents. En particulier dans le cas du vidage de poche, des critères comme le débit copeau maximal sont souvent retenus. Beaucoup de travaux de recherche ont été effectués dans ce domaine mais il s'avère qu'en dépit des progrès considérables des technologies mises en œuvre, les logiciels de FAO ne proposent à ce jour que des stratégies de vidage de poche dérivés des stratégies standards et éprouvées comme le zig-zag ou le contour parallèle.

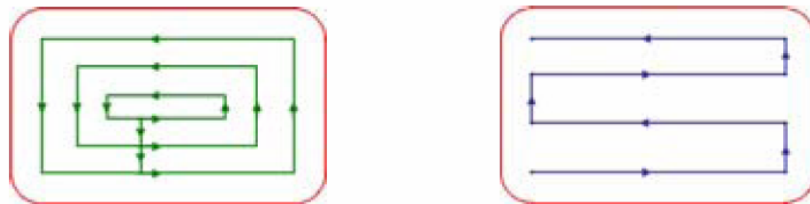


Fig. 30 : Stratégies 'contour parallèle' et 'zig-zag'

L'usinage trochoïdal et le tréflage sont des techniques récentes et innovantes en terme d'ébauche et ont fait leurs preuves dans le cas du vidage de poche dans des matériaux durs. Dans le cas de matériaux comme l'aluminium, ils peuvent aussi présenter des avantages [Rauch Hascoët 06].

Les caractéristiques de l'usinage trochoïdal sont les suivantes :

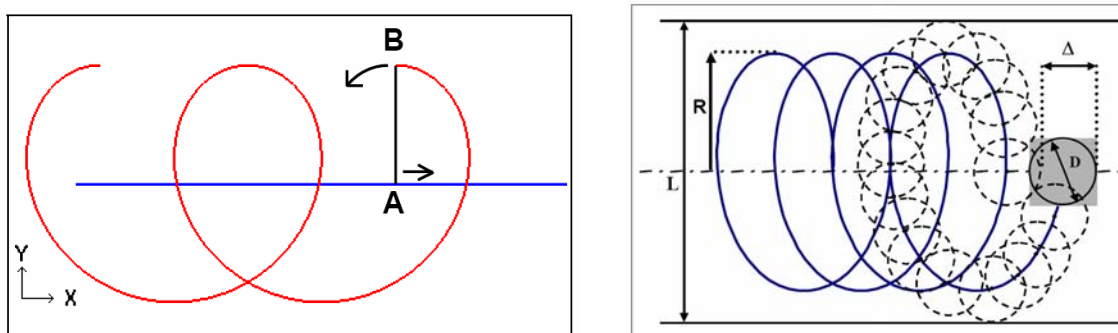


Fig. 31 : Mouvement trochoïdal et trajectoire trochoïdale [Rauch Hascoët 06b]

Le principal intérêt de l'usinage trochoïdal est de présenter une avance uniforme durant l'usinage ainsi qu'une épaisseur de copeau constante. Cela entraîne directement des répercussions favorables sur les conditions de coupe et en particulier cela permet de diminuer les chocs et le marquage de la pièce. Les efforts de coupe et le temps de contact outil – pièce sont plus faibles que dans des stratégies classiques, ce qui peut permettre d'augmenter la durée de vie de l'outil mais aussi d'usiner dans des matériaux plus durs avec des productivités multipliées par 3.

Les caractéristiques du tréflage sont les suivantes :

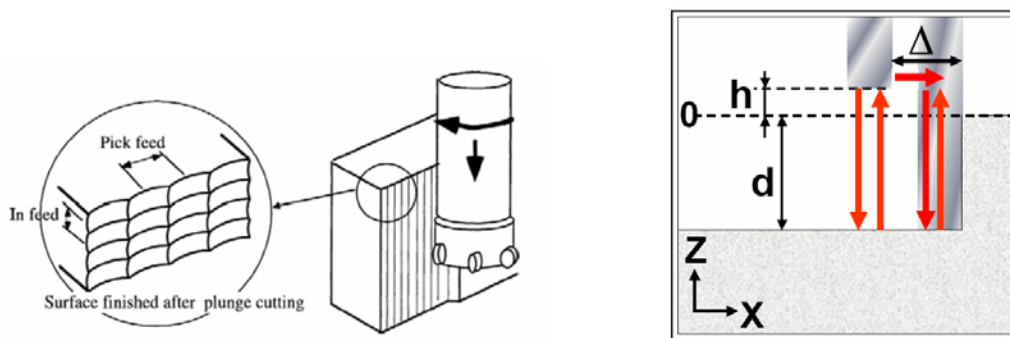


Fig. 32 : Principe et paramètres influents en tréflage

Le principal avantage du tréflage est que l'outil travaille suivant son axe, c'est-à-dire suivant sa configuration la plus rigide. Cela permet d'éviter la flexion d'outil ainsi que le broutement.

Pour chacune des trajectoires en tréflage et usinage trochoïdal il a été proposé une construction basée sur la décomposition en un motif plus une courbe guide. Une trajectoire élémentaire est alors appliquée, permettant ainsi de simplifier considérablement la génération de la trajectoire complète et les modifications qui en découlent.

Ces deux stratégies ont été comparées lors d'un vidage de poche à la stratégie zig-zag. L'analyse des résultats s'est faite en terme de productivité, d'efforts de coupe générés, et de qualité de la poche usinée :

- La stratégie zig-zag reste la meilleure en terme de productivité dans la mesure où les trajets outils utilisés sont les plus courts. L'usinage le plus pénalisé est l'usinage trochoïdal dans la mesure où le trajet effectué est plus de 3 fois plus important que le zig-zag. De plus, les trajets hors matière sont très importants pour le fraisage et l'usinage trochoïdal, ce qui les pénalise en terme de productivité.

- Concernant les efforts de coupe, l'usinage trochoïdal et le tréflage devançant la stratégie zig-zag dans la mesure où les efforts moyen et maximal sont jusqu'à 2 fois moins importants. Ainsi, le premier résultat en faveur du tréflage et de l'usinage trochoïdal concerne les sollicitations moins importantes de l'outil et de la pièce. C'est l'une des raisons pour laquelle ils sont très appréciés dans le cas de l'usinage des matériaux durs.
- La qualité de la poche usinée se relève en deux endroits : le fond et les parois de la poche. Concernant le fond de la poche, l'usinage trochoïdal présente les meilleurs résultats de planéité, suivi par la stratégie zig-zag. Le tréflage présente des erreurs plus importantes dans la mesure où la géométrie de l'outil utilisé laisse en fond de poche de la matière. Concernant les parois verticales, la stratégie zig-zag présente l'inconvénient de laisser une crête importante entre deux passes d'usinage, ce qui entraîne souvent la nécessité d'une passe de demi-finition. Le tréflage semble être le meilleur choix pour ce critère dans la mesure où il est possible de choisir le pas entre deux plongées, influençant ainsi directement la hauteur de crête sur la paroi.

Cet article met aussi en avant le fait que la dynamique de la machine est très importante. Il y a peu d'avantages à utiliser les stratégies tréflage et usinage trochoïdal dans le cas de machines dont les performances d'accélération sont inférieure à $10 m.s^{-2}$.

2.4. Conclusion

Nous avons vu l'avancée des travaux en terme d'optimisation des trajectoires et en particulier par l'utilisation de stratégies encore peu pratiquées et proposées par les logiciels de CAO comme le tréflage ou l'usinage trochoïdal. D'autres travaux ont permis la réalisation d'un simulateur permettant de prendre en compte la vitesse d'avance réelle ainsi que les efforts de coupe. Ces derniers peuvent aussi être appréciés directement par le relevé en temps réel des couples fournis au niveau des axes pour, à terme, pouvoir être pris en compte directement en temps réel par la CN. Tous ces travaux peuvent être reliés à certaines problématiques liées au Step_NC. Une adaptation de ceux-ci dans le but d'implémenter les bases de données Step_NC et d'enrichir de ces méthodes les possibilités qu'il offre en terme de « feed-back » au niveau de la commande numérique est à envisager.

3. Partie stage de recherche

Nous avons vu dans la partie bibliographie l'étendue actuelle des travaux sur le format Step_NC ainsi que les apports qu'il est en mesure de permettre en terme de programmation. C'est une nouvelle opportunité dans l'optique de la programmation « intelligente » des machines à commande numérique. De plus, il s'inscrit parfaitement dans les travaux que nous menons en terme de développement de stratégies innovantes comme l'usinage trochoïdal ou le tréflage. Ces stratégies, souvent méconnues et peu proposées dans les logiciels de FAO, présentent un intérêt qui n'est plus à démontrer dans le domaine du vidage de poche en ébauche. Elles ne sont cependant pas encore implémentées et normalisées dans le format Step_NC.

La problématique de recherche sur laquelle nous avons travaillé est donc l'implémentation de l'usinage trochoïdal et du tréflage dans la chaîne numérique Step_NC. Pour ceci, nous proposons une méthode de définition de ces stratégies d'usinage qui nous a menée à la mise en place des paramètres nécessaires à la construction des trajectoires d'usinage. Nous avons réalisé un ensemble d'outils pour pouvoir implémenter le format Step_NC de ces stratégies.

Le but de ces travaux à terme, est de montrer la faisabilité et l'intérêt de la génération de ces stratégies en Step_NC. Un ensemble de stratégies courantes a déjà été développé et normalisé. Nous nous appuierons, dans un premier temps, sur les stratégies déjà existantes pour pouvoir proposer une méthode d'implémentation pour l'usinage trochoïdal et le tréflage.

Nous serons aussi amenés à travailler sur le prototype de chaîne numérique Step_NC développé par le laboratoire Coréen de l'Université de Postech. Nous commencerons donc nos travaux par une étude détaillée de l'architecture et du fonctionnement de ce prototype.

Nous serons ensuite en mesure de détailler les différentes phases de l'implémentation des stratégies à motifs. Ces phases sont au nombre de trois : la génération du fichier Step NC, la construction des trajectoires et l'usinage de pièces tests.

Avant cela, commençons par revenir sur les stratégies déjà en place dans la norme, ainsi que sur le prototype de chaîne numérique coréen.

3.1. Les normes AP238 et 14649

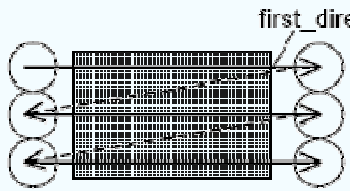
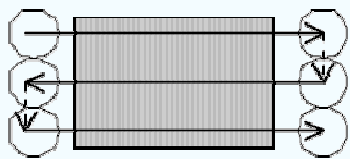
Pour pouvoir mettre en place les concepts que nous souhaitons développer en termes d'usinage trochoïdal et de tréflage, il est nécessaire de s'intéresser de plus près aux concepts sur lesquels sont basées les normes régissant le Step_NC. La normalisation ISO du Step_NC s'appuie en particulier sur deux normes dont les points de vue diffèrent sensiblement (Norme ISO 14649 et ISO 10303 AP238). Recentrons nous ici sur leur contenu au niveau de l'usinage et en particulier des stratégies d'usinage.

Pour ceci, la norme ISO 14649-Part 11 définit précisément l'ensemble des éléments nécessaires pour le fraisage. Ces éléments sont décrits dans le langage EXPRESS décrit dans la norme ISO 10303-11. L'EXPRESS G permet de visualiser ce langage sous forme graphique.

Dans le cas des stratégies connues en fraisage 2 ½ axes, la norme nous propose la classe *Two5DMillingStrategy*. Celle-ci regroupe l'ensemble des stratégies suivantes :

- Les quatre stratégies d'usinage de base : *Contour Spiral*, *Contour Parallèle*, *Unidirectionnel* et *Bidirectionnel* (zig-zag)
- Deux stratégies « composées » : *Contour-Bidirectionnel* : il s'agit de réaliser un contour de la poche avec une stratégie *Contour parallèle* puis usiner la matière restante en Zig-zag (*Bidirectionnel*). La seconde stratégie *Bidirectionnel-Contour* est l'inverse de la précédente : stratégie zig-zag en premier lieu puis *Contour Parallèle*.
- *Center_Milling* permet l'usinage au centre d'une entité (souvent utilisé dans le cas de rainures)
- *Explicit Strategy* regroupe toutes les stratégies qui ne peuvent pas être décrites par les stratégies précédentes. Elle nécessite alors l'entière définition de tous les mouvements.

Le tableau suivant récapitule les stratégies proposées par la norme ISO 14649-Part 11:

Stratégie	Schéma	Attributs
Unidirectionnel	 <p>Le schéma illustre l'usinage unidirectionnel sur une surface rectangulaire. Des flèches indiquent des passes parallèles effectuées dans une seule direction, de gauche à droite. Une étiquette 'first_direction' pointe vers la première passe.</p>	<ul style="list-style-type: none"> - Feed_direction - Cutmode
Bidirectionnel (zig-zag)	 <p>Le schéma illustre l'usinage bidirectionnel (zig-zag) sur une surface rectangulaire. Des flèches indiquent des passes alternées dans les deux directions (de gauche à droite, puis de droite à gauche).</p>	<ul style="list-style-type: none"> - Feed_direction - Stepover_direction - Stroke_connection_strategy

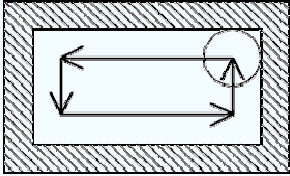
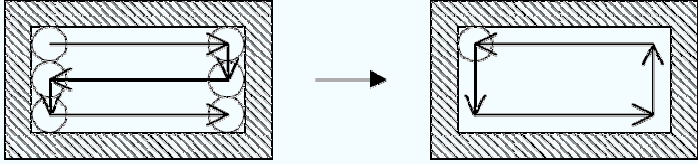
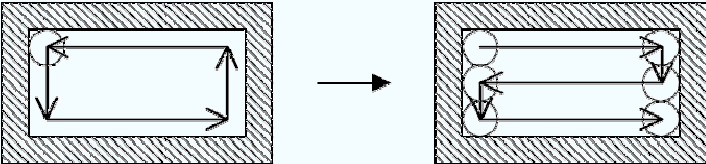
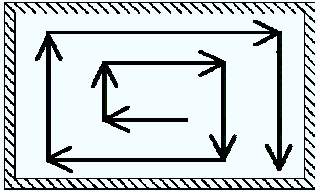
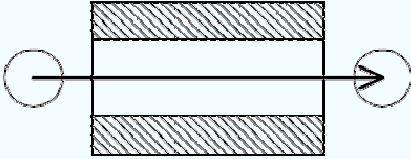
Contour Parallèle		-
Bidirectional-Contour		- Feed_direction - Stepover_direction - Rotation_direction - Cutmode
Contour-Bidirectional		- Feed_direction - Stepover_direction - Rotation_direction - Cutmode
Contour Spiral		- Rotation_direction - Cutmode
Center Milling		-
Explicit	Toutes les stratégies non décrites précédemment.	-

Tableau 1 : Stratégies mises en place dans la norme ISO 14649 Part 11

Les attributs associés à chaque stratégie permettent de les définir. Suivant le type de stratégie, nous retrouvons les attributs suivants :

- *Cutmode* : spécifie si l'usinage se déroule en avalant ou en opposition. Par défaut, l'usinage en avalant est sélectionné.
- *Feed_direction* : c'est la direction d'avance exprimée par un vecteur dans le repère lié à la pièce. L'attribut *Cutmode* est prioritaire en cas de conflit.
- *Stepover_direction* : direction de "propagation" de la stratégie zig-zag possible sur la droite ou sur la gauche.
- *Stroke_connection_strategy* : permet de choisir la stratégie à adopter lors des phases de connection entre les différents trajets : ligne droite, montée-décalage-descente, boucle, descente.
- *Rotation_direction* : sens de rotation en particulier dans le cas de spirales : horaire ou anti-horaire. Par défaut, le sens anti-horaire (trigonométrique) est sélectionné.

L'attribut *Cutmode* est prioritaire en cas de conflit.

