

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE
DEPARTEMENT GENIE MECANIQUE



DOMAINE : SCIENCES TECHNIQUES
FILIERE : GENIE MECANIQUE
OPTION : GENIE DES MATERIAUX

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: ZEDAME Ahmed Said

Intitulé

**ETUDE ET REALISATION D'UNE PIECE SUR
MACHINE A COMMANDE NUMERIQUE
TYPE F1 CNC EMCO**

Soutenu le 4/07 /2019 devant le jury composé de:

ZAOUI	Université de M'sila	Président
AMROUNE Salah	Université de M'sila	Rapporteur
BELHOCINE	Université de M'sila	Examineur

Année universitaire : 2018 /2019



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



Remerciements

D'abord je remercie Allah de m'avoir donné le courage et la détermination nécessaire pour finaliser ce travail.

Et n'oublions pas mes parents qui apportent tout le soutien nécessaire dans notre parcours scolaire.

ma femme pour son énorme soutien dans l'élaboration de ce travail.

Je tiens aussi à remercier Mon encadreur enseignant : Mr. AMROUNE Salah Pour son entière disponibilité, ses conseils et ses éclaircissements qui m'ont tant servi à réaliser ce travail.

Je remercie aussi Mr SILEM mourad responsable laboratoire.

Mes sincères remerciements vont au président et aux membres de jury de soutenance qui m'ont fait l'honneur d'examiner ce travail.

A tous ceux qui, de près ou de loin, ont œuvré pour l'aboutissement de ce travail et de ma formation, et dont les noms n'ont pas été mentionnés ici, qu'ils veuillent bien accepter l'expression de ma profonde reconnaissance.

DEDICACES

Je dédie ce modeste travail :

A Ma très chère Mère qui a toujours été à mes côtés

A Mon Père qui m'a toujours guidé et soutenu

A ma femme et ma fille rinade

A mes frères et sœurs pour leur soutien moral

A tous mes amis

A tous ceux qui m'ont aidé de près ou de loin

Sommaire

Remerciement	
Dédicace	
INTRODUCTION GENERALE	1
CHAPITRE I:	2
I- NOTIONS SUR LES MACHINES OUTILSA COMMANDE NUMERIQUE	
I.1 Introduction	2
I.2 Historique	2
I.3 Définitions et structure d'une machine-outil à commande numérique	4
I.3.1 Définition	4
I.3.2 Structure d'une machine-outil à commande numérique	5
I.3.2.1 Partie commande	6
I.3.2.2 Partie opérative	6
I.4 Origines	7
I.5 Systèmes d'axes	8
I.6 Types des MOCN	9
I.7 Décalage et géométrie d'outil	12
I.7.1. Décalage de l'origine machine	13
I.7.2 Géométrie des outils	14
I.7.3 Décalage d'origine (G59)	16
I.8 Programmation des MOCN	17
I.8.1 Organisation et structure d'un programme CNC	17
I.8.2 Langage et programmation CNC	18
I.8.3 Décalage d'Origine réglable	19
I.8.4 Décalage d'Origine Programmable TRANS / ATRANS	20
I.8.5 Les principales fonctions	20
II- CHEMINEMENT DE L'INFORMATION CONCEPT- MOCN	23
II.1 Cheminement Concept- CAO	24
II.2 Cheminement CAO-FAO	26
II.3 Cheminement FAO- Post-Processeur	28
II.4 Cheminement Post-Processeur - MOCN	29
II. 5 Module de simulation d'usinage	31
II.6 Module de la MOCN	32
II.7 Module FAO	32
II.8 Transfert d'informations FAO- Post-processeur	33
II.8-1 Module Post-processeur	33
II-8-2 Module MOCN	34
CHAPITRE II: PROGRAMMATION PARAMETRIQUE CNC	36
I-INTRODUCTION	36

II-OPERATIONS ARITHMETIQUES	37
III-OPERATIONS LOGIQUES	38
IV-CONCLUSION	55
CHAPITRE III : PARTIE EXPERIMENTALE	56
I-INTRODUCTION	56
II-PRESENTATION DE LA PIECE	56
III. LANCEMENT DE PROGRAMME SUR WincTs	56
III.1 Gestion et création de programmes	56
IV-REALISATION DE LA PIECE SUR LA MACHINE CNC EMCO TYPE F1	64
V- CONCLUSION	68
CONCLUSION GENERALE	69
REFERENCES BIBLIOGRAPHIQUE	70

Les machines-outils à commande numérique permettent d'augmenter la productivité tout en assurant une meilleure qualité des produits. De plus, avec le développement des logiciels de Conception Assisté par Ordinateur (C.A.O) et de Fabrication Assisté par Ordinateur (F.A.O).

La fabrication de la pièce se fait par enlèvement de copeau sur les machines-outils conventionnelles. Cette dernière prend beaucoup de temps, et demande une grande expérience des opérateurs pour arriver à une qualité (précision) d'une pièce juste moyenne. Par contre en utilisant une machine-outil à commande numérique, l'usinage est beaucoup plus précis et ne demande pas beaucoup de temps.

Dans ce contexte, ce travail a pour but de faire une simulation d'usinage d'une pièce sous les logiciels MACHMILL et la réalisation sur la machine EMCO-CNC type F1. Cette étude comprend trois chapitres :

Le premier chapitre est consacré à la présentation des notions sur les machines-outils à commande numérique MOCN. Un aperçu général sur la technologie des MOCN, citant la classification, l'architecture, programmation des MOCN pour différentes commandes.

Ensuite un deuxième chapitre présentant la programmation paramétrique et leurs exploitations.

Quand au troisième chapitre, il relate en détail les étapes de la réalisation d'une pièce sous les logiciels cités ci- dessus.

11.1 Introduction

Apparue il y a seulement quelques dizaines d'années, la commande numérique (CN) impose actuellement sa technologie dans le monde de l'usinage. Conçue Pour piloter le fonctionnement d'une machine à partir des instructions d'un programme sans intervention directe de l'opérateur pendant son exécution, elle a dans un premier temps, permis de franchir un pas important dans l'automatisation des machines-outils traditionnelles. Tours, fraiseuses, perceuses et aléseuses sont ainsi devenues capables d'assurer, en quantité comme en qualité, une production à peine imaginable quelques années auparavant. La CN est également à l'origine de nouvelles conceptions des machines polyvalentes comme le centre d'usinage, par exemples. [1]

Dans ce chapitre on présente un aperçu général sur la technologie des MOCN, citant la Classification et l'architecture des MOCN.

1.2 Historique

Les travaux menés par Falcon et jacquard à la fin du XVII siècle ont montré qu'il était possible de commander les mouvements d'une machine à partir d'informations transmises par un carton perforé. Leur métier à tisser de 1805 fut le premier équipement à être doté de cette technique et, de ce point de vue, il peut être considéré comme l'ancêtre. Il faut cependant rattacher l'exploitation industrielle de la CN au développement de l'électronique.

En 1947, à Traverse City dans l'Etat du Michigan, John Parsons fabrique pour le compte de l'US air force dans pales d'hélicoptère par reproduction. Pour façonner ses gabarits, il utilise une méthode consistant à percer plusieurs centaines de trous faiblement espacés de manière à approcher le profil théorique. L'emplacement et la profondeur de chaque trou sont calculés avec précision par un ordinateur IBM à cartes perforées. La finition de la surface est obtenue par des opérations manuelles de polissage.

Mais, lorsque l'US Air Force confie à ce même Parsons la réalisation de pièces de formes encore plus complexes pour ses futurs avions supersoniques, celui-ci réalise que sa méthode est trop approximative et que seul un usinage continu en 3 dimension sera en mesure de donner satisfaction Au printemps 1949, il confie alors au Massachusetts Institute of Technologie (MIT) le soin de développer des asservissements capables de piloter une machine qui recevra des instructions intermittentes à partir d'un lecteur de cartes.

Cette machine, une fraiseuse prototype Cincinnati à broche verticale (**Fig I.1**). Conçue pour exécuter des déplacements simultanés suivant 3 axes, est officiellement présentée en

septembre 1952 dans le Servomechanisms laboratory du MIT. L'information mathématique étant la base du concept, on lui donne de numerical control. Il aurait tout aussi bien s'appeler commande symbolique



Figure I.1 : Premier MOCN en 1952 [1]

Il faut encore attendre quelques années de vastes fonds l'US Air Force et l'appui des chercheurs du MIT pour rendre la première Machine-outil à commande numérique (MOCN) réellement opérationnelle.

- Les différentes étapes de développement de la CN sont les suivantes :

1954 : Bendix acquiert le brevet de Parsons et fabrique la première CN industrielle.

1955 : à Font de Lac (Wisconsin), le constructeur américain Giddins & Lewis commercialise la première MOCN.

1959 : apparition de la NC en Europe (foire de Hanovre). Le MIT annonce la création du langage de programmation APT (Automatic Programed Tools).

1960 : apparition du système DNC (Direct Numerical Control)

1964 : en France, la télémécanique Électrique lance la CN NUM 100 conçue à base de relais Téléstatic.

1968 : la CN adopte les circuits intégrés ; elle devient plus compacte et plus puissante.

Le premier centre d'usinage est mis en vente par Kearney & Trecker (USA).

1972 : les minicalculateurs remplacent les logiques câblées ; la CN devient CNC.

1976 : développement des CN à microprocesseurs.

1984 : apparition de fonctions graphiques évoluées et du mode de programmation conversationnel, début de la fabrication assistée par ordinateur (FAO).

1986 : les CN s'intègrent dans les réseaux de communication, début de l'ère de la fabrication flexible (CIM : computer integrated manufacturing).

1990 : développement des CN à microprocesseurs 31 bits (**Fig1.2**).

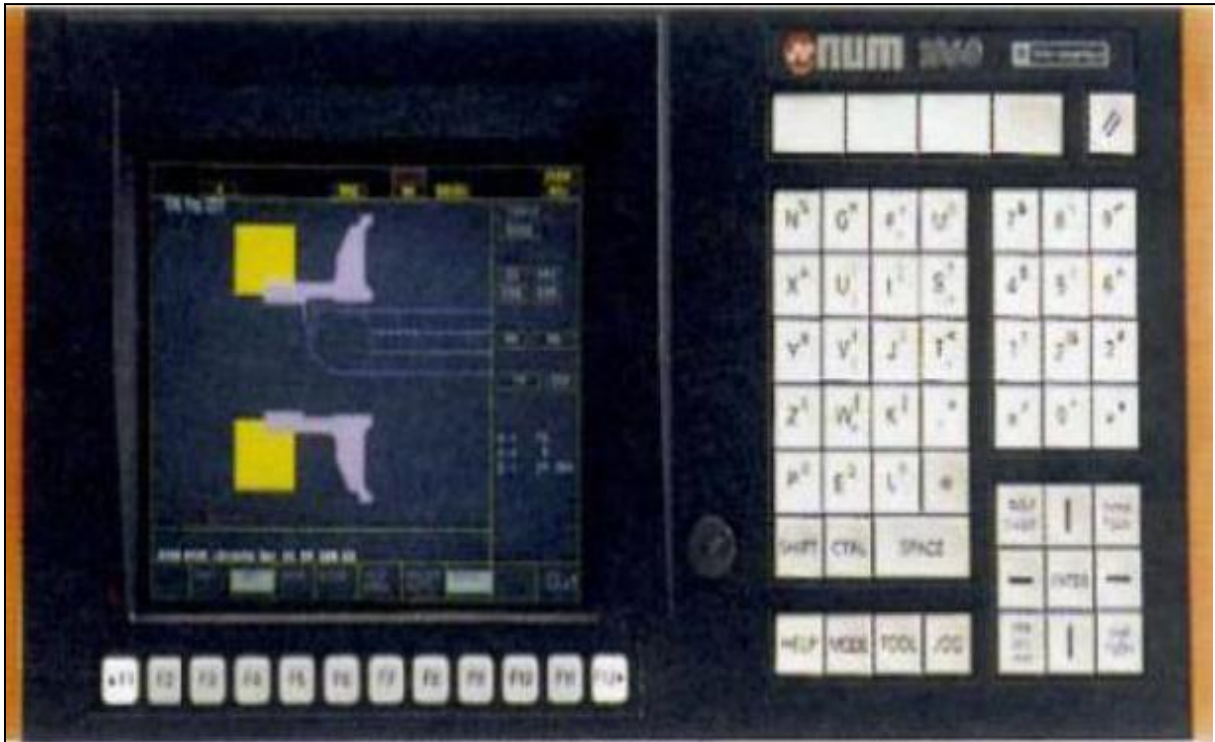


Figure I.2 : Pupitre opérateur d'une CN 32 bits [1]