Pour synthétiser tout cela, le graphique Express G de la fonction *Two5DMillingStrategy* (Fig. 33) permet de regrouper l'ensemble de ces informations et d'en comprendre l'articulation :

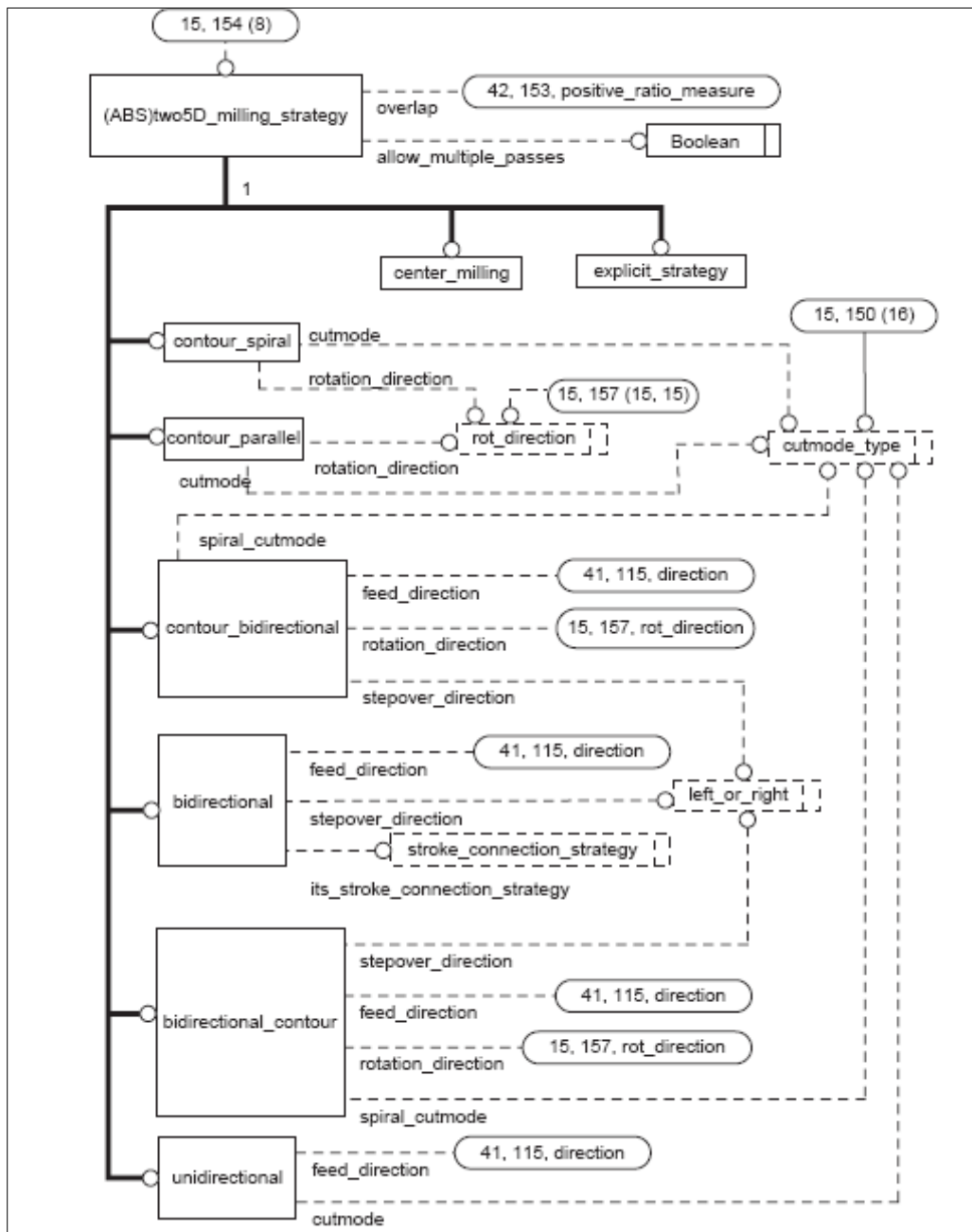


Fig. 33 : Diagramme ARM de la fonction *Two5DMillingStrategy* [ISO 10303-238 Fig. G15]

Ainsi, nous pouvons remarquer qu'il existe deux autres attributs communs à toutes les stratégies :

- Overlap : représente le recouvrement entre deux passes consécutives. Il est exprimé en pourcentage du diamètre de l'outil.

- Allow_multiple_passes : par défaut, sa valeur est « vraie » comme dans le cas de tous les usinages d'ébauche communs nécessitant d'usiner la matière en plusieurs passes suivant l'engagement axial maximum. Il arrive dans des cas spéciaux que l'on veuille forcer l'usinage en une passe en sélectionnant « faux ».

Nous verrons par la suite que la construction des trajectoires d'usinage trochoïdal et de tréflage s'appuie au préalable sur la construction de courbes guides. Nous serons alors en mesure de nous appuyer sur ces stratégies déjà définies dans la norme pour construire ces courbes guides.

3.2. La chaîne numérique coréenne : PosSFP – TPG – TPV

Les normes sur le Step_NC présentent donc certaines lacunes au niveau du choix de stratégies de vidage de poche comme le tréflage ou l'usinage trochoïdal. Nous proposons ici de mettre en place l'ensemble des bases permettant la définition des trajectoires, l'implémentation dans le fichier Step_NC et la génération de ces trajectoires pour la machine.

L'objectif est donc de réaliser l'ensemble de la chaîne numérique Step_NC permettant de mettre en œuvre le vidage d'une poche en usinage trochoïdal et tréflage, de la maquette numérique jusqu'à la pièce usinée.

Pour ceci, et dans le cadre de notre collaboration avec le laboratoire de l'université coréenne de Postech, nous nous sommes en partie basés sur le prototype de chaîne numérique Step_NC coréenne: PosSFP – TPG – TPV. Celle-ci est constituée d'un ensemble d'outils permettant de traiter une pièce de la CAO jusqu'à l'usinage :



Fig. 34 : Prototype coréen de chaîne numérique Step_NC

Ce prototype complet s'appuie sur des applications qui permettent, à partir d'un fichier de géométrie CAO d'une pièce, de générer le programme Step_NC, puis de générer et visualiser les trajectoires outil. Une interface homme machine permet de visualiser le déroulement du

programme et le contrôle des axes. Le contrôle de la machine se fait alors par l'intermédiaire d'un dernier module.

L'ensemble des modules développés dans les laboratoires de l'Université de Postech comporte le préfixe « Pos » en référence à celle-ci. L'architecture fonctionnelle est basée sur 5 modules principaux (Fig. 35) :

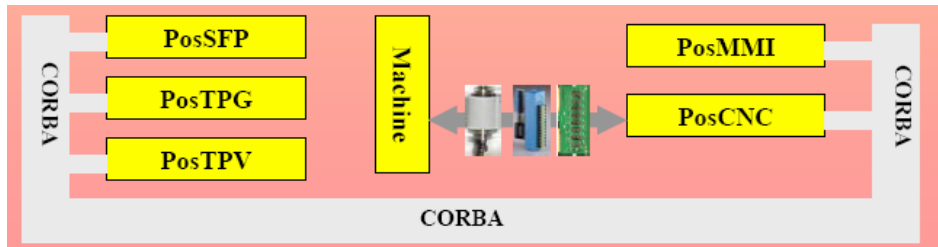


Fig. 35 : Architecture basée sur 5 modules

3.2.1. PosMMI (Men Machine Interface)

C'est l'interface homme/système Step_NC. C'est à partir de celui-ci que l'opérateur charge les programmes Step_NC, effectue les réglages de la machine outil, exécute les opérations sur la machine ainsi que la surveillance lors de l'usinage. Les principaux avantages et éléments différenciateurs qui la distinguent d'une interface homme machine habituelle sont les suivants :

- Elle comporte une interface Internet pour avoir accès aux bases de données et en particulier à celles des fichiers Step_NC
- Elle est fondée sur la technologie Java, ce qui lui confère une opérabilité étendue aux différents environnements d'atelier et indépendamment des parties opératives.
- Elle centralise la communication avec les autres modules (comme PosSFP, PosTPG, PosTPV et PosCNC) par l'intermédiaire d'un protocole de communication CORBA. Elle fournit à l'opérateur des possibilités de configuration pour régler les paramètres de connexion entre les différents modules.

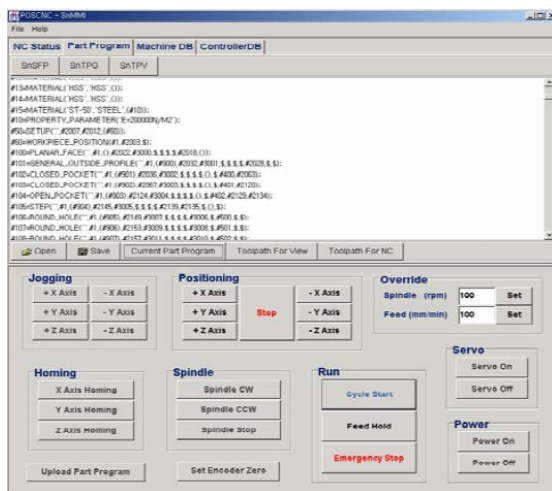


Fig. 37 : Visualisation du programme Step_NC dans le viewer PosMMI

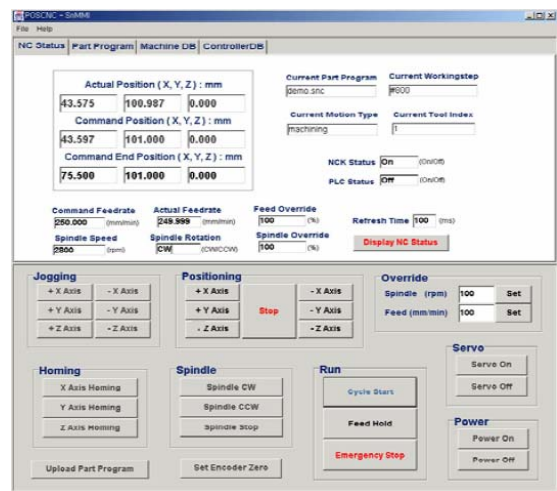


Fig. 36 : Contrôle des axes et affichage de l'état de l'usinage

3.2.2. PosSFP (Shop-floor Programing).

Le module PosSFP permet de créer le fichier ISO 14649 (Step_NC) d'une pièce à partir des informations de géométrie d'un fichier Step AP203 généré par n'importe quel logiciel de CAO.

La première étape consiste en la lecture de la géométrie du fichier Step. Dans le cadre du développement de ce prototype, seuls les fichiers au format Parasolide de Solidworks sont pour l'instant lisibles. Cependant, à terme, tout type de fichier de géométrie Step pourra être lu et traité.

Un système de reconnaissance de forme développé par l'équipe de recherche coréenne prend alors le relais pour reconnaître automatiquement les entités géométriques.

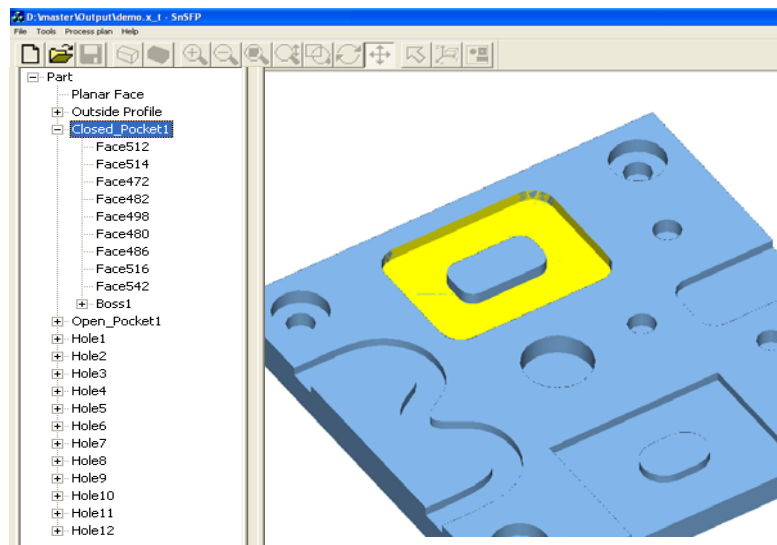


Fig. 38 : Reconnaissance des entités géométriques

Celles-ci sont alors vues en termes d'entités d'usinage définies dans la norme ISO 14649 Part 10. Pour chacune de ces entités d'usinage, l'utilisateur est invité à choisir les caractéristiques de la gamme de fabrication en tant que succession de phases d'usinage associées aux entités (Fig. 39).

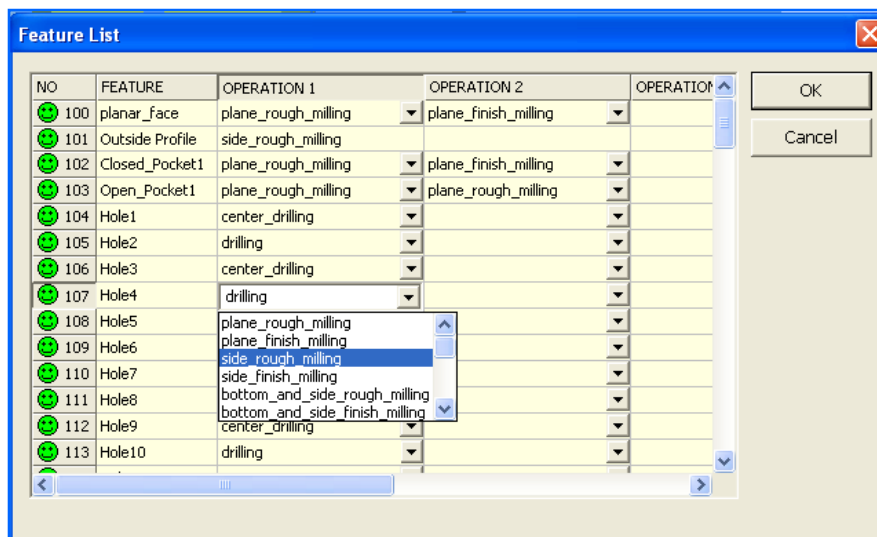


Fig. 39 : Sélection des opérations d'usinage associées aux entités

Pour chaque entité, il est possible d'associer plusieurs opérations (comme par exemple une opération d'ébauche puis de finition). Pour chacune de ces opérations, l'utilisateur sélectionne un outil (Fig. 40).

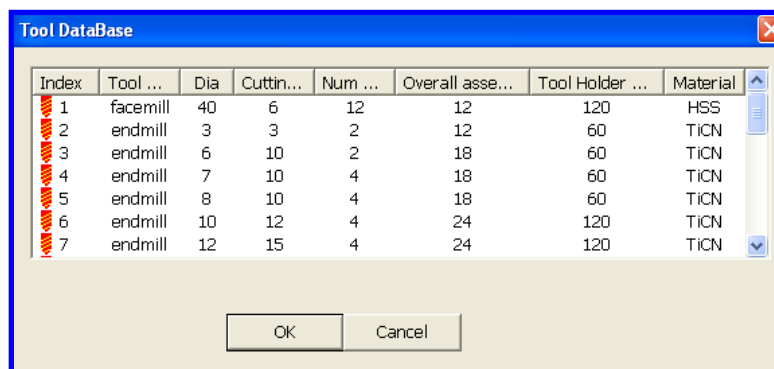


Fig. 40 : Sélection de l'outil dans la base de données

Il est également invité à sélectionner les paramètres et stratégies d'usinage. L'ensemble de ces paramètres et stratégies d'usinage sont regroupés dans la norme ISO 14649 Part 11. On y retrouve les attributs des opérations comme les stratégies d'approche et de retrait matière, des fonctions machine (lubrification, etc.), les conditions de coupe, et en particulier les stratégies d'usinage. Les stratégies actuellement proposées sont (Fig. 41): unidirectionnel, bidirectionnel (ou zig-zag), contour parallèle, contour spiral, fraisage au centre (souvent utilisé dans le cas de rainurages), et 'explicit' qui regroupe toutes les stratégies qui ne peuvent pas être décrites par celles citées précédemment. Il est de même défini des combinaisons entre différentes stratégies comme *bidirectionnel contour* qui consiste tout d'abord à effectuer un vidage du centre en zig-zag (bidirectionnel) et ensuite à finir les flans de la poche en contour parallèle.

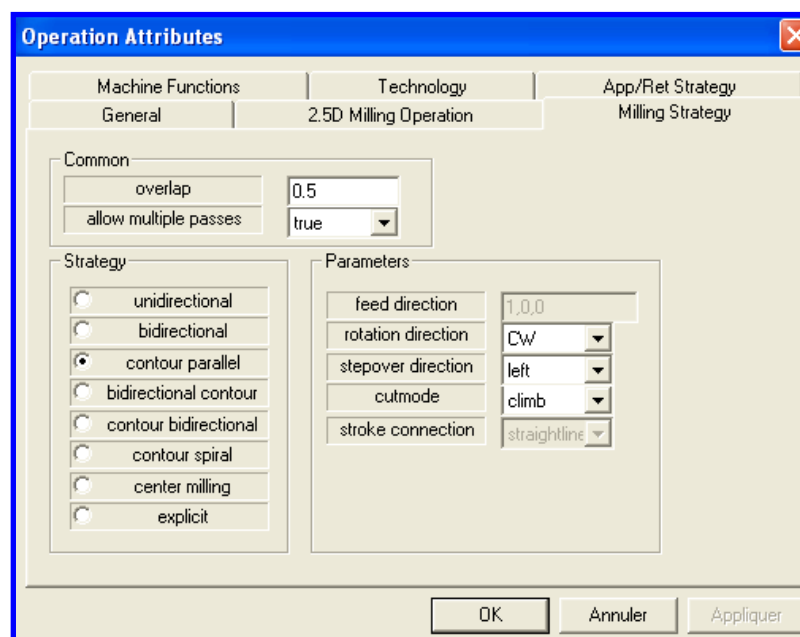
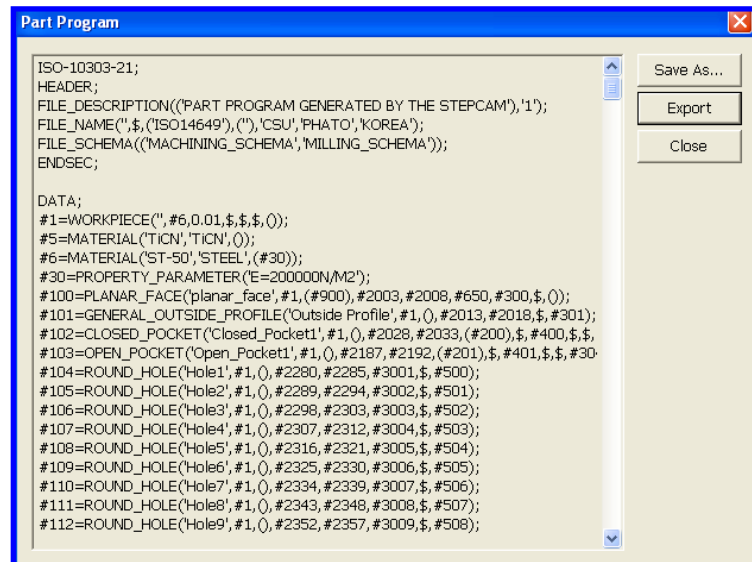


Fig. 41 : Stratégies d'usinage ébauche proposées par PosSFP

Il est alors possible de choisir les caractéristiques associées aux stratégies.

Ces manipulations ont pour but de pouvoir générer le fichier Step_NC ISO 14649. Ce dernier contient l'ensemble des informations nécessaires et suffisantes pour pouvoir, à l'étape suivante, définir les conditions d'usinage et les trajectoires outil.



```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('PART PROGRAM GENERATED BY THE STEPCAM'),'1');
FILE_NAME('','$','(ISO14649)','(',')','CSU','PHATO','KOREA');
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1=WORKPIECE('','#6.0.01,$,$,$,0);
#5=MATERIAL('TICN','TICN',0);
#6=MATERIAL('ST-50','STEEL',(#30));
#30=PROPERTY_PARAMETER('E=2000000N/M2');
#100=PLANAR_FACE('planar_face',#1,(#900),#2003,#2008,#650,#300,$,0);
#101=GENERAL_OUTSIDE_PROFILE('Outside Profile',#1,0,#2013,#2018,$,#301);
#102=CLOSED_POCKET('Closed_Pocket1',#1,0,#2028,#2033,(#200),$,#400,$,$,
#103=OPEN_POCKET('Open_Pocket1',#1,0,#2187,#2192,(#201),$,#401,$,$,#30
#104=ROUND_HOLE('Hole1',#1,0,#2280,#2285,#3001,$,#500);
#105=ROUND_HOLE('Hole2',#1,0,#2289,#2294,#3002,$,#501);
#106=ROUND_HOLE('Hole3',#1,0,#2298,#2303,#3003,$,#502);
#107=ROUND_HOLE('Hole4',#1,0,#2307,#2312,#3004,$,#503);
#108=ROUND_HOLE('Hole5',#1,0,#2316,#2321,#3005,$,#504);
#109=ROUND_HOLE('Hole6',#1,0,#2325,#2330,#3006,$,#505);
#110=ROUND_HOLE('Hole7',#1,0,#2334,#2339,#3007,$,#506);
#111=ROUND_HOLE('Hole8',#1,0,#2343,#2348,#3008,$,#507);
#112=ROUND_HOLE('Hole9',#1,0,#2352,#2357,#3009,$,#508);
```

Fig. 42 : Génération du programme Step_NC

Remarque : PosSFP peut aussi lire en entrée un fichier physique Step_NC ISO 14649. Il est alors possible d'effectuer des modifications sur tous les paramètres de la même façon que celle présentée précédemment. Un nouveau programme Step_NC modifié est alors généré.

PosSFP représente donc un outil complet permettant, à partir de données de géométrie ou d'un fichier Step_NC, de générer le programme ISO 14649 physique. Contrairement au code G, ce programme ne comporte pas d'instructions directes concernant la trajectoire de l'outil.

La première partie du travail que nous avons mené s'intéresse à implémenter les stratégies d'usinage trochoïdale et de tréflage dans ce module PosSFP.

3.2.3. PosTPG (Tool Path Generator) / PosTPV (Tool Path Viewer)

La génération des trajectoires à partir du fichier Step_NC se fait par l'intermédiaire du module PosTPG. Les parcours outil sont donc générés à cette étape au sein de la commande numérique. Le fichier Step_NC est lu par un interpréteur pour construire les trajectoires pour chaque entité. Ces trajectoires sont décomposées en quatre segments : l'approche, l'usinage, le retrait et le départ. PosSFP génère les trajectoires pour chacun d'eux.

La prévention des collisions est aussi un point très important. Dans ce cadre, il existe un plan de sécurité spécifié dans le programme Step_NC. L'outil revient à cette position de sécurité après l'usinage des entités.

L'algorithme de génération des trajectoires inclut des propriétés d'adaptation permettant un usinage autonome et intelligent. Celui-ci permet, par exemple, de reconnaître des situations inattendues comme la non disponibilité d'un outil suite à une casse durant l'usinage d'une entité. L'opération est alors momentanément arrêtée jusqu'au changement d'outil. Par la suite, le programme reprend là où il avait été interrompu.

Le module PosTPV permet de visualiser graphiquement les parcours outil générés, ce qui permet une vérification graphique des trajectoires effectuées (Fig. 43).

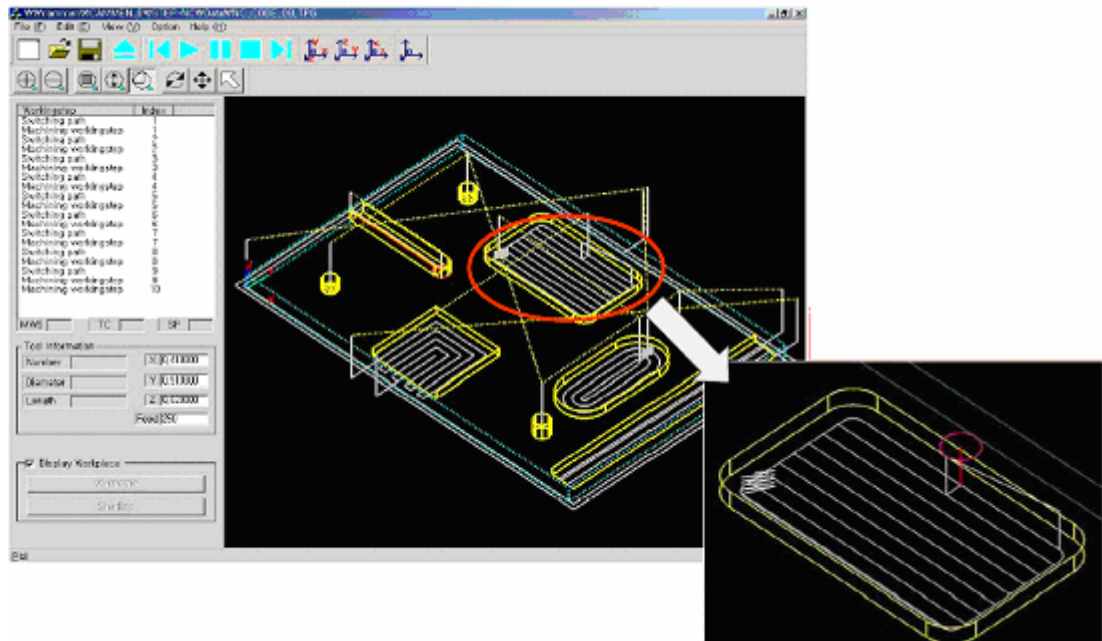


Fig. 43 : Visualisation des trajectoires avec PosTPV

Cette visualisation est alors stockée dans la base de données des trajectoires outil.

3.2.4. PosCNC

PosCNC est le noyau qui sert d'interface avec la machine outil. Il permet l'exécution des mouvements et de l'ensemble des fonctions de la machine outil. Dans ce but, il existe déjà sur le marché des noyaux (ou Kernel) qui auraient pu être utilisés, comme dans le cas du prototype européen faisant appel à la Siemens 840D. Cependant, ce noyau a été développé car les CN existantes ne permettent pas un accès suffisant à leur structure interne, dans le but de tirer l'ensemble des bénéfices de la programmation Step_NC.

L'objectif des travaux que nous avons menés s'inscrit dans le cadre du développement du format Step_NC par l'implémentation de deux stratégies d'usinage bien adaptées au vidage de poche. Des travaux précédents [Rauch Hascoët 06] ont permis de mettre au point une méthodologie de construction des stratégies d'usinage trochoïdal et de tréflage basées sur le principe de courbes à motifs. Ces deux stratégies étant pour l'instant absentes de la normalisation Step_NC, nous proposons ici une méthode d'implémentation et définissons les paramètres et attributs permettant la construction de ces stratégies.

Par la suite, nous nous attacherons à mettre en place l'ensemble de la chaîne numérique Step_NC dans le cas du vidage d'une poche avec une stratégie qui s'appuie sur les courbes à motifs. Cela a pour objectif de montrer la pertinence de la définition des courbes ainsi que la faisabilité de nos propositions. Nous mettrons donc au point un ensemble d'outils permettant de traiter l'ensemble du cheminement de la pièce CAO jusqu'à la pièce réelle en s'appuyant sur le format Step_NC.

Notre démarche se compose de *trois étapes* distinctes :

La *première étape* a pour objectif de concevoir, à partir du modèle CAO, le fichier Step_NC. Il contient l'ensemble des informations concernant la pièce et son usinage. Nous nous attacherons à implanter les paramètres que nous avons définis comme nécessaires aux stratégies à motifs

La *seconde étape* consiste à générer les trajectoires d'usinage à partir des informations lues dans le fichier Step_NC. Cette étape a pour but de montrer la manière que nous proposons pour construire les trajectoires à partir des paramètres que nous avons définis ainsi que leur pertinence.

Nous mettrons en œuvre dans la *troisième étape* l'ensemble des outils mis en place dans les deux premières étapes pour parcourir l'ensemble de la chaîne numérique Step_NC, du modèle CAO jusqu'à l'usinage de 4 pièces test.

Nous allons détailler dans la suite ces étapes.

5. Etape 1 : Génération du fichier Step_NC

La première étape de l'implantation de l'usinage trochoïdal et du tréflage dans l'environnement Step_NC consiste à construire un modèle permettant de communiquer l'ensemble des paramètres nécessaires à la génération des trajectoires à travers un fichier Step_NC. Pour cela, nous nous sommes tout d'abord intéressés à comprendre la structure du fichier Step_NC pour ensuite pouvoir y implanter les paramètres choisis pour l'usinage trochoïdal et le tréflage. Ces deux stratégies ont alors pu être mises en place dans le logiciel coréen PosSFP permettant la génération du fichier Step_NC

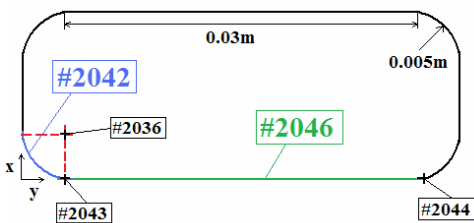
5.1. Structure d'un programme Step_NC et de PosSFP

Afin de pouvoir implanter l'usinage trochoïdal et le tréflage dans le fichier Step_NC, il est nécessaire de bien comprendre la structure d'un fichier Step_NC ainsi que la manière dont il est généré par PosSFP.

Intéressons nous en premier lieu à la structure d'un fichier Step_NC qui contient :

- Une partie d'entête (Header) relative aux informations d'ordre général sur le programme (auteur, date, etc.)
- Des informations de géométrie :

Elles concernent les entités d'usinage reconnues par l'algorithme de reconnaissance de forme. L'écriture dans le programme Step_NC de la géométrie de ces entités se fait grâce à leur décomposition jusqu'à leur plus simple expression, sous forme de pointeurs. Pour schématiser cette décomposition, chaque entité d'usinage 2½ D est traitée comme l'extrusion d'une esquisse 2D. La définition de cette esquisse se fait à partir de polygones décomposés sous forme élémentaires (segments, cercles, etc.). Les coordonnées des pointeurs élémentaires de ces formes sont alors indiquées ligne à ligne dans le programme. Les numéros des lignes entre parenthèses permettent de faire le lien entre les pointeurs et les entités qu'ils décrivent.



```
••#102=CLOSED_POCKET('Closed_Pocket1',#1, #2033,(#200),#400,#302);
```

```
••#302=GENERAL_CLOSED_PROFILE($,#2086);
```

```
#2086=COMPOSITE_CURVE(",(#2042,#2046,#2055,[...],#2085),F.);
```

```
#2036=CARTESIAN_POINT(",(0.005000,0.005000,0.000000));
```

```
#2037=DIRECTION(",(0.000000,0.000000,-1.000000));
```

```
#2038=DIRECTION(",-1.000000,0.000000,0.000000));
```

```
#2039=AXIS2_PLACEMENT_3D(",#2036,#2037,#2038);
```

```
#2040=CIRCLE(",#2039, 0.005000);
```

```
#2041=TRIMMED_CURVE(",#2040,(#2034),(#2035),F.,CARTESIAN.);
```

```
#2042=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,T.,#2041);
```

```
#2043=CARTESIAN_POINT(",(0.005000,0.000000,0.000000));
```

```
#2044=CARTESIAN_POINT(",(0.000000,0.035000,0.000000));
```

```
#2045=POLYLINE(",(#2043,#2044));
```

```
#2046=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,T.,#2045);
```

- Des informations pour l'usinage :

Pour chaque opération sur une entité d'usinage, le fichier Step_NC spécifie l'ensemble des informations suivantes :

- Le type d'opération (comme par exemple surfacage ébauche, finition, perçage, etc.). De la même manière que pour la définition de la géométrie, des liens entre parenthèses renvoient aux lignes du programme, ce qui donne l'opportunité de définir plus précisément des éléments et paramètres sélectionnés pour réaliser l'opération, et en particulier :
- Les outils ainsi que leurs caractéristiques (type d'outil, dimensions, etc.)
- Les paramètres de coupe (vitesse de coupe, d'avance, sens de rotation de la broche, etc.)
- Les paramètres liés à la machine (lubrification, etc.)
- Les stratégies d'usinage (unidirectionnel, zig-zag, contour parallèle...et bientôt usinage trochoïdal et tréflage). Il est important d'analyser ce point spécifique car c'est lui qui supportera la mise en place de l'usinage trochoïdal et le tréflage dans le fichier Step_NC. Pour les stratégies d'usinage qui sont déjà en place dans la norme, les paramètres permettant la construction des parcours outils sont donnés entre parenthèse dans la ligne du programme correspondant. Prenons l'exemple de la stratégie bidirectionnel (ou zig-zag). Les attributs liés à cette stratégie sont (Tableau 1) : *feed_direction*, *stepover_direction*, *stroke_connection* ainsi que les attributs *overlap* et *allow_multiple_passes* qui sont communs à toutes les stratégies. Le choix de ces paramètres dans le logiciel PosSFP se fait par l'intermédiaire de l'onglet « Milling Strategy » qui correspond directement aux choix proposés par la norme pour la fonction *Two5DMillingStrategy*. La ligne correspondante du programme Step_NC reprend ces paramètres sous la forme suivante (Fig. 44) :

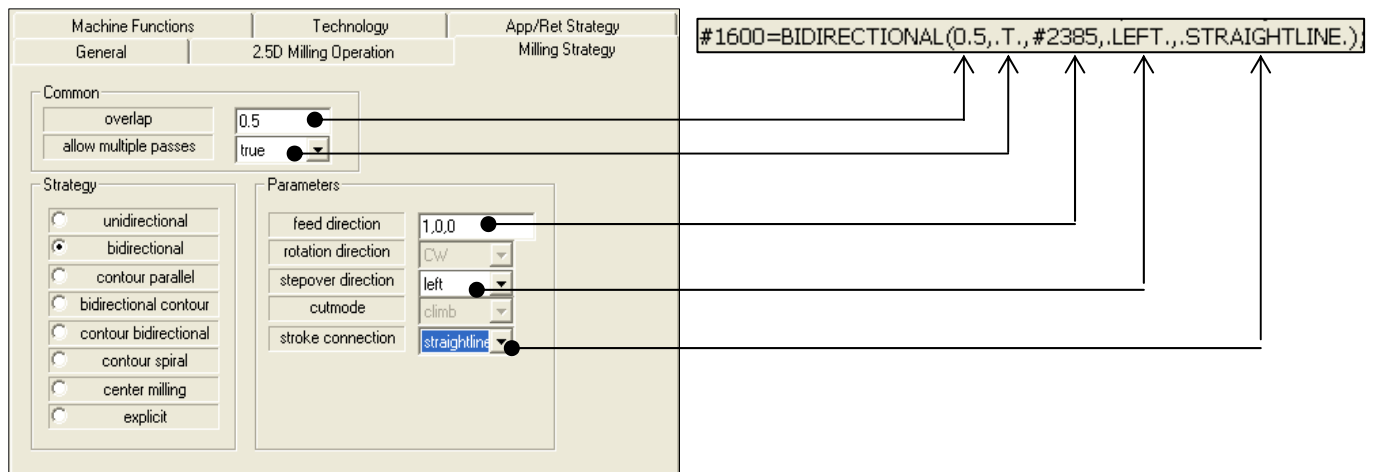


Fig. 44 : Exemple du choix de stratégie dans PosSFP et traduction dans le programme Step_NC

Nous pouvons remarquer que les paramètres apparaissent sous forme littérale dans le fichier Step_NC, ce qui permet la relecture. Le cas de la direction d'avance (*feed_direction*) est particulier dans la mesure où la valeur attendue s'exprime sous la forme d'un vecteur. Ce vecteur n'est pas directement précisé dans la fonction « Bidirectional » mais est reporté à la ligne repérée #2385 dans le programme Step_NC. Cela permet de mieux structurer la définition des attributs de la stratégie d'usinage.