I.3 Définitions et structure d'une machine-outil à commande numérique

I.3.1 Définition

La MOCN est une machine totalement ou partiellement automatique à laquelle les ordres sont communiqués grâce à des codes dans un programme CN. Lorsque la machine-outil est équipée d'une commande numérique capable de réaliser les calculs des coordonnées des points définissant une trajectoire (interpolation), on dit qu'elle est à calculateur. Elle est appelée Commande Numérique par Calculateur (CNC). La plupart des MOCN sont des CNC. La figure ci-dessous montre une machine à commande numérique de type BOXFORD. [2]



Figure I.3 : MOCN de type BOXFORD (FANUC). [2]

I.3.2 Structure d'une machine-outil à commande numérique

Une machine-outil à commande numérique est composée de deux principales parties :

- ❖ Partie commande
- ❖ Partie opérative

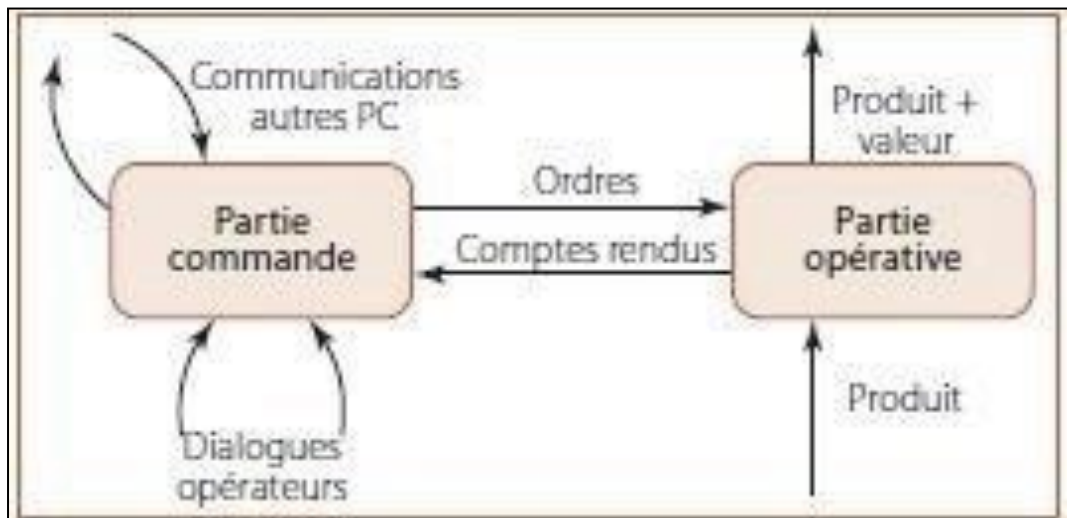


Figure I.4 : Structure d'une M.O.C.N [2]

I.3.2.1 Partie commande

Différente d'une machine conventionnelle, elle est constituée d'une armoire dans laquelle on trouve :

- ❖ Le pupitre permettant de rentrer les commandes à l'aide d'un clavier,
- ❖ Le lecteur de données (ce lecteur peut être une option lors de l'achat de la machine),
- ❖ La sortie RS 232 pour les liaisons avec les Périphériques externes,
- ❖ L'écran de visualisation de toutes les données enregistrées,
- ❖ Le calculateur,
- ❖ Les cartes électroniques (commandes d'axes, mémoire ...).

La partie commande est alimentée par un courant faible et ne peut donc pas alimenter en direct les moteurs de la machine (**Fig I.5**).

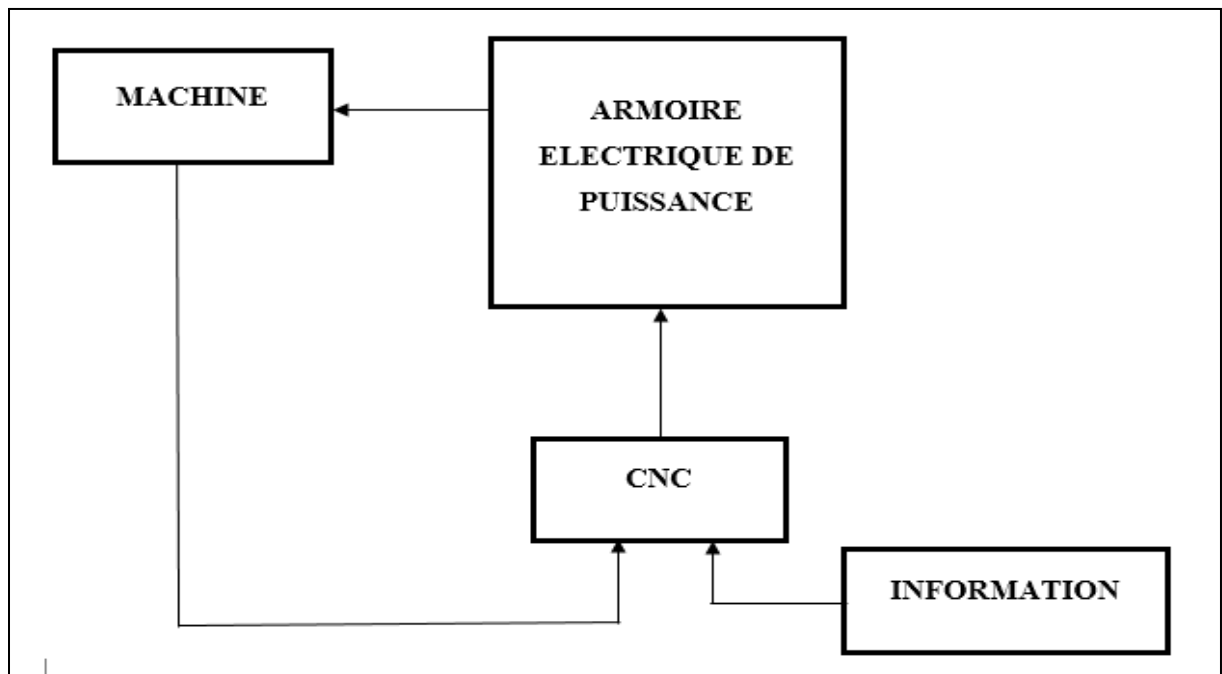


Figure I.5 : Fonction originale d'une commande numérique [2]

I.3.2.2 Partie opérative

Les mouvements sont commandés par des moteurs ; presque comparable à une machine-outil classique, et elle comprend :

- ❖ Un socle, très souvent en béton hydraulique vibré, assurant l'indépendance de la machine au sol,
- ❖ Un bâti, un banc, dont les larges glissières sont en acier traité,
- ❖ Un support outil (broche, torche, laser, jet d'eau ...),

- ❖ Une table support pièce, mobile selon 2 ou 3 axes, équipée de système de commande à vis et écrou à bille. Le granit, ou le granit reconstitué, est utilisé pour la fabrication des tables et des bâtis des machines à mesurer tridimensionnelles des rectifieuses et de certains tours,
- ❖ Des moteurs chargés de l'entraînement de la table,
- ❖ Un élément de mesure ou capteur de position renseignant à tout moment sur la position du mobile sur chaque axe,
- ❖ Une dynamo tachymétrique assurant la mesure de la vitesse de rotation

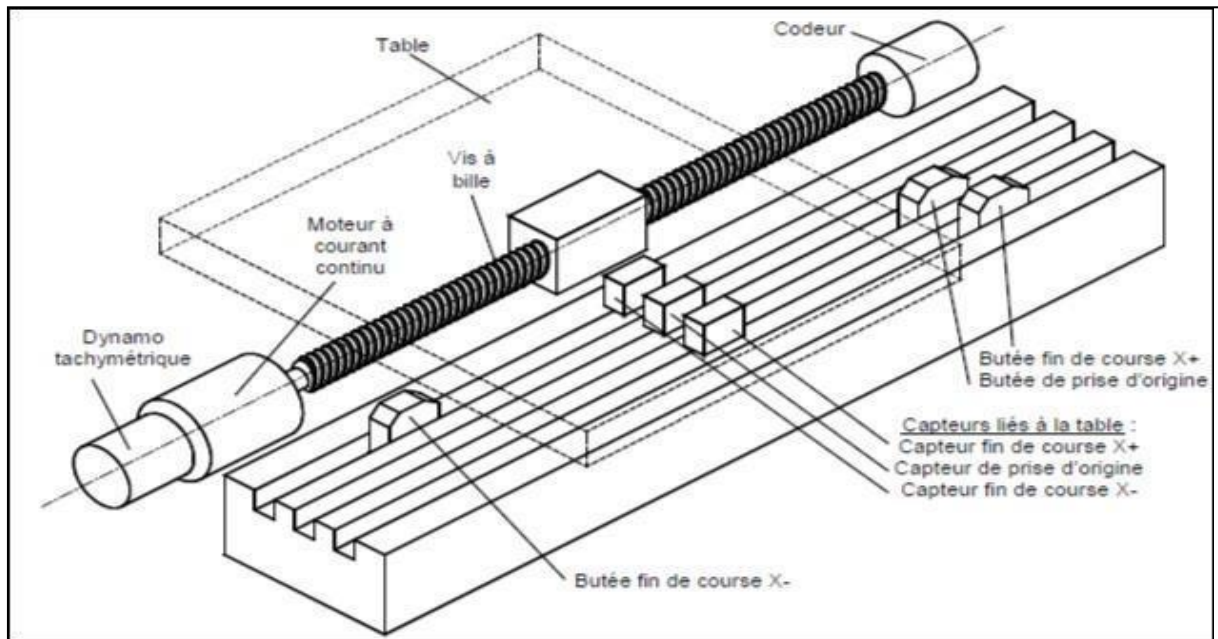






Figure I.6 : Eléments de la partie opérative [2]

I.4 Origines

Le tableau ci-dessous résume les différentes origines des MOCN

Tableau I.1 : Point utilisées dans des MOCN [2]

Points utilisées	Symbole	Définition
Origine machine M (OM)		C'est la référence des déplacements de la machine. C'est un point défini (sur chaque axe) par le constructeur qui permet de définir l'origine absolue de la machine. OM et om peuvent être confondues.
Origine mesure R (Om)		C'est le point de départ de toutes les mesures dans l'espace machine
Origine Programme OP		C'est le point de départ de toutes les cotes.
Origine Pièce W (Op)		origine de la mise en position (isostatique de la pièce)