- Les stratégies d'entrée/sortie matière (rampe, hélice, etc.)

L'implantation de l'usinage trochoïdal et du tréflage nécessite donc de prendre en compte la structure du fichier Step_NC et de s'interroger sur les paramètres et attributs permettant de construire les trajectoires outils. Nous nous sommes donc intéressés au choix de ces paramètres pour pouvoir par la suite les intégrer à PosSFP.

5.2. Sélection des paramètres nécessaires à la définition d'une trochoïde et du tréflage

La sélection des paramètres d'usinage trochoïdal et de tréflage s'appuie sur les travaux réalisés au sein de l'équipe MO2P par Matthieu RAUCH [Rauch & Hascoët 06]. Nous proposons ainsi de construire les trajectoires d'usinage en nous basant sur la construction préalable d'une courbe guide sur laquelle vient se greffer un motif. Le tableau suivant (Tableau 2) schématise la construction des trajectoires ainsi que les différents paramètres permettant de construire les motifs.

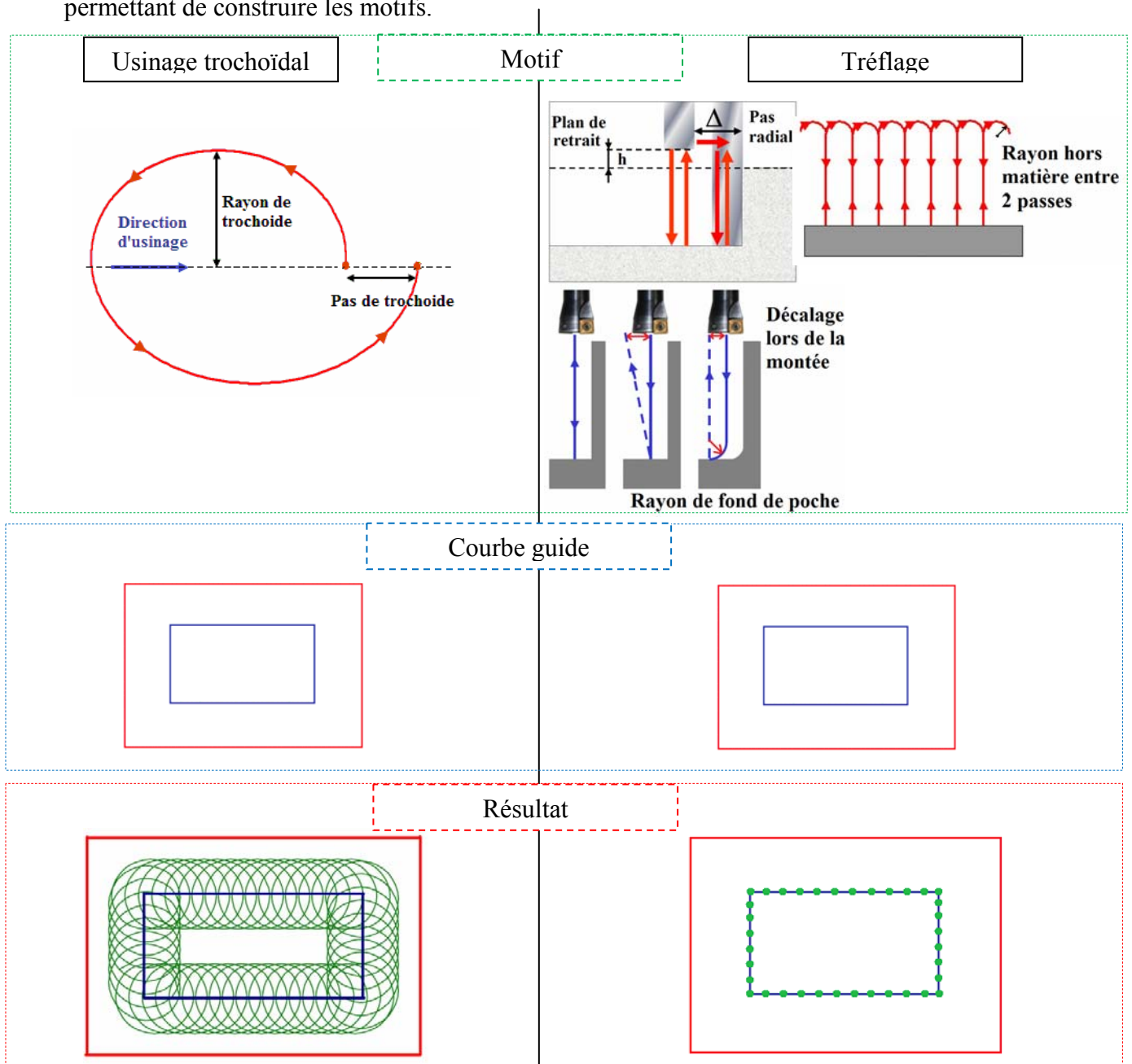


Tableau 2 : Construction des trajectoires à motifs d'usinage trochoïdal et de tréflage

L'usinage trochoïdal et le tréflage peuvent donc être décrits comme des trajectoires à motifs s'appuyant sur une courbe guide. Il sera donc nécessaire, dans le fichier Step_NC, de décrire ces deux composantes de la construction des trajectoires outils.

L'un des avantages majeurs de cette méthode de construction des trajectoires est la possibilité d'utiliser les trajectoires existantes dans la norme comme base pour pouvoir construire les courbes guides. De même, les paramètres de génération des courbes guide seront repris de ceux cités dans la norme, ce qui permet d'harmoniser la construction des courbes guides. Les motifs, quant à eux, nécessitent certains paramètres qui pourront être à celle-ci. Ainsi, en reprenant les paramètres de construction des motifs (définis dans le Tableau 2) et en ajoutant des courbes guides donc la construction se base sur les trajectoires définies dans la norme, nous sommes en mesure d'établir nos trajectoires d'usinage trochoïdal et de tréflage. Le tableau suivant (Tableau 3) reprend l'ensemble des informations qu'il est nécessaire de spécifier dans le programme Step_NC afin de générer correctement les trajectoires à motifs :

Stratégie	Usinage trochoïdal	Tréflage
Attributs de la stratégie : Courbe guide + attributs du motif	<ul style="list-style-type: none"> - Courbe guide - Rayon de trochoïde - Pas de trochoïde 	<ul style="list-style-type: none"> - Courbe guide - Hauteur du plan de retrait - Pas radial entre 2 passes - Décalage lors de la montée - Rayon de fond de poche - Rayon hors matière entre 2 passes
<i>Courbe guide</i>	<ul style="list-style-type: none"> - Unidirectionnelle - Bidirectionnelle - Contour parallèle - Contour spiral 	<ul style="list-style-type: none"> - Unidirectionnelle - Bidirectionnelle - Contour parallèle - Contour spiral
Attributs de la courbe guide	Attributs donnés par la norme pour la courbe guide sélectionnée	Attributs donnés par la norme pour la courbe guide sélectionnée
Attributs communs aux stratégies	<ul style="list-style-type: none"> - Overlap - Allow_multiple_passes 	<ul style="list-style-type: none"> - Overlap - Allow_multiple_passes

Tableau 3 : Paramètres nécessaires à la génération des trajectoires d'usinage trochoïdal et de tréflage

L'ensemble de ces paramètres étant à présent bien défini, l'étape suivante consiste à définir la forme sous laquelle nous serons en mesure de les intégrer dans le fichier Step_NC.

5.3. Implantation des stratégies dans le fichier Step_NC.

Pour faire apparaître l'ensemble des paramètres dans le fichier Step_NC et en se basant sur les stratégies déjà existantes, nous proposons la syntaxe suivante pour les stratégies à motifs :

#Réf=STRATEGY(overlap,allow_multiple_passes,#courbe_guide,attributs du motif)

- #Réf : fait référence à la ligne dans le programme Step_NC
- overlap et allow_multiple_passes sont les attributs communs à toutes les stratégies
- #courbe_guide : renvoie à la ligne faisant référence à la courbe guide ainsi que ses attributs.

Ce renvoi à une ligne différente pour définir la courbe guide présente l'avantage de mieux distinguer les paramètres liés au motif et ceux qui permettent de définir la courbe guide. La ligne pointée sera construite sous la forme suivante :

#Réf_courbe_guide=STRATEGY(attributs de la courbe guide)

- attributs de la stratégie : les différents attributs des stratégies sont séparés par des virgules.

Leur nombre dépend du fait que l'usinage trochoïdal ou le tréflage sont sélectionnés.

Nous avons donc l'ensemble des informations nécessaires à l'implantation de nos stratégies à motifs dans un modèle Step_NC : les paramètres et attributs de chacune des stratégies et la syntaxe dans le fichier Step_NC. Afin de mettre cela en œuvre et montrer la faisabilité de nos propositions d'implantation des stratégies à motifs dans le Step_NC, nous nous proposons d'implanter les stratégies usinage trochoïdal et tréflage dans le logiciel PosSFP qui permet la génération des fichiers Step_NC.

5.4. Mise en place des stratégies à motifs dans PosSFP

Les sources du logiciel PosSFP nous ont été fournies par le laboratoire coréen de l'Université de Postech pour que nous puissions enrichir l'ensemble des stratégies proposées avec l'usinage trochoïdal et le tréflage. Ce logiciel n'a pas pour but principal d'être commercialisé mais a été créé dans une optique de développement de la chaîne numérique Step_NC. Il s'enrichit au fur et à mesure des travaux coréens et ici, de nos travaux.

L'architecture de PosSFP se base sur une programmation orientée objet en C++ ainsi que sur l'utilisation de MFC (Microsoft Function Class). Ces dernières sont des bibliothèques de classes en C++ et permettent un rendu graphique inspiré du motif de conception modèle-vue-contrôleur de Windows. L'utilisation du logiciel Microsoft Visual C++ est la plus appropriée dans notre cas car elle est bien adaptée pour traiter les MFC.

L'espace de travail permettant le développement du logiciel est constitué de fichiers source, de fichiers « header », de fichiers Ressources ainsi que de dépendances externes. L'utilisation des MFC entraîne que l'ensemble des classes fenêtres est une encapsulation des contrôles Windows de base. Les classes forment une hiérarchie et il existe un ensemble de classes pour l'architecture de l'application, la gestion des fenêtres, des fichiers, des menus, etc. Pour en savoir plus sur les MFC, il existe plusieurs tutoriaux intéressants dont certains sont cités en référence [Tut. MFC].

Si l'on s'intéresse au cas précis des stratégies d'usinage, leurs choix se fait dans l'onglet « Milling Strategy » de la fenêtre « Operation Attributes ». Celle-ci correspond à la classe CTwo5DMillingStrategy en référence à la norme. L'interface graphique de Visual C++ permet d'insérer des boutons radio, des zones d'édition et des menus déroulants qui seront utiles à l'utilisateur pour sélectionner les bons paramètres des stratégies. A chacun de ces objets sont associés des identifiants qui permettent de récupérer et de traiter la valeur des champs correspondants. Certaines manipulations d'ordre de la pure programmation ont alors été mises en œuvre pour pouvoir adapter la génération du programme Step_NC à ces nouvelles entrées offertes à l'utilisateur.

Au niveau du graphisme de l'onglet des stratégies de fraisage (Fig. 45), nous avons fait le choix de séparer les stratégies à motifs (Patterned strategies) ainsi que leurs attributs respectifs de l'ensemble des autres stratégies. La justification de ce choix se fait dans la manière différente de les générer en se basant sur les stratégies déjà existantes pour construire la courbe guide.

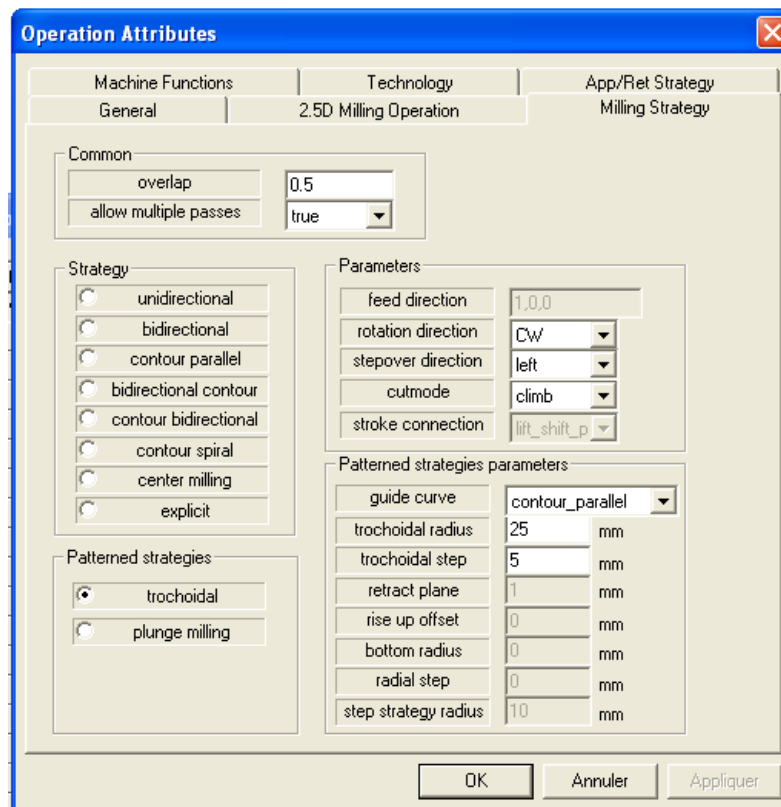


Fig. 45 : Choix de la stratégie d'usinage dans PosSFP suite à l'implantation de l'usinage trochoïdal et du tréflage

Après avoir sélectionné les paramètres des stratégies, PosSFP est en mesure de générer le fichier Step_NC. Celui-ci peut être exporté dans divers formats dont le format texte.

Remarque importante : pour des questions de programmation dans Visual C++, la version de PosSFP que nous avons développé (PosSFP V4) ne permet pas encore de renvoyer la définition de la courbe guide sur une autre ligne. L'ensemble des attributs ainsi que ceux de la courbe guide sont donc tous indiqués sur la même ligne de la manière suivante :

```
#STRATEGY(overlap,allow_multiple_passes,Courbe_guide,attributs de la courbe guide,attributs du motif)
```

Syntaxe utilisée dans PosSFP V4 au lieu d'un renvoi à une ligne #Courbe_guide pour la définition de la courbe guide

Le résultat dans PosSFP est le suivant (Fig. 46):

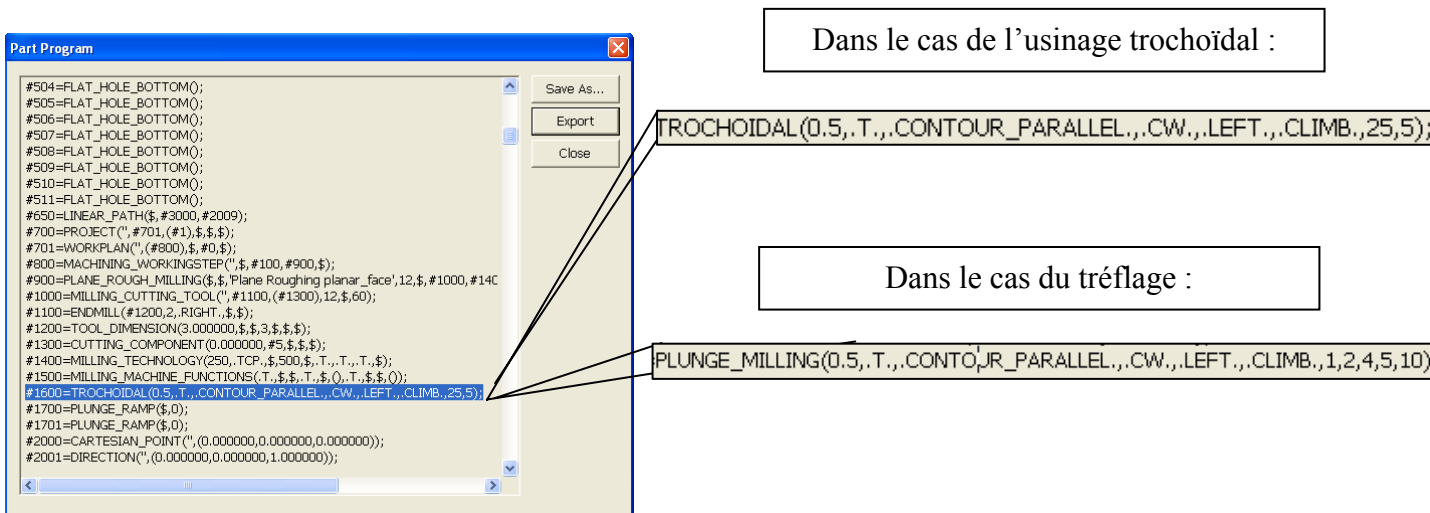


Fig. 46 : Syntaxe des stratégies à motifs dans le fichier Step_NC

Cette étape d'implémentation étant franchie, nous pouvons tout de même nous interroger sur les possibilités d'amélioration qui pourraient être mises en œuvre pour simplifier certaines définitions ou éliminer quelques défauts de l'ordre de la programmation ou de la syntaxe.

5.5. Possibilités d'amélioration

La première amélioration, comme nous l'avons évoqué dans le paragraphe précédent, consiste à implanter la syntaxe de définition de la courbe guide en renvoyant celle-ci à une autre ligne du fichier Step_NC par l'intermédiaire d'un lien (#Courbe_guide). Cela permettrait de clarifier la définition des courbes à motifs.

Dans une autre phase de recherche, il serait intéressant d'améliorer le choix des critères ainsi que des valeurs de sélection des paramètres. Cela est à envisager pour permettre à un utilisateur non spécialiste de l'usinage trochoïdal et du tréflage de sélectionner des paramètres par défaut calculés en fonction de la géométrie de la pièce et de l'outil. En effet, cette première étape de l'implantation de l'usinage trochoïdal et tréflage a pour but de fixer les bases en terme de définition des paramètres et attributs nécessaires à la génération des trajectoires mais nous pouvons aisément imaginer que le choix des valeurs de ces paramètres a une grande influence sur la productivité. Il reste donc dans ce domaine des avancés à effectuer en recherche qui pourront par la suite être intégrées à la maquette numérique Step_NC pour en tirer tous les avantages liés à la simplicité d'utilisation et à la productivité.

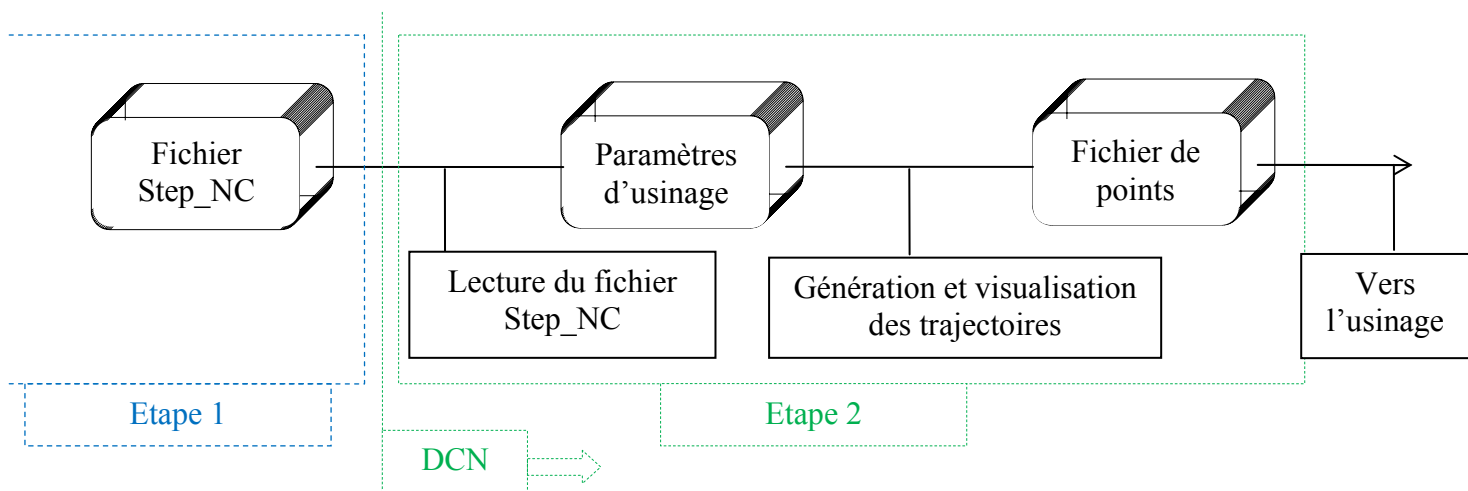
5.6. Conclusion de la première étape

Cette première étape nous a permis de mettre en place l'ensemble des attributs et paramètres permettant de construire les trajectoires d'usinage trochoïdal et de tréflage. A partir de cela, nous avons proposé une syntaxe d'implémentation dans le fichier Step_NC fondée sur les stratégies d'usinage déjà existantes au sein de la norme. Concrètement, ces propositions se sont traduites concrètement par l'implémentation de l'usinage trochoïdal et du tréflage dans le logiciel PosSFP permettant la génération du fichier Step_NC.

La construction des stratégies à motifs présente un réel intérêt quant à la simplicité de mise en œuvre et des modifications qui sont alors possibles. Ces propriétés s'accordent parfaitement avec la philosophie liée au développement du format Step_NC. Les étapes suivantes nous permettront de mettre en avant les possibilités qui s'offrent à nous en relation avec la programmation et la génération des trajectoires à motifs en Step_NC. Nous serons alors en mesure de montrer la faisabilité de ce que nous venons de proposer dans l'étape 1 tout au long de la chaîne numérique Step_NC, de la maquette CAO jusqu'à la pièce usinée.

6. Etape 2 : Du fichier Step_NC à l'usinage

Nous avons donc, dans la partie précédente, mis en place l'ensemble des éléments nécessaires à la génération du fichier Step_NC et implanté l'usinage trochoïdal et le tréflage dans PosSFP, ce qui nous permet d'obtenir rapidement et aisément le fichier Step_NC. L'étape suivante, schématisée ci-dessous, consiste à traiter ce fichier Step_NC pour en extraire les paramètres nécessaires à la génération des trajectoires.



Dans la chaîne numérique Step_NC, l'ensemble des étapes qui vont être développées par la suite peuvent être resituées physiquement au niveau du directeur de commande numérique (DCN). En effet, le Step_NC est développé de telle façon qu'une partie de l'« intelligence » peut être déplacée au pied de la machine pour que celle-ci puisse mieux interagir et effectuer des corrections en temps réel de l'usinage.

Nous avons donc mis au point un ensemble d'outils qui rendent possibles, à partir de la lecture du fichier Step_NC, la construction des trajectoires et la création de l'ensemble des points pour l'usinage. Dans le temps qui nous était imparti, nous avons fait le choix de nous concentrer en particulier sur l'usinage trochoïdal, ce qui nous a permis de réaliser l'ensemble de la chaîne numérique complète. La construction des courbes à motifs présentant de grandes similarités entre les différentes stratégies, la génération du tréflage pourrait être calquée sur la démarche que nous allons présenter pour l'usinage trochoïdal.

L'ensemble des outils que nous avons mis au point ont pour la plupart été développés grâce au logiciel Matlab, pour son utilisation simple et les nombreuses possibilités qu'il offre.

La première étape de cette partie consiste en la lecture du fichier Step_NC.

6.1. Lecture du fichier Step_NC

La lecture du fichier Step_NC consiste à récupérer l'ensemble des valeurs des paramètres qui vont nous être nécessaires pour pouvoir construire les trajectoires. Nous avons tout d'abord répertorié ces paramètres dans le cas de l'usinage trochoïdal (Tableau 4) :

Sous opération	Paramètres à récupérer (usinage trochoïdal)
Géométrie de la pièce	<ul style="list-style-type: none">- Coordonnées des points caractéristiques des entités. Dans le cas d'une poche simple : 4 sommets
Entrée en matière	<ul style="list-style-type: none">- Angles de rampe
Génération des trajectoires	<ul style="list-style-type: none">- Courbe guide- Attributs de la Courbe guide- Rayon de trochoïde- Pas de trochoïde- Rayon d'outil
Dégagement	<ul style="list-style-type: none">- Plan de retrait
Paramètres de coupe	<ul style="list-style-type: none">- Vitesse d'avance- Vitesse de coupe- Profondeur de passe

Tableau 4 : Paramètres d'usinage trochoïdal à récupérer dans le fichier Step_NC

Nous avons ensuite développé une routine Matlab afin de lire le fichier Step_NC. Pour des questions de simplification, nous avons limité la géométrie de la poche à une poche simple quelconque à quatre sommets. Cependant, si nous sommes capables de traiter facilement une poche simple pour la construction de la courbe guide, des formes plus complexes (poches avec angles de dépouille ou rayons de raccordement aux sommets, bossages, etc.) demanderaient quelques adaptations qui pourraient faire l'objet de futurs travaux de recherche. Le fichier Step_NC est exporté de PosSFP avec l'extension .snc. C'est un format de type ASCII qui peut être lu par tous les éditeurs de texte ainsi que par Matlab.

Nous avons choisi de programmer la lecture du fichier Step_NC sous forme d'une fonction, ce qui permettra par la suite de construire un programme Matlab automatisant toutes les étapes de la chaîne numérique Step_NC NC.

Pour expliquer rapidement la façon dont a été conçu le programme, les boucles de récupération des données commencent par lire chacune des lignes du fichier Step_NC. A partir de cela, elles sont en mesure de reconnaître des mots clefs qui lui sont proposés dans le

programme (comme par exemple le mot « TROCHOÏDAL »). Une fois le mot repéré, on rentre les différents éléments entre parenthèses dans un tableau de cellules, ce qui permet de récupérer l'ensemble des données, que la nature de celles-ci soit un entier, un booléen, une chaîne de caractères (lettres). Les cellules du tableau sont remplies par les données séparées par des virgules. Les manipulations des données sont alors faciles par l'intermédiaire de ces tableaux. Il est nécessaire de faire attention à la place des paramètres que nous souhaitons récupérer, car celle-ci varie suivant la courbe guide sélectionnée (en effet, le nombre d'attributs lié aux courbes guide diffère suivant la stratégie sélectionnée).

Il est un peu plus complexe de récupérer la géométrie de la poche car cela implique de suivre plusieurs liens avant de pouvoir accéder aux pointeurs qui la définissent. Malgré cela, nous avons utilisé une structure similaire à la précédente pour pouvoir lire les données de géométrie.

Dans cette première phase de développement, nous avons restreint le choix de la géométrie d'une pièce à quatre sommets. Il est donc suffisant de lire dans le fichier Step_NC les coordonnées de ces quatre sommets.

Notre routine Matlab (fonction intitulée « reconnaissanceForme() ») est en mesure de récupérer la géométrie de la pièce, la nature de la courbe guide et les attributs de la stratégie usinage trochoïdal. Nous pouvons donc à présent générer les trajectoires d'usinage.

6.2. Génération des trajectoires pour l'usinage trochoïdal

Tout d'abord, revenons sur la construction des trochoïdes. Une trochoïde (ou épicycloïde) est une courbe décrite par un point lié à un cercle de rayon R roulant sans glisser sur une droite.

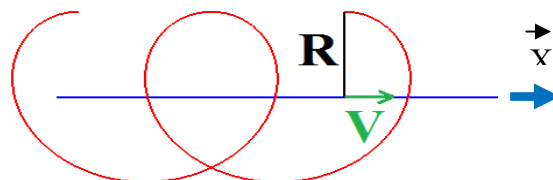


Fig. 47 : définition d'une trochoïde

$$x = V \times t + R \times \cos(\omega.t) \quad \text{Eq. 1}$$

En coordonnées cartésiennes, la trajectoire paramétrée s'écrit :

$$y = R \times \sin(\omega.t)$$

En usinage, cela se traduit concrètement dans le cas particulier d'une rainure par les paramètres suivants :

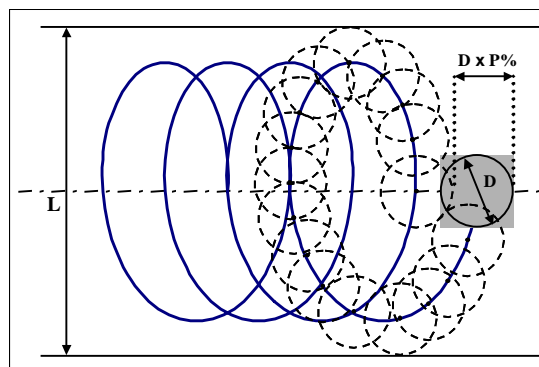


Fig. 48 : Données d'entrée pour la génération d'une trajectoire trochoïdale [Rauch Hascoët 06]

Ces paramètres peuvent aussi s'exprimer à l'aide de ceux récupérés dans le fichier Step_NC :

- $L = 2x(\text{Rayon_outil} + \text{Rayon_trochoïde})$: Largeur de la rainure
- $D = D_outil = 2x\text{Rayon_outil}$: Diamètre de l'outil utilisé
- $P\% = \text{Pas_trochoïde}/D$: Pas de balayage (exprimé en % du diamètre outil)

La correspondance entre ces paramètres d'usinage et l'équation paramétrée de la trochoïde (Eq. 1) se fait de la manière suivante :

- Le paramètre t est incrémenté pour pouvoir définir les coordonnées des points de la trochoïde. Pour des raisons de simplification et de clarification, nous avons choisi une pulsation $\omega = 2\pi$. Cela permet de réaliser un motif complet (un tour) de trochoïde pour un incrément unitaire de t . Le produit $\omega.t$ représente alors l'angle entre la droite reliant le centre instantané de la trochoïde et le point de la trochoïde, et le vecteur directeur de la courbe guide.
- Le paramètre V représente dans ce cas directement le pas de la trochoïde. On a alors : $V = P\% \times D_outil$.

Cette définition des points de la trochoïde fonctionne parfaitement dans le cas d'une courbe guide d'axe \vec{x} et d'origine O . Ce ne sera évidemment pas toujours le cas et il est donc nécessaire de donner une définition dans le cas d'un vecteur directeur de la courbe guide quelconque. Les coordonnées cartésiennes des points de la trochoïde sont alors définies de la manière suivante (Eq. 2) :

$$\begin{cases} x = X_0 + V_x \times (t - t_0) + R \times \cos(\omega \times t) \\ y = Y_0 + V_y \times (t - t_0) + R \times \sin(\omega \times t) \end{cases} \quad \text{Eq. 2}$$

Cette définition est bien adaptée dans le cas d'une courbe guide polygonale. Chacun des segments constituant la courbe guide est considéré localement comme une courbe guide simple dont le vecteur directeur unitaire est projeté sur chacun des axes. Un décalage des coordonnées du point d'origine ainsi que de l'incrément permettent de réinitialiser la construction à chaque segment. Cette réinitialisation n'a pas lieu dans les fonctions sinusoïdales, ce qui permet de repartir de la position du segment précédent et ainsi d'assurer la continuité de la courbe.