La figure suivante représente la répartition des origines en tournage et en fraisage (**Fig I.7**)

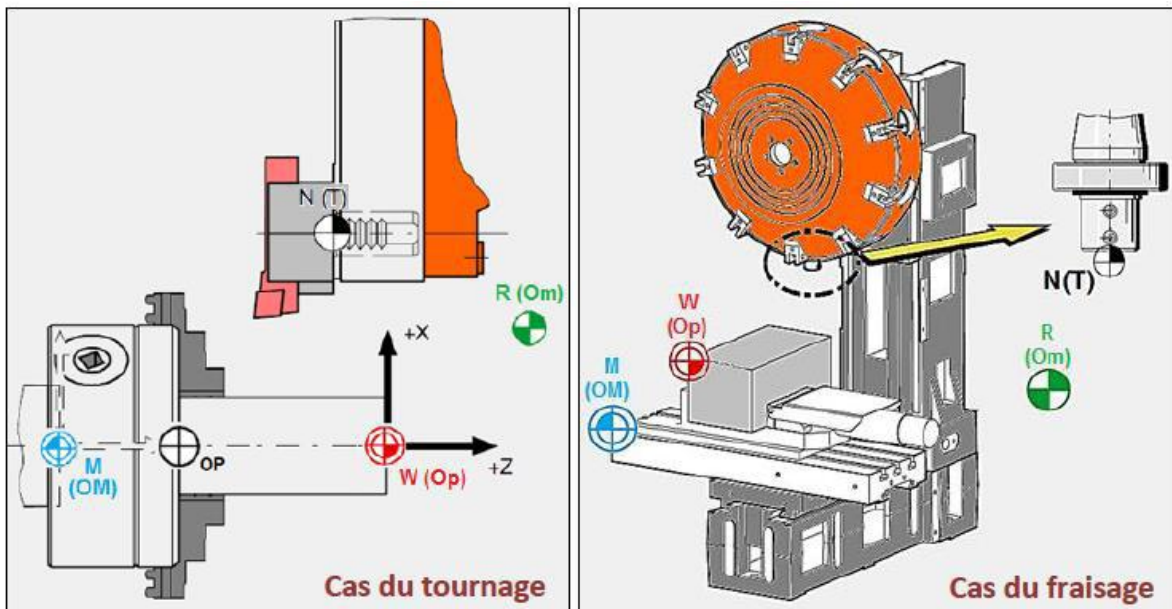


Figure I.7 : Cas du tournage et du fraisage [2]

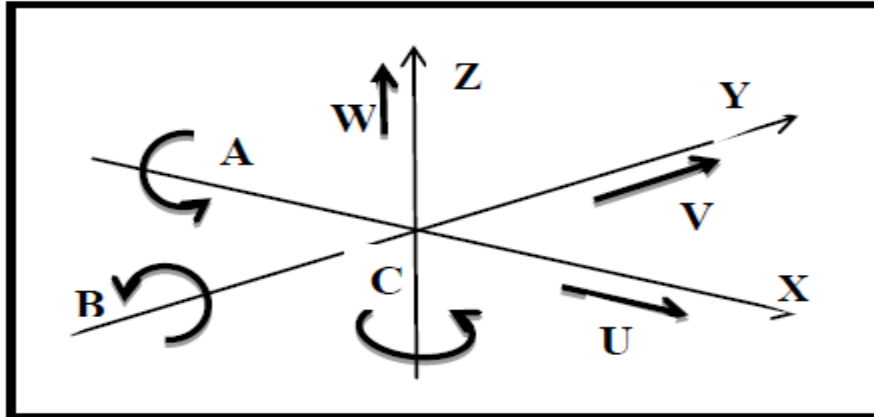
I.5 Systèmes d'axes

Les déplacements de l'outil ou du porte-pièce s'effectuent par combinaisons de translations et/ou de rotations.

Chaque mouvement élémentaire (axe) est repéré par une lettre affectée du signe + ou - indiquant le sens du déplacement.

- Les translations primaires s'effectuent suivant les axes **X, Y, Z** formant ainsi le trièdre de référence.

- Les rotations primaires sont les trois rotations **A, B, C** autour de ces trois axes. L'axe **Z** : est celui de la broche.
- Les translations primaires (**U, V, W**) sont parallèles à **X, Y** et **Z**.



Systèmes d'axes.

La figure (FigI.8) ci-dessous montre un exemple d'un centre d'usinage à cinq axes.

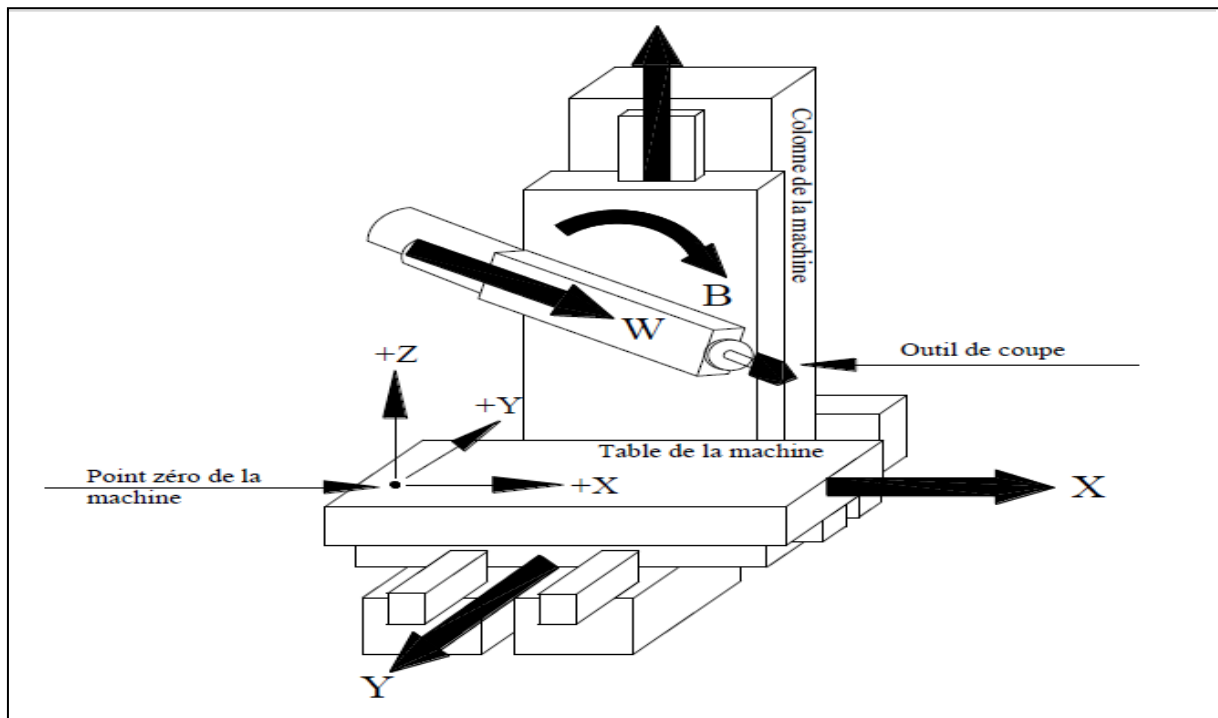


Figure I.8 : Fraisage à cinq axes [2]

I.6 Types des MOCN

Le classement des machines est nécessaire car il aide au choix de machines, lors d'étude de gammes de fabrication. Traditionnellement, on classe les machines en fonction des formes de surfaces à réaliser : cylindriques / parallélépipédiques, tournage / fraisage. Cette classification est remise en cause, car la commande numérique et l'adaptation des structures de machine

cassent le lien entre les deux couples. [3]

On classe maintenant les machines-outils par le nombre de mouvements élémentaires qu'elles peuvent mettre en oeuvre lors du déplacement de l'outil par rapport à la pièce. (**Tableau I.2**) Seuls les axes sont décomptés. La mise en oeuvre simultanée de plusieurs outils entraîne l'augmentation du nombre d'axes.

Cette classification ne permet pas d'associer directement un type de forme usinable à une classe de machine, car elle ne reflète pas la cinématique de l'outil. Par exemple un tour à cinq axes ne permet pas de faire des pièces différentes par rapport à un tour à trois axes.

Tableau I.2 : Classification des machines-outils [3]

Nb d'axes	Mouvements	type d'usinage et des opérations possibles
1	Z	brochage, presse
2	X, Z	tournage : toutes les formes obtenues ont le même axe de symétrie
3	X, Y, Z	Fraisage : surfacage, perçage, fraisage de poches, de rainures et de surfaces gauches. L'axe outil reste parallèle à une direction fixe par rapport à la pièce.
3	X, Z, C	Tournage, avec asservissement de la broche, permet le fraisage sur tour : tournage général, fraisage à l'outil tournant, perçage hors axe. La broche est asservie en position.
4	X, Y, Z, B	Fraisage : surfacage, perçage, fraisage de poches, de rainures et de surfaces gauches. L'axe outil reste contenu dans un plan fixe par rapport à la pièce.
4 fraisage	X, Y, Z, C	(cf. X, Y, Z, B)
4	X, Y, Z, C	tournage
4 tournage	2x(X,Z)	deux tourelles
5	2x(X,Z),C	deux tourelles et asservissement de la broche, (cf X, Z, C)
5	X, Y, Z, A, C	fraisage de formes gauches : fraisage avec le flanc de l'outil, fraisage avec dépinçage, perçage en toutes directions.
5	X, Y, Z, B, C	fraisage de formes gauches (cf. X, Y, Z, A, C)
5	X, Y, Z, A, B,	fraisage de formes gauches (cf. X, Y, Z, A, C)

De plus en plus, la broche se comporte comme un axe asservi. C'est le cas en tournage, où l'on peut être obligé de mettre en place une seconde motorisation lorsqu'elle génère un mouvement d'avance (tournage avec outil tournant).

En fraisage, la broche est plutôt un demi-axe, car on cherche à asservir sa vitesse de rotation à la vitesse d'avance de l'outil pour assurer, par exemple, des interpolations hélicoïdales (taroudage) ou de façon à garantir l'avance par dent de l'outil.

Les machines 5 axes du type (X, Y, Z, A, C) sont privilégiées dans le cadre de l'usinage de pièces de dimensions importantes (**Fig I.9**) et (**Fig I.10**) bien qu'elles perdent en rigidité au niveau de la tête de la machine.

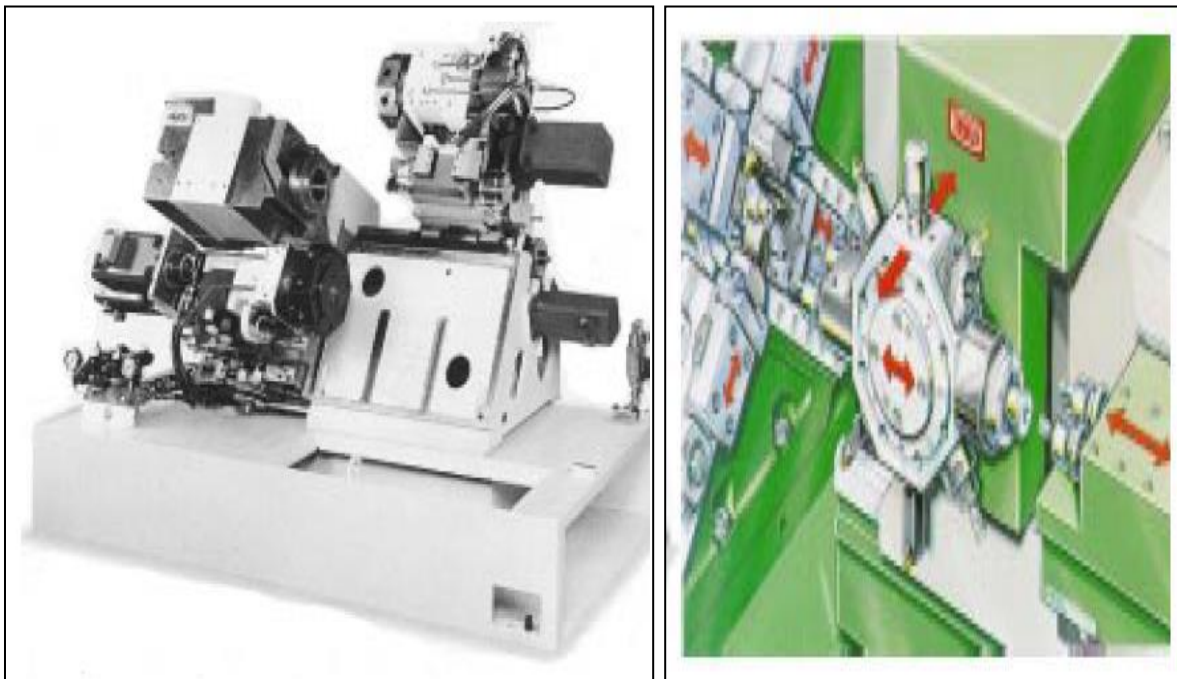


Figure I.9 : cinématique de tour multi-axes [3]

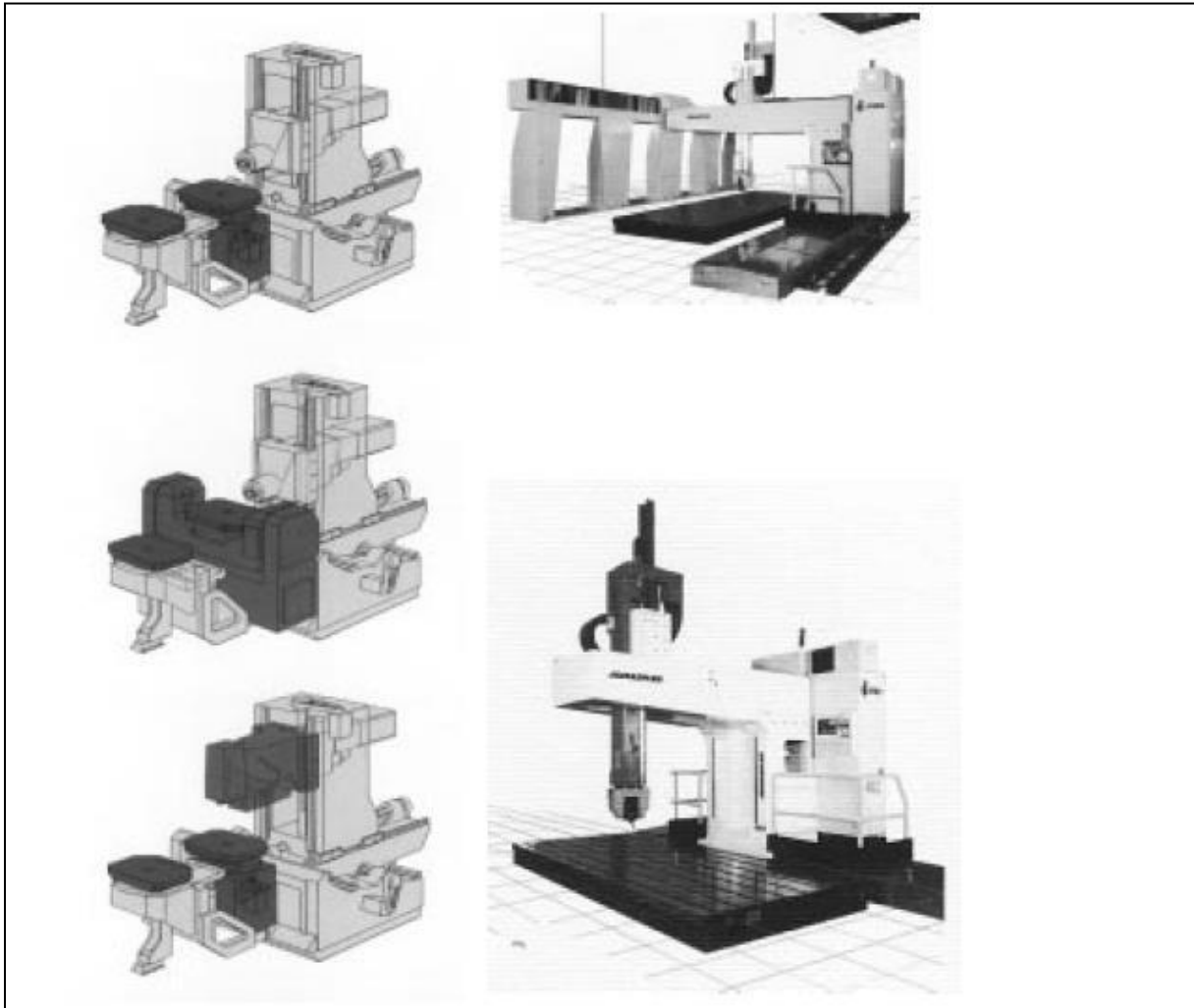


Figure I.10 : Structure des machines de 3 à 5 axes [3]

On peut classer les machines à plus de cinq axes en trois catégories :

- ✓ Les mouvements génèrent une forme particulière, ce sont les affuteuses (jusqu'à neuf axes), ou des machines spéciales ;
- ✓ Les mouvements sont dédoublés, ce sont les machines transferts ;
- ✓ Les mouvements ne sont pas dédoublés, ce sont les robots. La dissociation entre le robot et la machine-outil vient de la comparaison des performances respectives sur la rigidité, la vitesse de déplacement et la précision.

I.7 Décalage et géométrie d'outil

Dans l'espace de travail d'une MOCN. Sont définis différents points de référence.

Ces points sont nécessaires pour le pré-réglage et la programmation de la machine.

Nous présenterons les différents points ainsi que leur positionnement dans l'espace de travail

d'une MOCN. La figure.1.11, et figure.1.12, montre les points de référence dans le volume d'usinage. (**Fig 1-11**) (**Fig 1- 12**).

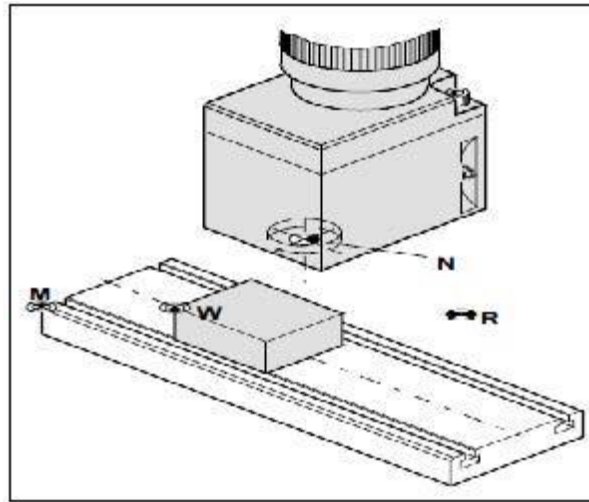


Figure I.11 : Point de référence dans le volume d'usinage dans le cas fraisage. [4]

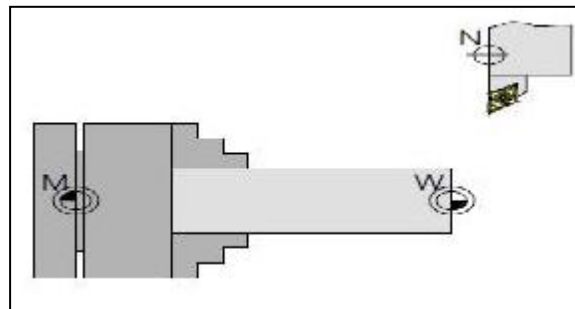


Figure I.12 : Point de référence dans le volume d'usinage dans le cas tournage. [4]

I.7.1. Décalage de l'origine machine

La position de point « M », l'origine machine, étant très éloignée de la pièce à usiner, ne convient pas en tant que point de départ de la programmation. Donc, il va falloir décaler l'origine machine vers un point qui facilite la programmation. Ce point est l'origine de la pièce (**Fig I.13**).

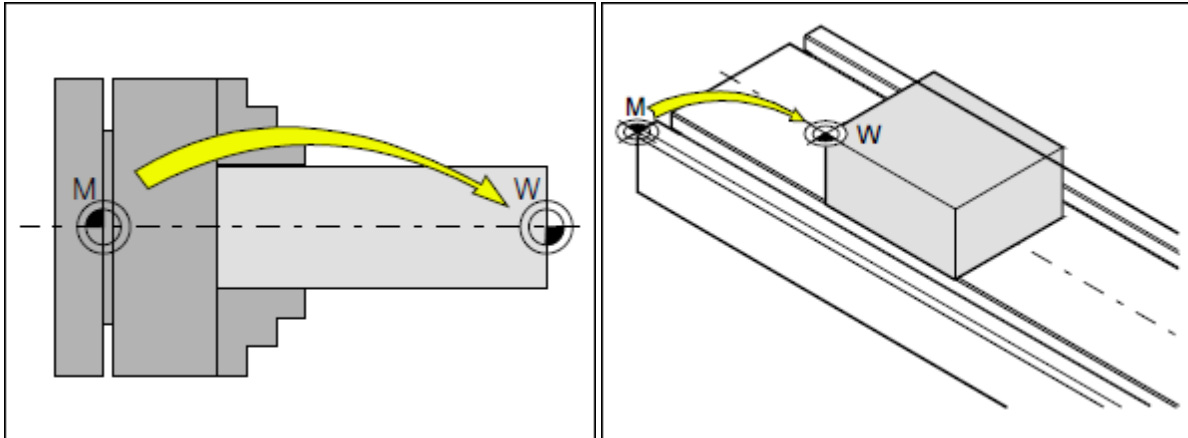


Figure I.13 : Décalage d'origine. [4]

I.7.2 Géométrie des outils

La saisie des données de l'outil est nécessaire pour que le logiciel utilise la pointe de l'outil (cas de tournage) ou le centre de l'outil (cas de fraisage) pour le positionnement, et non pas le point de référence du logement de l'outil « F ».

Chaque outil utilisé pour l'usinage doit être mesuré (**FigI.14**). Il s'agit ici de calculer l'écart entre le point de référence du logement de l'outil et la pointe respective de l'outil. Une fois que les données sont déterminées. Nous les mémorisons dans le registre des outils. Les données spécifiques à l'outil diffèrent d'un type d'outil à un autre.

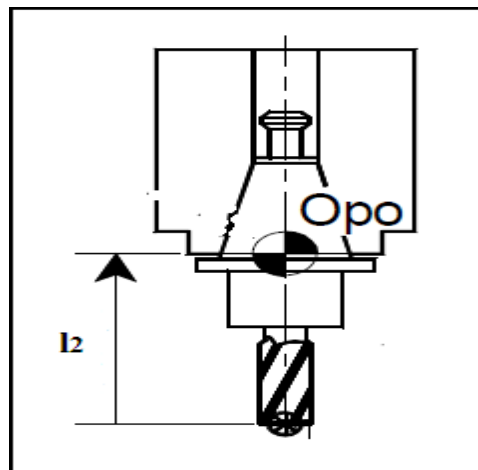


Figure I.14 : Longueur d'une fraise. [5]

✓ **Données de l'outil : cas de fraisage**

Pour que l'outil soit reconnu par la machine, il faut définir les données suivantes :

- Type de l'outil
- Longueur en direction de X(L1)
- Longueur en direction de Z(L2)

Rayon de l'outil

✓ **Type de l'outil :**

En fraisage, deux types d'outil sont envisagés s'il s'agit d'un foret : c'est le type 10 si l'outil est une fraise axée suivant la direction de z ; c'est le type 20 et si elle est axée suivant X, Y.