Construction de la courbe guide :

A partir des coordonnées des sommets de la poche récupérés dans le fichier Step_NC, nous sommes en mesure de construire la courbe guide. Comme nous l'avons expliqué de façon détaillée dans les chapitres précédents, il existe plusieurs stratégies supports de la courbe guide. Dans le cadre de nos travaux, nous en avons développé deux en particulier, pour bien montrer la faisabilité de la définition des courbes à motifs que nous proposons, tout en restant

dans les limites du réalisable en un temps imparti. Ces deux stratégies sont le contour parallèle et le zig-zag (bidirectionnel).

Construction de la courbe guide liée à la stratégie contour parallèle :

Dans le cas d'une poche de forme quadrilatère, la courbe guide correspondante est un quadrilatère. Les coordonnées des sommets de la courbe guide sont calculées de la manière suivante (Fig. 49) :

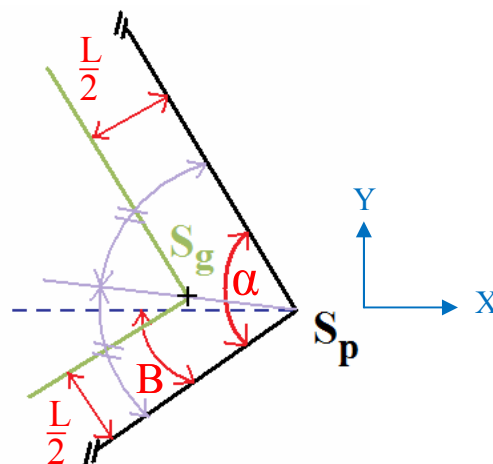


Fig. 49 : Calcul des coordonnées d'un sommet de la courbe guide

L'ensemble des coordonnées des sommets de la poche étant connu, il est possible de calculer les angles au sommet (α) du quadrilatère ainsi que l'angle avec le repère local (β). Le sommet de la courbe guide (S_g) se situe sur la bissectrice de l'angle et sa distance aux cotés de la poche est $L/2$ (Largeur de la rainure). Un calcul de trigonométrie sur ces paramètres nous permet d'accéder aux coordonnées locales des sommets de la courbe guide. Elles sont ensuite transférées dans le repère lié à la pièce.

Pour automatiser ces calculs, nous avons développé un programme Matlab (courbeguide.m) qui renvoie les coordonnées des quatre sommets de la courbe guide.

Si la poche n'est pas vidée avec un seul contour parallèle mais qu'elle ne nécessite pas un second contour parallèle « à l'intérieur du premier », la courbe guide est étendue d'un segment rejoignant les milieux des cotés du contour parallèle, ce qui permet de vider la matière restante (Fig. 50) ;

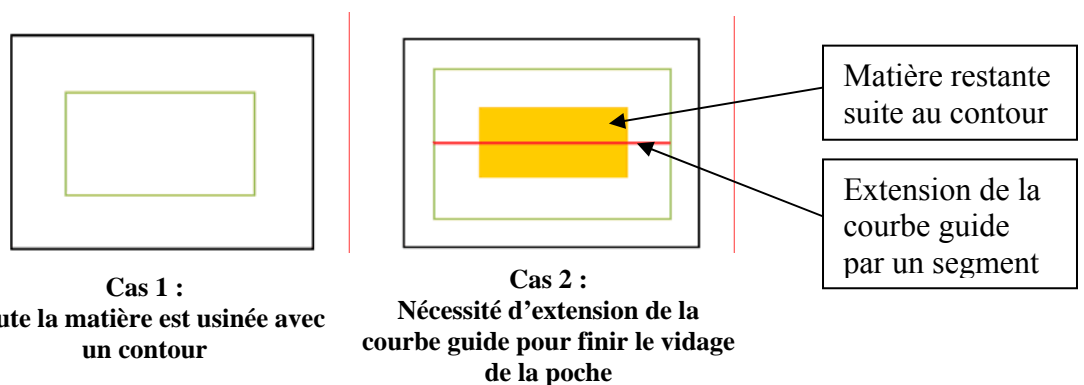


Fig. 50 : Cas 1 : Toute la matière est usinée avec un contour

Cas 2 : Nécessité d'extension de la courbe guide pour finir le vidage de la poche

Ainsi, lorsque la courbe guide spécifiée est construite sur le modèle de la stratégie contour parallèle, le programme Matlab est en mesure de sélectionner, en fonction des paramètres de trochoïde et de l'outil si la poche peut être vidée en un seul contour (dans ce cas il y a « recouvrement » des usinages) ou si la pièce nécessite un dernier passage pour enlever la matière restante. Nous nous sommes restreints à traiter ces 2 cas.

Si l'on généralise encore l'usinage d'une poche avec une courbe guide contour parallèle, dans le cas de poches de dimensions importantes, il est possible que celle-ci nécessite plusieurs contours parallèles les uns dans les autres. L'implantation pourrait alors se faire en construisant directement de nouvelles courbes guides en considérant les dimensions de la matière restante comme celle d'une nouvelle poche. Cette méthode semblerait la plus simple mais nécessiterait une passe supplémentaire pour éliminer la matière restante dans les coins (Fig. 51).

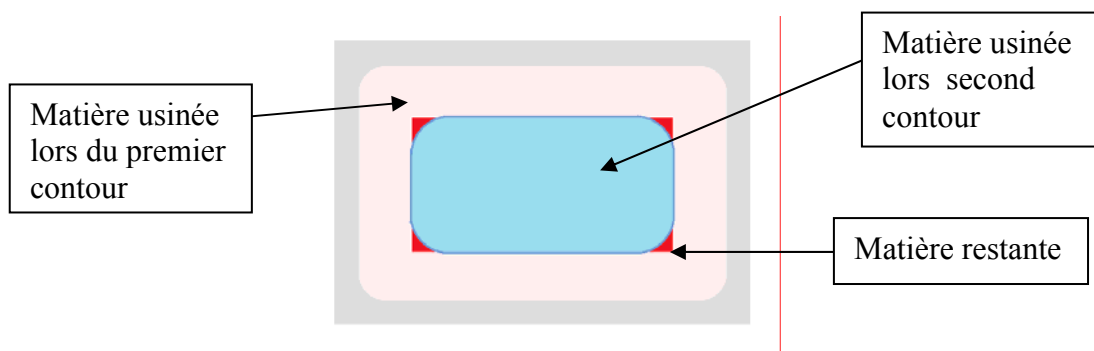


Fig. 51 : Usinage avec plusieurs contours

Une autre méthode pourrait consister à modifier (agrandir aux extrémités des segments de la courbe guide par exemple) le rayon de trochoïde localement au niveau des sommets pour être sûr de bien usiner toute la matière. Cette réflexion pourra avoir lieu dans des travaux ultérieurs.

Construction de la courbe guide liée à la stratégie zig-zag :

Pour construire cette courbe guide, nous nous sommes basés sur les sommets de la courbe guide obtenue précédemment. Nous la nommerons ici « courbe support ». A partir de cela, nous construisons la courbe zig-zag comme suit :

- Calcul de la longueur des côtés opposés de la courbe support
- Calcul de la largeur de la rainure engendrée par l'usinage avec les paramètres récupérés
- Calcul du nombre de sommets nécessaires pour la courbe zig-zag (c'est la partie entière du rapport de la longueur du plus grand coté sur la largeur de la rainure).
- Calcul des coordonnées des sommets.

Le résultat provisoire de l'implémentation de ce type de courbes guide se traduit, dans le cas d'une poche quelconque, par des zig-zag de directions non colinéaires. La courbe guide ne suit donc pas à proprement dit une stratégie zig-zag comme nous avons l'habitude d'entendre (Fig. 52). Cette construction a été conçue ainsi pour des raisons de commodité de programmation. Après réflexion, une stratégie zig-zag classique est certainement plus efficace dans la majorité des cas.

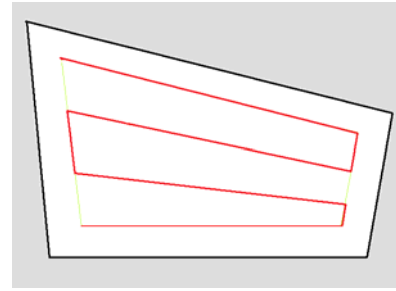
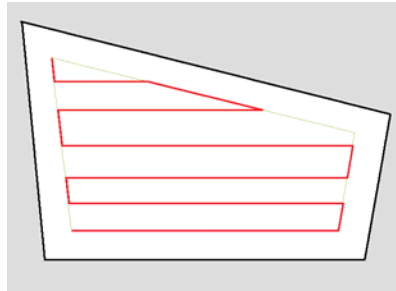


Fig. 52 : Cas 1 : « vrai » zig-zag

Cas 2 : zig-zag implémenté

Une fois la courbe guide définie, nous pouvons maintenant calculer les points de la trajectoire avec la définition de la trochoïde que nous avons donnée précédemment (Eq. 2).

Le programme de construction des points de la trajectoire a été développé avec Matlab (generationPointsTrochoïde.m). Outre la génération des points, il permet de visualiser les trajectoires et également créer un fichier texte contenant l'ensemble des points. Le nombre de points, qui correspond à la discrétisation de la trochoïde, dépend directement de l'incrément que nous fixons pour le paramètre t . Si le nombre obtenu de points est trop important, il risque d'entraîner des problèmes de calcul dans le DCN lors de l'usinage. A l'inverse, s'il est trop faible, il risque d'entraîner un mauvais suivi de la trajectoire.

Pour les deux types de courbes guide que nous avons développés, la génération et la visualisation des trajectoires trochoïdales donnent les résultats suivants dans le cas du vidage d'une poche de dimensions 130 x 90 (Tableau 5):

Stratégie – Courbe guide	Paramètres	Visualisation
Contour parallèle seul	$R_{\text{trochoïdal}} = 17 \text{ mm}$ $R_{\text{outil}} = 10 \text{ mm}$	

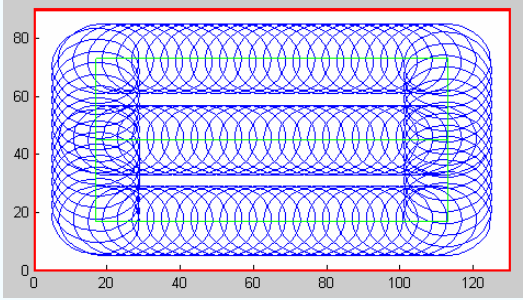
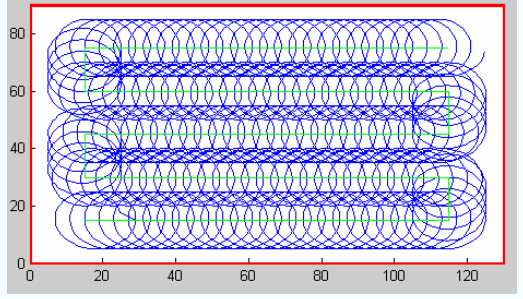
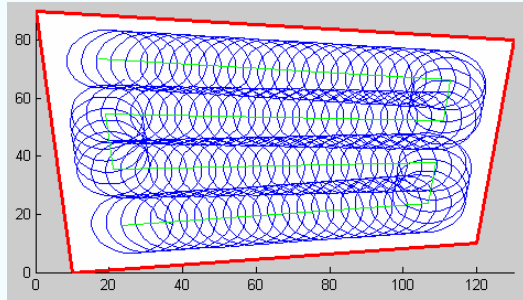
<p>Contour parallèle + milieu</p>	<p>$R_{\text{trochoïdal}} = 16 \text{ mm}$ $R_{\text{outil}} = 5 \text{ mm}$</p>	
<p>Zig-zag</p>	<p>$R_{\text{trochoïdal}} = 12 \text{ mm}$ $R_{\text{outil}} = 5 \text{ mm}$</p>	
<p>Zig-zag Poche quelconque</p>	<p>$R_{\text{trochoïdal}} = 12 \text{ mm}$ $R_{\text{outil}} = 5 \text{ mm}$</p>	

Tableau 5 : Quatre stratégies test d'usinage trochoïdal

Les entrées matière ont été construites à partir de là par une plongée en Z proportionnelle pour chaque incrément de t de telle façon que l'outil prenne la profondeur de passe indiquée au bout de trois motifs de la trochoïde. Cela représente, dans les usinages effectués par la suite, un angle de rampe suffisamment faible pour que l'outil ne talonne pas. Il reste cependant dans ce domaine particulier quelques améliorations à mettre en place pour optimiser l'entrée matière en fonction des capacités de l'outil. Enfin, le dégagement matière se fait directement par une montée en Z pour éviter toute collision.

6.3. Conclusion et bilan de la génération des trajectoires

L'ensemble des outils que nous avons développé permet de générer les points de la trochoïde à partir des paramètres récupérés dans le fichier Step_NC. Les courbes guides supports des

trajectoires à motifs que nous avons implémentées se basent sur les stratégies contour parallèle et zig-zag. Il est, sur le même modèle, possible d'étendre le choix de la courbe guide à d'autres possibilités comme l'unidirectionnel, le contour spiral ou encore l'association de plusieurs de ces stratégies.

Nous avons donc démontré dans cette partie la faisabilité de la génération des trajectoires trochoïdales avec les paramètres des courbes à motifs définis précédemment. Nous avons vu, dans la première étape, que certains attributs des stratégies peuvent être « prioritaires » dans certains cas de conflit (par exemple, le paramètre *Cutmode* est prioritaire sur le paramètre *feed_direction* en cas de conflit). Nous avons vu apparaître dans la génération des trajectoires trochoïdales certains conflits : il n'est pas possible d'imposer un rayon d'outil et de trochoïde tout en donnant une valeur au paramètre *overlap* (recouvrement entre passes). Dans ce cas, nous avons privilégié la définition de la trochoïde. De la même façon, le choix direct dans PosSFP de la valeur des paramètres des stratégies à motifs enlève certaines possibilités d'optimisation suivant le type de poche. Si l'on ajoute à cela le choix potentiellement hasardeux de ces choix, les stratégies à motifs peuvent perdre une partie de leurs avantages en termes de productivité. Cependant, dans la première phase d'implémentation, nous avons fait le choix de laisser à l'utilisateur l'entière possibilité de rentrer ces paramètres puisque nous ne sommes pas encore dans une démarche de gain de productivité mais plutôt de développement et d'implémentation des stratégies à motifs dans la chaîne numérique Step_NC.

De plus, bien que nous ayons développé ici le cas particulier de l'usinage trochoïdal, nous pouvons noter que la démarche pourrait être reprise dans le cas d'autres stratégies à motifs comme par exemple le tréflage.

L'étape suivante de l'intégration des stratégies à motifs dans la chaîne numérique Step_NC consiste à mettre en œuvre ces outils dans le cas de l'ébauche de poches.

7. Etape 3 : Mise en œuvre de la chaîne numérique Step_NC et validation par l'usinage de pièces.

Nous avons mis au point dans les étapes précédentes l'ensemble de la chaîne numérique Step_NC pour l'usinage trochoïdal. A partir de la maquette numérique CAO d'une pièce, nous sommes en mesure de construire le fichier Step_NC par l'intermédiaire du logiciel PosSFP. Ce fichier Step_NC sert ensuite de base pour la génération des trajectoires d'usinage. Notre apport se traduit par l'implémentation de stratégies à motifs dans cette chaîne numérique. Nous allons maintenant nous intéresser à la mise en œuvre de l'ensemble de cette chaîne numérique Step_NC dans le cas du vidage de poche. A partir de la maquette CAO de pièces test, nous utiliserons l'ensemble des outils que nous avons développé jusqu'à l'usinage de la pièce.

7.1. Construction des pièces test et traitement dans la chaîne numérique Step_NC

Notre choix s'est porté sur la conception de 2 pièces test simples. L'une d'entre elles est constituée d'une poche rectangulaire et la seconde est une poche quelconque (Fig. 53).

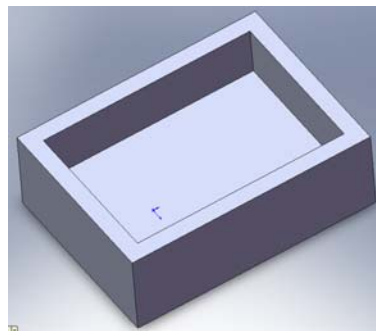
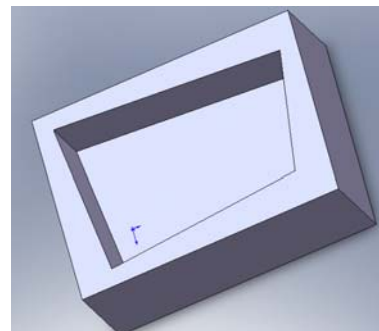


Fig. 53 : Pièce 1 : Poche rectangulaire



Pièce 2 : Poche quelconque

Les poches de ces pièces test sont simplifiées au maximum (pas de rayon de raccordement aux sommets, de dépouille, etc.) et ont pour forme un quadrilatère pour pouvoir être traités avec les outils que nous avons développés. A partir de ces deux formes de poche test, nous réaliserons la chaîne numérique Step_NC complète et l'usinage de quatre pièces test qui nous permettront de montrer :

- le vidage de poche rectangulaire en usinage trochoïdal avec une courbe guide contour parallèle seul
- le vidage de poche rectangulaire en usinage trochoïdal avec une courbe guide contour parallèle et finition de la matière restante suite au contour
- le vidage de poche rectangulaire en usinage trochoïdal avec une courbe guide zig-zag (bidirectionnel)
- le vidage de poche quelconque en usinage trochoïdal

Les 2 modèles CAO ont été construits sous Solidworks et exportés dans un format parasolid pour être facilement compatibles avec le traitement dans PosSFP. Nous avons ensuite traité ces 4 pièces avec l'ensemble des outils que nous avons développé, du modèle CAO jusqu'au fichier de points pour l'usinage. Nous avons ainsi généré le fichier Step_NC à partir de la reconnaissance de formes et d'entité dans le logiciel PosSFP. Le fichier Step_NC a ensuite été lu et traité par l'ensemble de nos outils Matlab pour donner lieu à la génération des trajectoires. L'ensemble des points des trajectoires ont alors été regroupés dans un fichier texte qui servira à l'usinage de la pièce.

7.2. Usinage sur les moyens à disposition : nécessité d'un interpréteur

Malgré l'existence de prototypes de directeurs de commande numériques permettant d'interpréter directement le fichier Step_NC, ils ont été très peu diffusés et nous ne disposons pas actuellement de tels moyens. C'est la raison pour laquelle nous avons développé nos propres outils pour pouvoir lire le fichier Step_NC, générer les trajectoires et usiner la pièce. Il a donc fallu que nous prenions en compte les spécificités des moyens qui sont à notre disposition.

7.2.1. Machine parallèle Verne et Siemens 840D

La machine Verne est une machine à structure parallèle. En termes de programmation, cela a très peu d'impact au niveau de l'utilisateur dans la mesure où c'est le directeur de commande numérique qui effectue l'ensemble des transformations géométriques nécessaires à l'usinage des trajectoires programmées. Le directeur de commande numérique installé sur cette machine est un Siemens 840D. La définition des usinages se fait, comme sur la grande majorité des machines actuelles, à l'aide du code G. Il existe cependant quelques fonctions supplémentaires ajoutées aux fonctions de base de la norme ISO 6983 régissant le code G. Ces fonctions, spécifiques à Siemens, peuvent être utiles pour des usinages nécessitant des trajectoires élaborées, comme par exemple la fonction CSPLINE que nous serons amenés à utiliser.

Les principales caractéristiques de la machine sont :

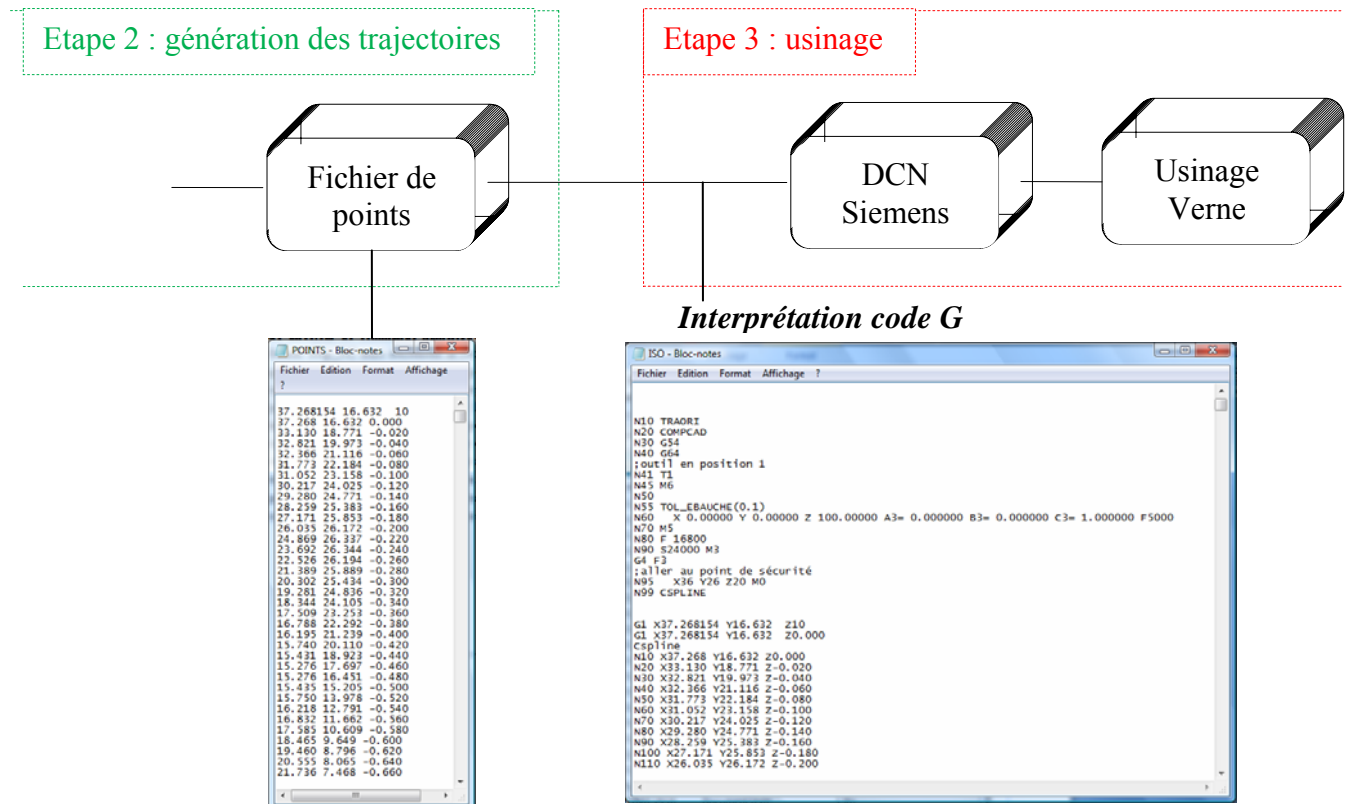
- Vitesse de rotation de la broche : 26000 tours/min
- Accélération : 15m/s^{-2}
- Avance maxi : 50 m/min
- Puissance à la broche : 40 kW



Fig. 54 : Machine parallèle Verne et Siemens 840D à l'IRCCYN

7.2.2. L'interpréteur

Comme nous venons de le voir, le directeur de commande numérique Verne n'intègre pas le format Step_NC mais seulement le code G. Il nous est donc nécessaire d'interpréter les stratégies générées en code G pour pouvoir usiner la pièce. Cette démarche peut paraître contradictoire avec l'utilisation du Step_NC dans la mesure où le développement de ce format a pour objectif final de remplacer le code G. Nous sommes cependant aujourd'hui dans une impasse car nous n'avons pas à disposition de DCN capable de s'affranchir du code G. Pour combler ce manque, certaines équipes de recherche (comme par exemple le laboratoire de Postech en Corée) ont développé leur propre prototype qui n'utilise plus de code G et traite les déplacements en pilotant directement les axes de la machine. De la même manière, nous pourrions développer nos propres algorithmes pour pouvoir piloter la machine sans avoir à passer par le code G mais cela ne représente pas, dans notre cas, un intérêt suffisant pour engager ce travail conséquent.



7.2.3. Usinage par interpolation de type Spline

Pour relier l'ensemble des points que nous avons généré, le DCN SIEMENS intègre plusieurs fonctions d'interpolation permettant de lisser les courbes. Cela est particulièrement utile dans le cas de l'usinage trochoïdal car les trajectoires sont relativement complexes. Une interpolation de type linéaire pourrait alors engendrer beaucoup de discontinuités et de changements de direction, ayant pour effet direct l'augmentation des calculs du DCN et une diminution de la productivité par non respect des conditions de coupe. Cela peut aussi engendrer des phénomènes de vibration peu appréciables. La programmation en G1 semble donc mal adaptée ici.

Le DCN SIEMENS propose plusieurs solutions pour remédier à cette lacune du code G. Trois d'entre elles sont particulièrement intéressantes : la Spline A (Akima), B (Bézier) et C (cubique) (Fig. 55). Les principales caractéristiques de ces courbes sont les suivantes :

- Spline A : c'est une interpolation des points (elle passe exactement par les points intermédiaires fournis). Elle assure la continuité en tangence mais pas en courbure. Son avantage est qu'elle ne génère que peu d'oscillations indésirables.
- Spline B : c'est une approximation des points (elle ne passe pas par les points intermédiaires). Une Bspline de troisième degré ne génère pas d'oscillation, malgré une continuité en tangence et courbure. Elle est souvent utilisée pour la description des trajectoires d'outils pour les surfaces gauches.
- Spline C : c'est une interpolation similaire à la Spline A mais dont la continuité est assurée en courbure. Elle peut être source d'oscillations indésirables.

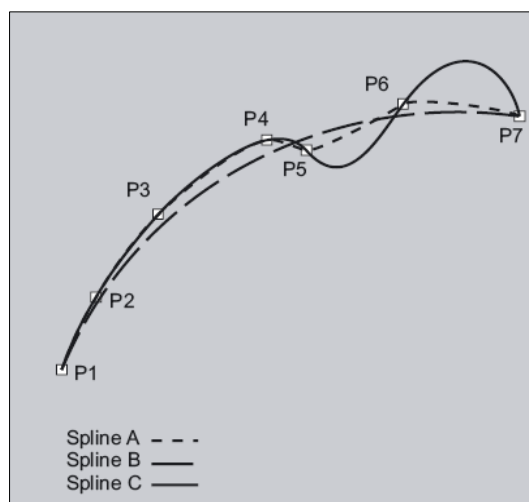


Fig. 55 : Courbes Spline disponibles [Pro Siemens]

Nous avons choisi de programmer l'usinage en Cspline pour les propriétés de cette interpolation (passe par les points intermédiaires, continuité du second ordre). De plus, le fait

que la trajectoire trochoïdale soit basée sur une courbe analytique connue limite fortement les risques d'oscillations indésirables.

Par ailleurs, nous pouvons faire une remarque concernant le nombre de points définissant la trajectoire trochoïdale. Il est possible, lors de la génération des points de la trajectoire outil, de jouer sur l'incrément de t dans l'équation de la trochoïde (Eq. 2), ce qui a pour effet de régler le nombre de points par motif. Certaines études préalables [Rauch & Hascoët 06] ont montré que, dans le cas de l'usinage trochoïdal en Cspline, une dizaine de points par motifs était suffisant pour définir les trajectoires. C'est ce que nous avons choisi dans notre cas. Nous avons également réalisé un essai à 50 points par motif : le résultat prévisible a été un usinage très perturbé par des problèmes de calcul au niveau du DCN, les vitesses programmées n'étant atteintes qu'à 50%. Le choix de ce nombre de points par motif est donc important. En effet, d'une part il ne faut pas dénaturer le motif en sélectionnant trop peu de point. A l'inverse, si il y a trop de points, il est possible que l'usinage soit ralenti suite au dépassement des capacités de calcul du DCN.

7.3. Validation expérimentale

Dans un premier temps, nous avons simulé l'usinage avec le simulateur d'usinage développé à l'IRCCyN [Dugas 02a]. Cela nous a permis de constater la cohérence des trajectoires avant usinage et de vérifier les entrées/sorties matière. Nous avons alors pu constater quelques défauts de la pièce usinée liés à la construction des stratégies. En effet, le centre instantané de la trochoïde évoluant sur la courbe guide, lorsque cette dernière a été entièrement parcourue, il est possible que l'outil soit en pleine matière. Il serait donc nécessaire qu'il termine sa trajectoire. Cela est particulièrement visible dans le cas d'une courbe guide zig-zag car l'outil revient à sa position de départ pour commencer la passe suivante (Fig. 56).

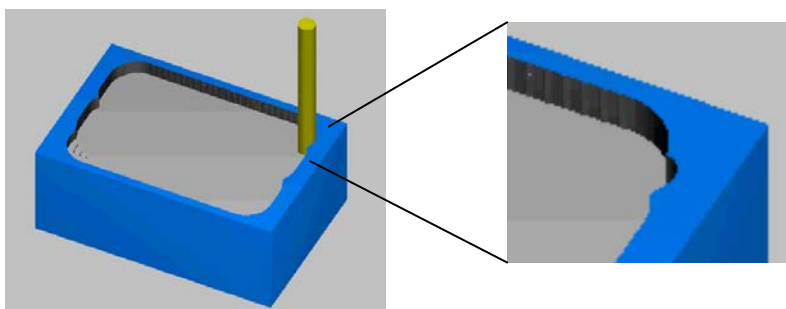





Fig. 56 : Outil dans sa position extrême dans le cas du zig-zag

Il serait donc possible de résoudre ce problème en terminant la trochoïde en cours pour éviter que l'outil ne se retrouve en pleine matière en fin de courbe guide. L'optimisation des trajectoires n'étant pas le point central de nos recherches pour l'instant, ce détail ne fait pas partie de nos priorités mais pourra être repris par la suite. Nous avons donc réalisé l'usinage

des pièces test que nous avons mis au point avec les différentes courbes guide que nous avons implémentées.

Ces usinages ont été menés sur la machine parallèle Verne de l'IRCCyN. Celle-ci est équipée du directeur de commande numérique Siemens Sinumerik 840D. Le protocole expérimental a consisté en l'usinage des quatre pièces test en étaux dans des bruts aluminium. Une fraise torique en carbure monobloc de diamètre 20mm a été utilisée dans le cas du premier vidage de poche (courbe guide contour parallèle seul). Pour les trois usinages suivants, l'outil est une fraise deux tailles carbure monobloc de diamètre 10mm. Les conditions de coupe n'ont pas été optimisées dans notre cas et ont été choisies de telle manière que l'usinage se déroule dans de bonnes conditions, avec une certaine marge de sécurité vis-à-vis des possibilités de la machine et des outils. La profondeur de la poche à vider est de 9mm réalisée en 3 passes de 3 mm. Les résultats sont les suivants (Tableau 6) :

Stratégie – Courbe guide	Paramètres	Visualisation
Contour parallèle seul	<ul style="list-style-type: none"> - R_trochoïdal = 14 mm - R_outil = 10 mm - Fz = 0,2 mm/dt - Z = 2 dents - N = 24000 tr/min - F = 9,6 m/min 	
Contour parallèle + milieu	<ul style="list-style-type: none"> - R_trochoïdal = 12 mm - R_outil = 5 mm - Fz = 0,2 mm/dt - Z = 2 dents - N = 24000 tr/min - F = 9,6 m/min 	
Zig-zag	<ul style="list-style-type: none"> - R_trochoïdal = 10 mm - R_outil = 5 mm - Fz = 0,2 mm/dt - Z = 2 dents - N = 24000 tr/min - F = 9,6 m/min 	


<p>Zig-zag Poche quelconque</p>	<ul style="list-style-type: none"> - R_{trochoïdal} = 10 mm - R_{outil} = 5 mm - Fz = 0,2 mm/dt - Z = 2 dents - N = 24000 tr/min - F = 9,6 m/min 	
---	--	--

Tableau 6 : Résultats de l'usinage des 4 pièces test

7.4. Conclusion

Nous avons mis en œuvre dans cette troisième étape de notre démarche les outils que nous avons développés pour pouvoir réaliser l'ensemble de la chaîne numérique Step_NC, de la pièce CAO jusqu'à la pièce réelle. Nous avons pu nous rendre compte de la simplicité de modification qui peut être apportée par le standard Step_NC dans le cas des trajectoires à motifs, et en particulier au pied de la machine. Prenons par exemple le cas de la modification d'un rayon d'outil au pied de la machine. Il suffit alors simplement de modifier un paramètre (en l'occurrence le rayon d'outil) dans le fichier Step_NC pour que l'ensemble des trajectoires soient modifiées en conséquence, directement dans la CN, pour pouvoir usiner immédiatement. Nous pouvons raisonner ainsi pour l'ensemble des paramètres d'usinage ainsi que pour la géométrie de la pièce et des entités d'usinage. Ces modifications deviennent simples au niveau du Step_NC et sont traitées par la suite dans la CN vers laquelle est « délocalisée une partie de l'intelligence ». Rappelons qu'avec le code G, la moindre modification d'un paramètre important entraîne la modification complète des trajectoires et donc du fichier ISO.