En fonction du type de l'outil ; les longueurs de l'outil à mémoriser dans le registre « TOOL » sont illustrées sur la figure I.15. **(Fig I.15)**

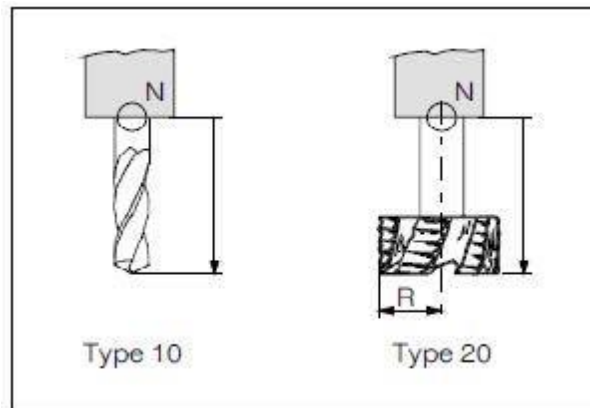


Figure I.15 : Type de l'outil (cas de fraisage). [5]

✓ **Données de l'outil : cas de tournage**

De même, pour que l'outil soit reconnu par la tour, il faut définir les données suivantes :

Type de l'outil

Longueur en direction de X(L1)

Longueur en direction de Z(L2)

La Rayon de la pointe de l'outil

✓ **Type de l'outil**

Pour définir le type de l'outil, nous regardons la fixation de l'outil sur le porte-outil. Les différents types sont donnés par la figure I.16. **(Fig I.16)**

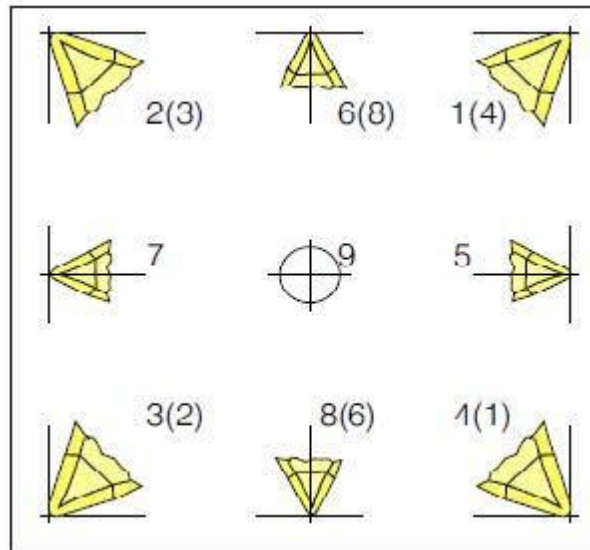


Figure I.16 : Position du bec des outils. [5]

✓ **Longueur de l'outil**

Les longueurs $L1$ et $L2$ sont les distances entre le point de référence du logement de l'outil «F» et la pointe de l'outil en direction de X et de Z respectivement. Dans le type 1a9, la saisie des données de longueur $L2$ en direction de Z comme c'est illustré sur la figure I.17. (**Fig I.17**)

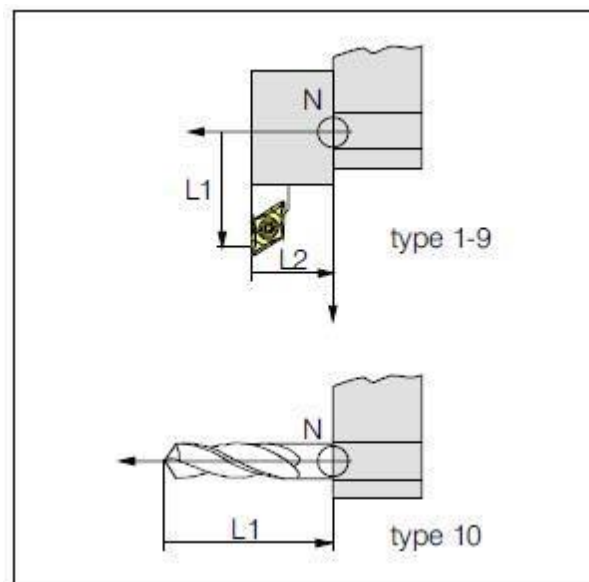


Figure I.17 : Longueur de l'outil. [5]

I.7.3 Décalage d'origine (G59)

C'est un décalage programmé qui peut être utilisé pour déterminer les origines de plusieurs parties de pièces (exemple : forme répétitives) ou de plusieurs pièces montées sur un même porte pièce. [6] Pour annuler la ou les décalages appliqués en G59, en programmation

absolue, il faut programmer G59 X0 Y0 Z0. En programmation relative, il est conseillé de repasser en programmation absolue G90 et de programmer G59 X0 Y0 Z0.

I.8 Programmation des MOCN

I.8.1 Organisation et structure d'un programme CNC

Un bloc est une suite de mots qui définit une seule instruction. Un bloc peut être un seul mot ou une combinaison de mots. Un bloc est terminé par un caractère de fin de bloc END OF BLOCK (EOB).

Dans la plupart des langages, l'EOB est visible, et dans certains langages, il n'est pas (NUM).

Quelques exemples de blocs sont fournis ci-dessous avec un EOB visible :

- ✓ M02 ; (bloc d'un mot)
- ✓ N11 M30 ; (bloc de deux mots)
- ✓ N12 T01 M06 ; (bloc de trois mots)
- ✓ N13 G80 G90 G18 ; (bloc de quatre mots)
- ✓ N14 G01 X5.0 Y107 F355 ; (bloc de cinq mots)

Chaque mot est composé d'une adresse et d'une valeur. Le code de l'adresse est le caractère de la seule lettre (A-Z) au début de chaque mot qui définit ce que l'ordinateur doit faire avec les données numériques qui suit.

Tableau I.3 : Exemple adresses de mot

Exemples adresses de mot
A axe rotatif X
B axe rotatif Y
C axe rotatif Z
X dimension du mouvement sur axe
Y dimension du mouvement sur axe
Z dimension du mouvement sur axe
D compensation outil
F Avance travail
G Fonction préparatoire (G-codes)
H Compensation outil
N Numéro de bloc
L répétition de cycle
R Rayon
S Vitesse de broche
T Numéro outil

Ces définitions d'adresses sont les plus couramment utilisées pour le fraisage et le tournage,

Ces définitions peuvent varier selon les langages. Les Mots composés d'adresses G, communément appelés fonctions G, sont les codes les plus utilisés en programmation G-Code (d'où son nom) (**Tableau I.3**). Beaucoup de ces G-codes sont normalisés à l'ensemble des langages et armoires CNC. Les plus connus sont les CODE G00 G01 G02 G03 Il s'agit de mouvements rapides et de travail, qui représentent les déplacements linéaires et circulaires. Les mots composés d'adresses M, communément appelé M-codes, sont des codes d'adresse machines qui n'impliquent pas de mouvement de l'outil.

Exemples :

- ♣ M03 mise en route la broche
- ♣ M07 Arrosage lubrification
- ♣ M06 changement de l'outil

I.8.2 Langage et programmation CNC

Le langage ISO

À l'origine, le langage de programmation était le G-code, développé par l'EIA au début des années 1960, et finalement normalisé par l'ISO en 1980 sous la référence RS274D/ (ISO 6983).

Le langage ISO est encore énormément répandu surtout sur les petites Machines commandes numériques. [8]

Le langage Fanuc

Le langage Fanuc prend pour base le langage ISO de 1980. Il ajoute des fonctions supplémentaires ce qui en fait un langage unique.

Les spécificités du langage sont:

- Parenthèses pour les commentaires
- Appel de sous programmes avec M98
- Points virgules en fin de blocs

Le langage NUM

Le langage NUM prend pour base le langage ISO. Il ajoute des fonctions supplémentaires ce qui en fait un langage unique.

Les spécificités du langage sont:

- Parenthèses pour les commentaires

- Appel de sous programmes avec G77

✚ Le langage SIEMENS

Le langage SIEMENS prend pour base le langage ISO. Il ajoute des fonctions supplémentaires ce qui en fait un langage unique. Les spécificités du langage sont:

- Points virgules pour les commentaires
- Des appels de cycles un peu compliqué.

✚ Le langage HEIDENHAIN

Le langage HEIDENHAIN est un langage inventé par les allemands pour animer principalement des robot CNC conversationnels.

✚ Le langage PROFORM

Le langage PROFORM a été inventé de toute pièces pour les robots charnière. Langage devenu totalement obsolète.

I.8.3 Décalage d'Origine réglable

Dans le programme CN, l'appel de l'une des instructions G54 à G57 déplace l'origine du système de coordonnées machine dans le système de coordonnées pièce. Dans le bloc CN qui suit et qui contient la programmation des déplacements, toutes les indications de position et, De ce fait, tous les déplacements d'outil se rapportent dorénavant à l'origine pièce définie précédemment. Grâce aux 4 décalages d'origine mis à disposition, il est possible, pour usiner par exemple des pièces identiques à des emplacements différents, de programmer et d'appeler simultanément 4 différentes prises de pièce dans le même programme (**Fig I.18**). [6]

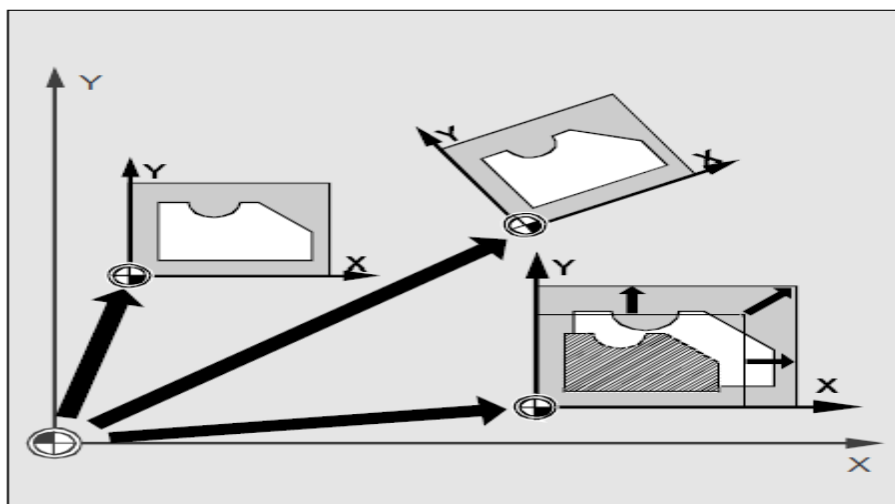


Figure I.18 : Décalage d'Origine réglable [6]

I.8.4 Décalage d'Origine Programmable TRANS / ATRANS

Les instructions permettent de programmer, pour tous les axes à interpolation et axes de Positionnement, des décalages d'origine dans le sens de l'axe indiqué. Vous pouvez ainsi travailler avec des origines variables, par exemple pour des opérations d'usinage qui se répètent en différentes positions de la pièce (**Fig I.19**). **TRANS** décalage d'origine en valeurs absolues, rapportés à l'origine pièce courante, origine pièce réglé avec G54 à G599. **ATRANS** comme TRANS, mais avec un décalage d'origine additif. [6]

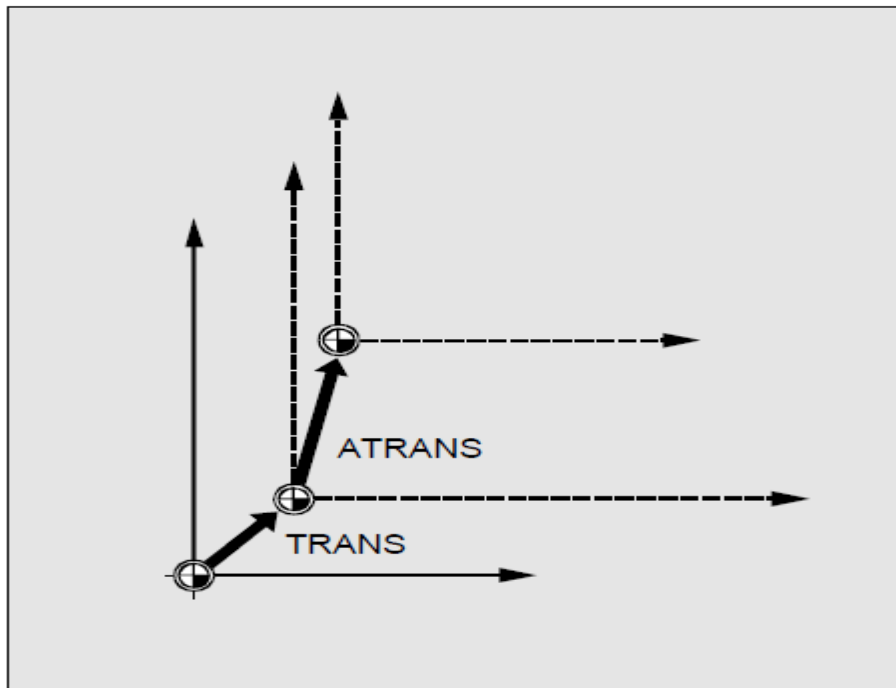


Figure I.19 : Décalage d'Origine Programmable TRANS / ATRANS [6]

I.8.5 Les principales fonctions

Le tableau I.4 montre les fonctions préparatoires G. [9]

Tableau I.4 : Les fonctions préparatoires, instructions G

COMMANDES	Description
G00	Avance rapide en coordonnées cartésiennes (la vitesse rapide est programmée par le constructeur).
G01	Interpolation linéaire en coordonnées cartésiennes.
G02	Interpolation circulaire dans le sens horaire en coordonnées cartésiennes.
G03	Interpolation circulaire dans le sens antihoraire (trigonométrique) en coordonnées cartésiennes.
G04	Temporisation.
G09	Arrêt précis.
G10	Avance rapide en coordonnées polaires.
G11	Interpolation linéaire en coordonnées polaires
G12	Interpolation circulaire dans le sens horaire en coordonnées polaires.
G13	Interpolation circulaire dans le sens antihoraire en coordonnées polaires.
G16	Sélection du plan.
G17	Sélection du plan.
G18	Sélection du plan.
G33	Filetage.
G40	Suppression
G 41	Compensation du rayon d'outil à gauche
G 42	Compensation du rayon d'outil à droite.
G 48	Retrait du contour de la manière dont s'est opérée l'approche.
G 50	Suppression modification de l'échelle.
G 51	Sélection modification de l'échelle.
G 53	Suppression du décalage d'origine.
G 54 à G 57	Décalage d'origine.
G 58 et G59	Décalage d'origines additives, programmables.
G 60	Mode d'arrêt précis.
G 62,63	Suppression du mode d'arrêt précis.
G71	Indication de cotes en millimètre.
G90	Programmation en cotes absolues.
G 91	Programmation en cotes relatives.
G 92	Limitation de vitesse pour G96.
G 94	Avance par minute, vitesse d'avance A en mm/mn.
G 95	Avance par tour (f) en mm/tr.
G 96	Vitesse de coupe (m/mn).
G 97	Vitesse constante

Le tableau I.5 montre les fonctions auxiliaires logiques M. [9]

Tableau I.5 : Les fonctions auxiliaires logiques M.

Commandes	Description
M00	Arrêt programmé
M01	Arrêt optionnel.
M02	Fin de programme pièce.
M03	Rotation de broche sens anti-trigonométrique.
M04	Rotation de broche sens trigonométrique.
M05*	Arrêt de broche.
M06	Changement d'outil.
M07	Arrosage n°2.
M08	Arrosage n° 1.
M09*	Arrêt des arrosages.
M10	Blocage d'axe.
M11	Déblocage d'axe.
M19	Indexation broche.
M40 à M45	6 gammes de broche.
M48*	Validation des potentiomètres de broche et d'avance.
M49	Inhibition des potentiomètres de broche et d'avance.
M64*	Commande broche 1.
M65	Commande broche 2.
M66*	Utilisation mesure broche 1.
M67	Utilisation mesure broche 2.
M998*	Mode MODIF, IMD et appel de sous-programme par l'automatenautorisé.
M999	Masquage par programmation des modes MODIF, IMD et appel de sous-programme par l'automate.

II- CHEMINEMENT DE L'INFORMATION CONCEPT- MOCN

Afin de permettre une compréhension des problèmes d'intégration des différents modules d'un système complet de fabrication, il convient tout d'abord de présenter le cheminement de l'information entre l'étape initiale du concept de la pièce à fabriquer jusqu'à l'obtention de la pièce physique. L'origine des problèmes d'intégration peut ainsi être mise à jour, ce qui permet d'adopter une meilleure approche à leur résolution.

Un système intégré de fabrication est un système qui comporte tous les modules nécessaires pour passer du concept d'une pièce à l'obtention de son modèle physique usiné. Ce système comporte différents modules logiciels et physiques, représentés par des rectangles à la figure 1, tel le module de conception assistée par ordinateur (CAO) ou encore la machine-outil à commande numérique (MOCN). Une flèche reliant deux modules à la figure 1 indique un transfert d'information entre les modules. Une flèche pleine indique la direction conventionnelle de transfert d'information alors qu'une flèche vide représente une direction possible que peut emprunter l'information selon les capacités des modules liés par les flèches.

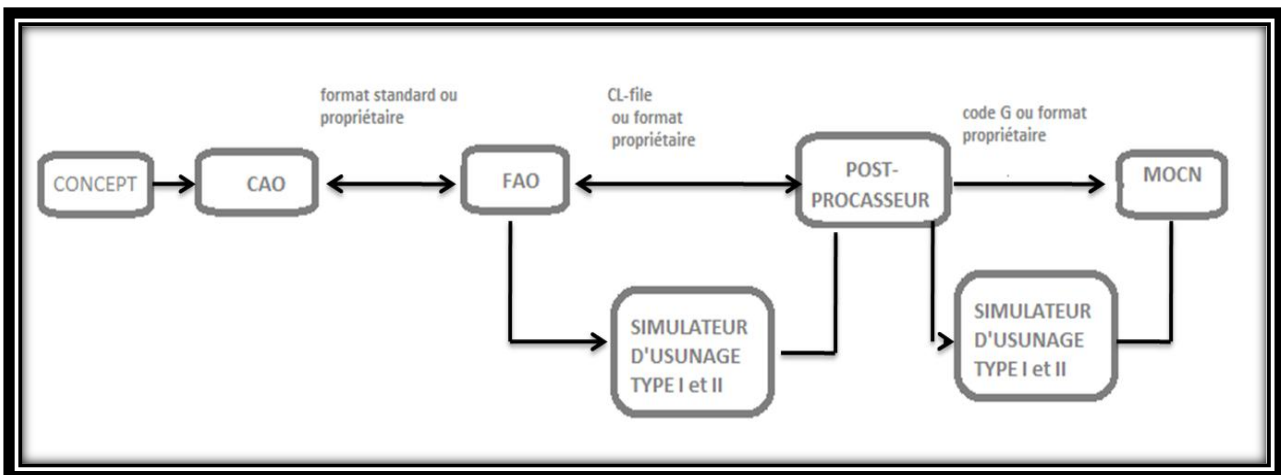


Figure 1 : Cheminement de l'information Concept- MOCN

On remarque qu'un format propriétaire ou standardisé est utilisé entre chacun des modules utilisés pour traiter l'information. Dans le cas du format propriétaire, une diminution de la flexibilité du système en entier en résulte.

L'information décrivant la pièce passe au format électronique dans le logiciel de CAO. pour ensuite être transférée au module de fabrication assistée par ordinateur (FAO). L'information qui en résulte est ensuite traduite par le module de post-processeur

afin de produire les données qui seront envoyées au contrôleur de la machine-outil à commande numérique (MOCN). Des étapes de simulation d'usinage peuvent être optionnellement incluses avant le module de post-processeur et avant le transfert à la MOCN. Les prochaines sections décrivent en détailles problèmes d'intégration présents lors du transfert d'information entre chacun des modules illustrés.

II.1 Cheminement Concept- CAO

Initialement, les caractéristiques de la pièce à fabriquer sont déterminées par le rôle de la pièce en service. On détermine alors des caractéristiques telles que le fini de surface, la précision requise, le matériel à utiliser, les traitements techniques et de surface à réaliser, pour ne citer que celles-ci. À partir de ces informations, la conception de la pièce à l'aide d'un logiciel de CAO peut être effectuée. C'est à cette étape que les informations qui définissent la pièce sont transformées sous une forme électronique qui sera sauvegardée dans un fichier, soit sous un format propriétaire au logiciel utilisé pour effectuer la conception ou encore sous un format neutre.

Il est toutefois courant que la compagnie qui effectue la conception de la pièce relègue la fabrication à un sous-traitant qui possède un logiciel CAO différent de celui qui a servi à réaliser la conception. Puisque les formats de fichiers des logiciels sont propriétaires et non divulgués, ceci force l'utilisation d'un format neutre et standardisé pour effectuer le transfert entre les deux plate-formes.

Toutefois, l'utilisation d'un format standardisé résulte inmanquablement en une perte d'information, et l'ampleur de cette perte d'information est dépendante à la fois des capacités du format neutre choisi et des différents types d'information qui peuvent être représentées par le logiciel CAO utilisé. La majorité des formats standardisés actuels ne permettent de représenter que la géométrie et la topologie de la pièce, mis à part le format STEP (Standard for the Exchange of Product model data) dont il sera question à la prochaine section.

Les logiciels de CAO modernes incluent beaucoup plus d'information dans le modèle de la pièce que la géométrie seule, et toute cette information supplémentaire ne peut transiger par le format neutre et sera donc éliminée. Parmi les informations qui ne sont pas transmises par le format neutre, notons les types suivants :

1. Caractéristiques de la pièce (features)
2. Paramétrisation
3. Propriétés physiques
4. Associativité géométrique

La majorité des logiciels de CAO permettent d'utiliser une approche basée sur les caractéristiques de la pièce afin de réaliser la modélisation "featurebased part modeling". Ainsi, le design de la pièce est effectué à l'aide des éléments qui représentent la pièce tels que des alésages, des poches, des chambrages ou des îlots. Une telle méthode est très efficace car elle est basée sur la fonctionnalité de la pièce et les caractéristiques à usiner qui sont incluses directement dans le modèle. Une telle représentation est beaucoup plus riche que celle ne contenant que la géométrie.