L'aspect que nous avons mis en avant ici et qui est une propriété majeure du format Step_NC est un peu occultée dans notre démarche dans la mesure où nous sommes contraints d'interpréter nos trajectoires en code G au lieu d'envoyer directement l'information aux cartes d'axes et autres éléments de la machine. Nous sommes malheureusement contraints d'utiliser cette méthode car nous n'avons pas encore à disposition de CN adaptée au format Step_NC. Pour avoir une réelle vision d'ensemble de ce que pourrait être une chaîne numérique Step_NC, nous pourrions étendre à l'intérieur du directeur de commande numérique les tâches de lecture du fichier Step_NC, de génération des trajectoires et de traitement de l'usinage. Cela permettrait, ce qui est le but à terme, de se dispenser totalement du code G ainsi que de tous ces inconvénients.

8. Conclusion, intérêt des travaux réalisés et perspectives

8.1. Les travaux réalisés

Les travaux que nous avons menés ont pour but de mettre en avant la faisabilité de l'implantation de l'usinage trochoïdal et du tréflage dans le format Step_NC. Tout d'abord, notre démarche a consisté à nous appuyer sur une construction simple et efficace des trajectoires avec l'utilisation des courbes à motifs. Pour cela, nous répétons un motif (une trochoïde ou une plongée) le long d'une courbe guide. L'ensemble des paramètres et attributs nécessaires à la construction des courbes à motifs a pu être défini dans le but d'être intégré au fichier Step_NC. Dans cette étape, nous avons implanté l'usinage trochoïdal et le tréflage dans le logiciel coréen PosSFP qui, à partir du fichier CAO de la pièce, reconnaît les entités d'usinage et génère le fichier Step_NC. Nous avons proposé une syntaxe de description des paramètres de la stratégie dans le fichier Step_NC en nous basant les stratégies déjà présentes dans la norme. Le choix de la courbe guide s'appuie lui-même sur les stratégies déjà normalisées. A travers l'implantation dans le logiciel PosSFP, nous avons proposé une manière d'intégrer l'usinage trochoïdal et le tréflage au fichier Step_NC.

La seconde étape de notre démarche a consisté à développer un ensemble d'outils permettant de lire et d'exploiter le fichier Step_NC en vue de l'usinage de la pièce dans le cas particulier de l'usinage trochoïdal. Un programme permet ainsi de lire le fichier Step_NC, d'en extraire la géométrie des entités ainsi que les paramètres des stratégies et de construire les trajectoires outil. Ces dernières peuvent alors être visualisées et exportées sous forme d'un fichier de points pour permettre l'usinage de la pièce.

L'ensemble de cette chaîne numérique basée sur le Step_NC a été mise en œuvre par l'intermédiaire du traitement de quatre pièces test, de la maquette CAO jusqu'à la pièce usinée. L'usinage de ces quatre pièces test a pour but de montrer les possibilités des différentes stratégies d'usinage trochoïdal implémentées.

8.2. Intérêts de ces travaux dans la recherche sur le format Step_NC

Les travaux que nous avons effectués s'inscrivent dans une démarche de développement du format Step_NC, alternative à l'utilisation du code G dont nous avons souligné les défauts. C'est donc un format naissant dont les preuves restent à faire au sein de l'entreprise malgré des perspectives prometteuses. De grands travaux de recherche restent aussi à mener pour son développement, son amélioration et son intégration, tout d'abord sur le terrain mais aussi dans les esprits. Dans ce cadre, l'intérêt principal de notre travail se situe dans les points suivants :

- Nous avons tout d'abord montré un prototype de chaîne numérique Step_NC complète, ainsi que le traitement de plusieurs pièces jusqu'à l'usinage sur machine industrielle. Cela nous a aussi permis de soulever la nécessité et l'attente que nous avons de voir des

directeurs de commande numérique compatibles avec le Step_NC. En effet, cela représente réellement un frein au développement de ce format de devoir « interpréter » les trajectoires en code G pour pouvoir usiner sur des machines industrielles. Nous sommes cependant pour l'instant contraints d'utiliser cette méthode.

- L'implantation de l'usinage trochoïdal et du tréflage enrichit les possibilités du Step_NC par des nouvelles stratégies de vidage de poche. Ces stratégies, même si elles sont peu présentes dans les logiciels de FAO actuels, dévoilent des intérêts souvent méconnus en termes de productivité, sans doute parce que se sont des stratégies relativement nouvelles. Cela n'a que plus d'intérêt pour le Step_NC, dans la mesure où ses possibilités en termes de choix de stratégies ne se restreignent plus seulement aux stratégies « classiques » mais offrent d'autres alternatives parfois mieux adaptées.
- La génération des trajectoires outils par l'utilisation de courbes à motifs comme dans le cas de l'usinage trochoïdal et du tréflage a ouvert de nouvelles méthodes de construction et de description des stratégies tout à fait dans l'esprit du Step_NC. La rapidité de définition des stratégies, la simplicité des modifications s'il y a lieu, leur construction par répétition d'un motif en font des stratégies intéressantes et correspondantes aux critères du Step_NC.
- Les travaux qui ont été présentés ici font partie des axes développés par l'équipe à l'ISO et pourra contribuer à une éventuelle normalisation de l'implantation des stratégies à motifs d'usinage trochoïdal et tréflage.

8.3. Les perspectives

Il reste de vastes domaines de recherche pour faire du standard Step_NC un outil performant au service de l'entreprise. Nous pouvons d'ores et déjà distinguer les travaux qui, au sein de notre équipe et de nos thématiques de recherche, pourraient contribuer à cet effort et de façon plus générale à la programmation avancée des machines à commande numérique.

En effet, des travaux ont déjà été menés pour l'intégration des mesures temps réel en vue de la correction des trajectoires d'usinage en fonction des efforts de coupe. L'intégration à la chaîne numérique Step_NC pourrait constituer une avancée dans le domaine de la programmation « intelligente » des machines outils. La programmation Step_NC offre de nouvelles possibilités dans la génération et correction temps réel des trajectoires d'usinage. Le principe sur lequel il se base permet de déléguer suffisamment de liberté à la commande numérique pour qu'elle puisse adapter l'usinage en fonction des phénomènes extérieurs mis en jeu.

Ce format permet aussi d'intégrer plus facilement les résultats de simulation qui ont pu avoir lieu. Ainsi, nous pourrions intégrer directement les résultats de simulation d'efforts de coupe

et d'avance réelle pour la génération des trajectoires. Nous avons à disposition un simulateur mis au point à l'IRCCyN, dont les résultats pourraient être mis à contribution.

Concernant plus particulièrement les stratégies d'usinage trochoïdal et de tréflage, certains points restent à développer, comme le choix des paramètres des stratégies qui reste relativement difficile pour un utilisateur non spécialiste. Dans le cas de l'usinage trochoïdal, des études sont en cours à l'IRCCyN pour déterminer les associations de paramètres permettant un engagement radial optimisé. Dans une prochaine étape, celles-ci pourront être étendues à la prise en compte de la géométrie de la poche.

Références

- [**Albert 01**] : Albert, M., *Feature recognition—the missing link to automated CAM*. *Modern Machine Shop*, 2001a (April), 1–4. Albert, M., STEP NC—the end of G-Codes? *MMS Online*, 2001 (November).
- [**Allen & al. 05**] : Allen, R.D., Harding, J.A. and Newman, S.T., *The application of STEP-NC using agent-based process planning*. *Int. J. Prod. Res.*, 2005, 43(4), 655–670.
- [**Altintas 00**] : ALTINTAS Y., *Modeling approaches and software for predicting the performance of milling operations at MAL-UBC*, *Int. J. of Machining Science and Technology*, Volume 3-4, pp 445-478, 2000
- [**Anonyme 03**] : Anonymous, *STEP-NC demonstrated at JPL*. *Manuf. Eng.*, 2003, 130(3), 34–35.
- [**Bouzaki & all 03**] : Bousakis K.D., Aichouh P., Efstathiou K., *Determination of the chip geometry, cutting force and roughness in free form surfaces finishing milling, with ball end tools*, *Int. J. of Machine Tools and Manufacture*, Volume 43, pp 499-514, 2003
- [**Brooks & Greenway 95**] : Brooks, S.L. and Greenway Jr., R.B., *Using STEP to integrate design features with manufacturing features*, in *ASME Computers in Engineering Conference*, edited by A.A. Busnaina, pp. 579–586, 17–20 September 1995, Boston, MA, New York State 10017.
- [**Denkena & all 02**] : Denkena, H.K., Tonshoff, J., Selle, A., Storr, S., Heusinger, S. and Rogers, G., 2002, « *Offline-Berechnung der Zerspankräfte in der NC-Programmierung. Vorhersage der Zerspankräfte beim HSC-Schlichtfrasen* ».
- [**Dugas, Lee, Hascoët 01**] : A. Dugas, J.J. Lee et J.Y. Hascoët, *Feed Rate and Tracking Errors Simulation in High Speed Milling*, *Int. Conf. ESAFORM*, Liege, Belgium, Avril, 2001.
- [**Dugas 02a**] : A. Dugas, CFAO et UGV- *Simulation d'usinage de formes complexes*, Thèse de doctorat – Université de Nantes / Ecole Centrale de Nantes, Nantes (France), Décembre 2002
- [**Dugas 02b**] : Dugas A., Lee J.J., Terrier M., Hascoët J.Y., *Virtual manufacturing for High Speed Milling*, 35th CIRP International Seminar on Manufacturing Systems, pp 199- 205, Séoul (Corée), Mai 2002
- [**Feeney & all 03**] : Feeney, A.B., Kramer, T., Proctor, F., Hardwick, M. and Loffredo, D., *STEP-NC Implementation—ARM or AIM? ISO TC 184/SC4 WG3 N1223*, March, 2003.
- [**Hardwick 01**] : Martin Hardwick, “e-Manufacturing using Step_NC”, STEP Tools Inc., 2001

[Hascöet & al. 01a] : Hascoët J.Y., Dugas A., Lee J.J., *Rapport Final – Tâche III – Simulation des opérations d’usinage, Rapport final du projet « Saut Technologique UGV »*, pp 161p, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN) (France), Juin 2001

[Hascoet & al. 06a] : JY Hascoet, P. Mognol, M. Rivette, M. Rauch. « *Step_NC applications for milling and Rapid Prototyping* », ISO TC 184/SC4 Meetings – Toulouse – Juin 2006

[Hascoet & al. 06b] : JY Hascoet, P. Mognol, M. Rivette, M. Rauch. « *Milling and Rapid Prototyping : proposal for Step_NC* », 50th ISO TC 184/SC4 Meetings – Pennsylvanie - Octobre 2006

[Hascoet, Rauch 06c] : J.Y. Hascoët, M. Rauch, “*A new approach of the tool path generation in manufacturing operations using CNC data*”, Conférence HSM, 2006

[Hascoet, Rauch, Suh 07] : J.Y. Hascoët, M. Rauch, Suk Hwan Suh " *Relevance of Step-Nc standard for high level toolpaths generation*" HSM conference, San sebastian, Mars 2007

[Hurco] : <http://www.hurco.fr>

[Komanduri & al. 82] : Komanduri R., Schroeder T., Hazra J., Turkovich (von) B.F., Flom D.G. – *On the catastrophic Shear Instability in High-Speed Machining of an AISI 4340 Steel* – Journal of Engineering for Industry – Vol.104, p.121-131, may 1982.

[Kramer & al. 01] : Kramer, T.R., Huang, H., Messina, E., Proctor, F.M. and Scott, H., *A feature-based inspection and machining system*. Comput.-Aid. Design, 2001, 33, 653–659.

[Kramer & al. 04] : Kramer, T.R., Horst, J., Huang, H., Messina, E., Proctor, F.M. and Scott, H. *Feature-Based Inspection and control System*, NISTIR 7098, 2004 (National Institute of Standards and Technology: Gaithersburg, MD).

[Malinc] : <http://www.malinc.com/>

[Mazak] : <http://www.mazak.com/>

[Mori] : <http://www.moriseiki.com/>

[Newman & al. 03] : Newman, S.T., Allen, R.D. and Rosso Jr. R.S.U., *CAD/CAM solutions for STEP-compliant CNC manufacture*. Int. J. Comput. Integr. Manuf., 2003, 16(7–8), 590–597.

[Rauch Hascoët 06] : M. Rauch, J.Y. Hascoët, “ *Trochoïdal and plugging strategies in rough pocket milling of aluminium alloys*“, 5ème congrès international UGV, Metz, Mars 2006

[Storr & Heussinger 02] : Storr, S., Heussinger, S., 2002, *STEP-NC—Grundlage einer AD/NC-Prozesskette Das STEPNC-Prozessmodell für die Drehbearbeitung*.

[Storr & All 02] : Storr, A., Pritschow, G., Heussinger, S. and Azotov, A., *Workingstep planning for turning with STEP-NC: planning methods for user support*. IWF Zeitschrift für Wirtschaftlichen Fabrikbetrieb, 2002, 97(7–8), 390.

[Suh & al. 02a] : Suh, S.H., Cho, J.H. and Hong, H.D., *On the architecture of intelligent STEP-compliant CNC*. Int. J. Comput. Integr. Manuf., 2002a, 15, 168–177.

[Suh & al. 02b] :Suh, S.H., Chung, D.H., Lee, B.E., Cho, J.H., Cheon, S.U., Hong, H.D. and Lee, H.S., *Developing an integrated STEP-Compliant CNC prototype*. J. Manuf. Syst., 2002b, 21(5), 350–362.

[Suh & al. 03] : Suh, S.H., Lee, B.E, Chung, D.H. and Cheon, S.U., *Architecture and implementation of a shop-floor programming system for STEP-compliant CNC*. Comput. Aid. Design, 2003, 35, 1069–1083.

[Suh, Cho, Hascoet 96] : S.H. Suh and J.H. Cho, J.Y. Hascoet, "*Incorporation of Tool Deflection in Tool Path Computation: Simulation and Analysis*," SME J. of Manufacturing Systems, Vol. 15, No. 5, 1996.

[Terrier 05] : M.Terrier, "*Optimisation du processus de fabrication en Usinage UGV sur machine outil à structure parallèle*", Thèse de doctorat d'université et Ecole Centrale de Nantes, Sept. 2005.

[Tönshoff & all 98] : Tönshoff H.K., Karpuschewski B., Lapp C., Andrae P. – *New lachine techniques for High-Speed Machining* – Proceedings of the Int. Seminar on Improving Machine Tool Performance – vol. n°1, p.65-76, july 1998.

[Weck & al. 01] : Weck, M., Wolf, J. and Kiritsis, D., *STEP-NC—The STEP compliant NC programming interface evaluation and improvement of the modern interface*, in Proceedings of the ISM Project Forum, Genf, October, Monte Verita/Ascona Schweiz, 2001.

[Weck & al. 03] : Weck, M. and Wolf, J., *ISO 14649 provides information for sophisticated and flexible numerically controlled production*, *Production Engineering—Research and Development*. Annals of the German Academic Society for production, 2003, 2, 1–5.

[Wolf & al. 01] : Wolf, J., *STEP-NC—integrating shop floor into industrial data flow for the enabling of intelligent near to process functions*, Korea–Germany Workshop on STEP-NC, 23 May 2001.

[Xu 05] : X.W.Xu & all. *Step_compliant NC research : the search for intelligent CAD/CAPP/CAM/CNC integration* » Int. Jour. of Prod. Reasearch, Sept. 2005

[Xu & He 03] : Xu, X. and He, Q., *Striving for a total integration of CAD, CAPP, CAM and NC*. Robot. Comput. Integr. Manuf., 2003, 20(2), 101–109.

[Xu 05b] : X.W. Xu « *Realization of STEP-NC enabled machining* », Robotics and Computer-Integrated Manufacturing 22 (2006) 144–153, Fevrier 2005