De plus, les principales plate-formes CAO permettent de décrire paramétriquement les relations entre les éléments géométriques de la pièce ou entre les caractéristiques. On peut ainsi définir mathématiquement les relations fonctionnelles de la pièce à même le modèle.

Ceci permet de pousser à un niveau plus élevé l'incorporation de la fonctionnalité de la pièce dans le modèle. Il est donc possible d'établir une relation qui assure, par exemple, que le diamètre d'un alésage est égal à 50% d'un second alésage appartenant à la pièce. Lors de la modélisation d'assemblages de pièces, des contraintes qui dictent la position relative des pièces sont incorporées dans le modèle. De plus, des relations paramétriques peuvent définir les liens entre les différentes pièces. À titre d'exemple, il est possible de spécifier que deux alésages situés sur deux pièces différentes de l'assemblage conserveront le même diamètre.

En addition au paramétrage, les modules de CAO permettent de définir les propriétés physiques et mécaniques de plusieurs matériaux utilisés dans les modèles, de sorte qu'il soit possible d'utiliser ces informations dans les modules d'analyse par éléments finis ou de génération de trajectoires d'outils (FAO). Dans ce dernier cas, ce type d'information pourrait être utilisé afin de suggérer des valeurs par défaut pour les vitesses de coupe et d'avance.

Tout ceci n'est possible que si ces informations peuvent être transmises entre les modules par le format de fichier choisi, ce qui n'est pas le cas avec l'utilisation des formats normalisés.

De plus, les formats normalisés utilisés par les logiciels de CAO ne permettent de représenter qu'un nombre fini de types différents d'entités géométriques. Habituellement, les logiciels comportent des type d'entités géométriques dont la représentation est propriétaire et ne possèdent pas de représentation équivalente dans un format neutre. Une conversion doit alors prendre place afin que les éléments géométriques touchés puissent être approximés à l'aide des types d'entités disponibles dans le format standardisé choisi pour le transfert d'information.

Citons comme exemple le format normalisé IGES (Initial Graphie Exchange Specification) qui est le plus largement utilisé et qui permet la représentation d'un maximum de 50 types différents d'entités géométriques [1]. Ainsi, un fichier converti en IGES et lu à nouveau dans le logiciel CAO l'ayant généré risque de présenter des différences au niveau de la représentation géométrique et jusqu'à un certain point des inexactitudes de topologies lorsque comparé au modèle original avant l'exportation.

Il est en effet courant qu'une réparation des surfaces représentant une pièce soit nécessaire suite à l'importation par l'entremise d'un format standardisé. Ces réparations visent principalement des surfaces initialement adjacentes et dont les arêtes initialement communes s'entrecroisent ou sont espacées.

Il va sans dire que la perte d'informations liée au transfert par l'entremise d'un format neutre résulte en une diminution de productivité et d'efficacité, tout en augmentant les risques d'erreurs et en diminuant la précision.

II.2 Cheminement CAO-FAO

Lorsque la conception CAO est achevée, il est nécessaire de générer à l'aide d'un logiciel FAO les trajectoires d'outils qui seront utilisées par la MOCN afin d'effectuer l'usinage requis pour obtenir le produit fini à partir du brut. n existe alors deux types de cheminements possibles, soit l'emploi d'un format natif commun au module de CAO et de FAO ou l'emploi d'un format neutre normalisé.

Dans le premier cas, le logiciel de FAO est intégré au logiciel de CAO utilisé pour la conception, ce qui permet de lire la géométrie à usiner directement à partir d'un fichier au format natif du logiciel utilisé pour la conception. Les caractéristiques à usiner sont ainsi transférées à partir du format natif du logiciel au module FAO, ce qui permet d'associer les opérations d'usinage aux caractéristiques de la pièce. Cette approche permet notamment l'obtention d'une associativité entre les trajets d'outils et la géométrie de la pièce, tout en conservant les qualités de paramétrisation et d'associativité géométrique.

Une modification des dimensions de la pièce ou de sa géométrie dans le module CAO sera immédiatement repercutée sur les trajectoires d'outils qui seront mises à jour afin de refléter les changements apportés à la pièce. Bien que ce type de cheminement de données entre les modules CAO et FAO ne présente pas de problèmes d'intégration, il peut dans certains cas comporter des désavantages. En effet, les efforts de développement d'une telle plate-forme CFAO sont distribués sur les différents modules du logiciel et ne sont pas uniquement axés sur l'évolution des stratégies d'usinage et la qualité des algorithmes de génération de trajectoires d'outils. Certains logiciels sont toutefois orientés purement vers la génération de trajectoires d'outils et offrent parfois des capacités supérieures en ce domaine que les plate-formes CFAO.

L'utilisation de logiciels axés sur la FAO représente le deuxième type de cheminement CAO-FAO possible. La géométrie à usiner sera donc créée dans un logiciel de CAO performant, puis elle sera transférée au logiciel de FAO par l'entremise d'un format neutre.

Toutefois, l'utilisation de ces formats standardisés de transfert fait en sorte que l'ensemble des caractéristiques composant la pièce est perdu. Seules des représentations de type fil de fer, surfacique ou solide sont transférées. De plus, cette représentation risque de nécessiter fréquemment des réparations de surfaces ou ne représente pas avec exactitude le modèle de départ.

De plus, cette approche n'est pas propice aux modifications de la géométrie de la pièce ou des dimensions. En effet, tout changement exige une modification du modèle dans le système de CAO qui doit de nouveau être transféré par un format neutre au logiciel FAO.

Puisqu'il n'existe pas d'associativité entre les surfaces et les trajectoires d'outils lors du transfert par format neutre, les trajectoires d'outils doivent être de nouveau définies en entier.

II.3 Cheminement FAO- Post-Processeur

C'est au niveau du transfert d'information entre les systèmes FAO et les Post-Processeurs qu'existent la majorité des problèmes d'intégration dans tout le système de fabrication intégré. Afin de comprendre la problématique de transfert d'information entre le système FAO et le Post-Processeur, il convient de présenter le langage APT (Automatically Programmed Tools). Ce langage a été créé au MIT en parallèle avec la première MOCN. Les entrées à ce logiciel sont la géométrie à usiner ainsi que les déplacements que l'outil doit effectuer en relation avec cette géométrie. La sortie, appelée CL-File (Cutter Location File), contient les coordonnées des points par lesquels devra passer l'outil de coupe afin d'usiner la géométrie définie et qui a servi d'entrée au programme [5]. Ce langage est standardisé par ANSI NCITS 37 : 1999 et est géré par le comité 17 [2] qui est toujours actif et se charge d'étendre les possibilités du langage afin de l'adapter aux nouvelles réalités technologiques du domaine de l'usinage. telles que les courbes NURBS (Non-uniform Rational B-Spline).

Les grandes plate-formes FAO et CFAO telles que Pro!ENGINEER et CATIA utilisent le CL-File comme format de représentation des trajectoires d'outils. Ce format contient les déplacements d'outils, traduit à l'aide d'un sous-ensemble de mots du langage APT, nécessaires à l'usinage de la pièce . Toutefois, le langage d'entrée utilisé par la majorité des MOCN est différent du CL-File. Il convient alors d'utiliser un post-processeur, qui va traiter à nouveau cette information et créer un fichier de données au format du langage d'entrée de la MOCN, soit le code G, que ISO 6983 tente de standardiser. Bien que le format des langages APT et CL-File soit normalisé, il existe des déviations à ces normes. Ainsi, il est possible que certaines commandes présentes dans le CL-File que produit un système de FAO ne soient pas utilisables par un post-processeur développé pour un autre logiciel de FAO qui utilise également le CL-File comme format de sortie.

De plus, seulement les grandes plate-formes FAO et CFAO utilisent nativement le CLFile comme format de sortie et utilisent le langage APT afin d'obtenir les trajectoires

d'outils. Les logiciels FAO de moins grande envergure utilisent un format intermédiaire propriétaire qui est converti par un post-processeur également propriétaire. Ce type de post-processeur utilise le format intermédiaire comme entrée afin de générer le code G requis par la machine. Ceci force l'utilisation du post-processeur livré avec le système de FAO afin de générer le code G nécessaire au contrôle de la MOCN. Généralement, ces post-processeurs intégrés sont beaucoup moins flexibles au niveau de leur capacité de personnalisation que ceux disponibles individuellement. Tel qu'illustré à la figure 1, il est possible d'effectuer une simulation d'usinage avant d'effectuer le transfert de l'information au module de post-processeur. Cette simulation permet de réaliser un usinage graphique virtuel de la pièce afin de s'assurer que les trajectoires d'outils ainsi que les paramètres associés soient définis de manière convenable dans le module de FAO.

Il existe principalement deux catégories principales de modules de simulation d'usinage. Le premier type effectue une simulation à l'aide de la géométrie de la pièce finie, de la pièce brute et de l'outil. Le second type, en plus de représenter les fonctionnalités du premier type, fournit également une représentation de la MOCN qui effectue l'usinage.

Ce second type est plus sécuritaire car il permet de détecter des erreurs d'usinage liées à la configuration et aux limites physiques de la MOCN. Lors du cheminement d'information entre le module de FAO et le module de post-processeur, l'information décrivant les trajectoires d'outils n'est disponible que sous forme de CL-File ou format propriétaire équivalent. C'est donc à partir de ces données que le module de simulation d'usinage pourra effectuer la représentation graphique des déplacements d'outils.

II.4 Cheminement Post-Processeur - MOCN

Le module de Post-processeur est utilisé avec chaque système FAO qui doit produire des données pour une MOCN. Le rôle principal du post-processeur consiste à traduire l'information contenue dans le CL-File en code G pour une machine spécifique. On parle ici de traducteur car le post-processeur ne distille pas l'information contenue

dans le CLFile mais se contente d'en transformer le format et la syntaxe de manière à produire de l'information directement utilisable par la MOCN.

Un post-processeur particulier est défini en fonction du format de code G que nécessite la MOCN pour laquelle il a été créé. Afin de pouvoir traiter correctement les données reçues sous forme de CL-File, l'information dont doit disposer le post-processeur inclut.

entre autres, le type de MOCN, la configuration de celle-ci ainsi que les particularités du contrôleur de cette machine. Le type de MOCN indique au post-processeur sur quelle machine sera effectué l'usinage, tel un tour, une fraiseuse ou encore une rectifieuse. C'est avec cette information qu'une interprétation correcte des informations contenues dans le CL-File est réalisée.

Le post-processeur doit également connaître la configuration de la machine. Ainsi, des informations concernant le nombre d'axes de déplacement ainsi que leur type, linéaire ou rotatif, doivent être fournies lors de la mise en place du module. Les limites de déplacement de chacun des axes doivent également être spécifiées, ainsi que l'orientation relative de chacun de ces derniers. Puisque chaque contrôleur possède des variantes au format de code G utilisé, l'interprétation des codes G par la MOCN doit être fournie afin que le post-processeur puisse générer les codes requis par la machine, et ce, au format requis. Puisque le post-processeur dépend des particularités de l'ensemble incluant la machine, sa configuration et son contrôleur, un post-processeur différent doit être créé pour chaque modèle et marque de MOCN. De plus, le format d'entrée de données aux contrôleurs de MOCN, soit le code G, dévie largement des standards définis par ISO, et ce, d'un fabricant de contrôleur à un autre et pour les différents modèles de contrôleur d'un même fabricant.

Ces écarts à la norme sont dus au marché hautement compétitif qui existe dans ce domaine. Lorsque des innovations technologiques permettent l'implantation de nouvelles fonctionnalités à même les contrôleurs de MOCN, elles sont incluses immédiatement et de nouveaux codes G sont créés en conséquence. Cet ajout est effectué sans attendre la normalisation de ces fonctionnalités, ce qui exigerait trop de temps. Ceci permet aux fabricants de conserver un avantage technologique sur leurs concurrents. Puisque chaque compagnie qui produit des contrôleurs agit de la sorte, il existe

présentement une multitude de fonctionnalités présentes dans les contrôleurs de MOCN et régies de manière propriétaire.

Ces différents formats de code G nécessitent la création d'un post-processeur par modèle de MOCN que possède une compagnie, ce qui entraîne des coûts élevés de mise en place, sans compter l'entretien des post-processeurs requis lorsque la plate-forme FAO est mise à jour ou modifiée. De plus, il existe des formats propriétaires autres que le code G utilisé afin de contrôler les MOCN. Par exemple, les contrôleurs Heidenhain acceptent à la fois le code G, de manière limitée, et un langage conversationnel propriétaire à Heidenhain [3].

Il en est de même avec les MOCN Mazak dont le contrôleur peut interpréter, outre le code G, le langage conversationnel propriétaire Mazak, le Mazatrol [4].

Dans le cas où un module de simulation d'usinage est utilisé entre le module de postprocesseur et celui de la MOCN, il doit utiliser l'information sous forme de code G afin d'illustrer les déplacements d'outils et accessoirement ceux de la MOCN en entier. n'importe de préciser que l'exécution du post-traitement ne requiert pas d'entrée de données de la part de l'utilisateur. On peut concevoir qu'il serait possible d'intégrer le postprocesseur au contrôleur de la MOCN et d'envoyer directement les données à celle-ci sous forme de CL-File, afin d'éliminer entièrement l'étape de post-traitement.

II. 5 Module de simulation d'usinage

Il existe principalement deux types de simulateur d'usinage. Le premier type (Type 1) simule uniquement l'outil et la pièce. Il permet d'évaluer qualitativement l'efficacité en usinage des trajectoires générées en permettant une observation graphique des déplacements d'outils dans le brut et en présentant l'état de la pièce suite à chaque séquence d'usinage. Il permet également de détecter les collisions en avance rapide entre l'outil et le brut ou encore les brides. Un tel simulateur est souvent intégré aux logiciels de FAO afin de vérifier les trajectoires d'outils pendant leur création.

Les simulateurs du second type (Type II) offrent l'ensemble des possibilités des simulateurs de Type 1 en plus d'offrir une simulation complète des déplacements des composantes de la MOCN, ce qui permet de détecter les erreurs de dépassement de course des axes ou les collisions machine lors de l'usinage 5 axes ou autres.

Ces deux types de simulateur peuvent réaliser l'usinage virtuel à partir de deux types de données, soit le CL-File (ou son équivalent) et le code G. Il va sans dire que la simulation basée sur le CL-File ne permet pas une vérification complète car la MOCN utilise l'information sous forme de code G pour effectuer l'usinage, et non le CL-File. Il est ainsi possible que la simulation basée sur le CL-File ne présente pas d'erreurs alors que celle basée sur le code G en contienne. Il convient donc, lorsque possible, d'effectuer la simulation à l'aide des données qui seront directement envoyées à la MOCN.

La création du simulateur d'usinage pour chaque MOCN exige de définir une réplique virtuelle de la géométrie de la machine, de ses axes de déplacement et de leurs limites ainsi que du contrôleur de la MOCN. Toute cette information a toutefois déjà été utilisée afin de construire le post-processeur. Il s'agit donc d'une double entrée de données qui est inutile et peut mener à des erreurs résultant de la mise à jour des informations de la machine d'une manière différente dans le post et dans le simulateur. [7].

II.6 Module de la MOCN

La vaste majorité des contrôleurs de MOCN en utilisation dans les industries et offerts par les fabricants sont basés sur un contrôleur à logique programmable (PLC). Ces contrôleurs sont caractérisés par une architecture relativement fermée, ce qui devient problématique lorsqu'ils doivent être intégrés à un système de fabrication complet. Habituellement, la principale information qui peut être transférée à ces contrôleurs est le programme d'usinage en format code G ou conversationnel. Afin d'offrir une meilleure vue d'ensemble des problèmes liés au transfert de données vers les contrôleurs de MOCN, nous présenterons en détail dans les sections qui suivent les fonctionnalités des MOCN ainsi que les interfaces permettant le transfert des programmes d'usinage au contrôleur.

II.7 Module FAO

Puisque les implantations de STEP présentement disponibles à même les modules CAO et FAO sont limitées au transfert d'information géométrique et topologique, tout modèle transféré au module FAO par un format autre que le format natif du logiciel n'est qu'un ensemble de fils de fer, de surfaces et de solides, selon le format neutre utilisé pour le transfert. Ceci pose un sérieux inconvénient au niveau FAO car toutes les caractéristiques (features) composant la pièce sont perdues. Il devient alors impossible, par exemple, de spécifier l'élément géométrique à usiner par un cycle de poche en choisissant celle-ci, il faut alors sélectionner l'ensemble des surfaces constituant

la poche. n en résulte une perte de performance et de flexibilité d'utilisation du modèle associé.

II.8 Transfert d'informations FAO- Post-processeur

L'information transmise au Post-Processeur par le système FAO, sous format propriétaire ou de CL-File, est constitué d'une information de piètre qualité. En effet, le système FAO qui possède l'ensemble de l'information géométrique et topologique de la pièce à usiner ainsi que les paramètres associés à l'usinage ne transmet au module de post-processeur que la position des points par lesquels devra passer la pointe de l'outil pendant l'usinage.

II.8-1 Module Post-processeur

L'existence du module de post-processeur n'est due qu'aux différences entre les formats utilisés à la sortie des systèmes FAO et à l'entrée des contrôleurs de MOCN. Le BCL en implantation native et le format STEP-NC permettent d'effectuer l'usinage sans avoir recours au module de post-processeur qui est voué à disparaître dans quelques décennies. Ceci explique que peu de recherche est effectuée afin d'améliorer l'intégration du post-processeur au reste du système intégré de fabrication.

En effet, depuis quelques années. Des tours à commande numérique comportant une broche de fraisage montée au porte-outil et capable de réaliser des déplacements selon trois axes cartésiens et un axe rotatif sont maintenant disponibles. Des tours comportant deux tourelles portes-outils et deux broches opposées qui permettent de l'usinage en simultané et le transfert de la pièce d'une broche à l'autre ont également fait leur apparition. [10]

La majorité des modules FAO ne peuvent générer en une seule opération les CL-Files nécessaires à l'utilisation optimale de l'ensemble des nouvelles stratégies d'usinage offertes par de telles machines. De plus, lorsque les CL-Files sont obtenus, la majorité des post-processeurs intégrés aux modules FAO ne peuvent prendre en charge de telles configurations de MOCN. Les solutions de post-traitement qui supportent de telles configurations de machine le font d'une manière souvent sommaire et peuvent présenter certains problèmes lorsque la vitesse d'avance de l'outil dans la pièce doit être conservée constante pendant un déplacement utilisant un ou plusieurs axes rotatifs en simultané [2]

De plus, les post-processeurs n'offrent aucun support natif permettant d'utiliser les fonctionnalités avancées disponibles sur plusieurs MOCN, tels des cycles d'usinage de géométrie prismatique et des cycles de mesure de pièce. Certains environnements de développement de post-processeurs offrent un langage de programmation de base permettant de mettre en place un support pour ces routines avancées.

II-8-2 Module MOCN

Mis à part le module de CAO, tous les modules d'un système intégré de fabrication n'ont pour but principal que de produire l'information que nécessite la MOCN afin de produire la pièce. Il existe un volume important de recherches portant sur l'intégration des MOCN aux autres modules du système de fabrication. Le BCL présenté à la section 1.2.3.2 et le STEP-NC introduit à la section 1.2.1.1 proposent d'augmenter le niveau d'intégration avec la MOCN en éliminant le post-processeur et en transférant l'information d'usinage directement au contrôleur de la MOCN. Toutefois, ces solutions exigent une architecture de contrôleur complètement différente à celle implantée présentement, ce qui diminue fortement les avantages de telles solutions. [10]

En effet, l'investissement initial nécessaire à l'acquisition d'une MOCN est élevé et inclut, mis à part le coût de la MOCN elle-même, celui de l'infrastructure, de la formation des employés, de l'intégration au système de fabrication de l'usine et de l'outillage. Puisque la durée de vie moyenne d'une MOCN varie de 5 à 30 ans selon l'application de la machine, un remplacement complet du parc de machine d'une compagnie n'est pas envisageable. Les solutions radicales du type BCL et STEP-NC ne seront donc implantées qu'à très long terme.

Les limitations à l'augmentation du niveau d'intégration entre le module MOCN et les autres modules d'un système de fabrication sont principalement dues à trois facteurs. Tout d'abord, les contrôleurs de MOCN ne peuvent traiter que des interpolations linéaires et circulaires. Récemment, des interpolateurs NURBS qui utilisent des codes G propriétaires ont été introduits. La seconde limitation concerne le langage utilisé pour transférer l'information à la MOCN, soit le code G, qui est maintenant désuet et ne permet qu'une représentation ponctuelle des points par lesquels l'outil devra se déplacer afin de réaliser l'usinage. Cette information appauvrie ne permet pas d'exploiter l'ensemble des capacités qui seraient disponibles à même les contrôleurs de MOCN. [10]

Finally, the method from which the machining program is obtained constitutes in itself a significant limitation. During the generation of tool trajectories in the CAM module, a precise parametric representation of the surfaces to be machined is available through the CAD model. Since the CL-File to be obtained, which must eventually be transformed into G-code, cannot contain only linear or circular displacements, an approximation of the tool trajectories is obtained using this type of elements. A significant degradation of the information quality is thus created. Moreover, a compromise must be made between the precision of the tool trajectories and the size of the resulting machining program. [7].

I-INTRODUCTION

La programmation paramétrique est une programmation normale où on fait intervenir des variables qui ont la possibilité de changer en cours de programme. En fait, c'est un simple tableau à une entrée qui porte un numéro en colonne commençant par #, et la valeur attribuée à cette variable. Elle constitue une innovation technique de programmation G / M-code de la langue de programmation. Cette innovation est mal connue, ce qui est inhabituel en raison du fait qu'il simplifie les procédures de la commande numérique de façon spectaculaire, diminue la dépendance à la CAO / FAO, et minimise le cycle du temps.

La programmation paramétrique peut être comparée à n'importe quel langage de programmation informatique tel que BASIC, langage C et PASCAL. Cependant, ce langage de programmation réside directement dans le contrôle CNC, il est accessible au niveau du code G, ce qui signifie qu'on peut combiner des techniques de programmation manuelle avec des techniques de programmation paramétriques. Des fonctionnalités informatiques telles que les variables, l'arithmétique, les instructions logiques et les boucles sont disponibles.

La programmation paramétrique existe en plusieurs versions. La plus populaire est Custom Macro B (utilisée par Fanuc et les contrôles compatibles avec Fanuc). D'autres incluent User Task (de Okuma), Q Routine (de Sodick) et langage de programmation avancé (Advanced Programming Language [APL] (de G & L)). Outre de nombreuses fonctionnalités informatiques, la plupart des versions de programmation paramétrique disposent de fonctionnalités étendues liées à la CNC. La macro personnalisée, par exemple, permet à l'utilisateur de la CNC d'accéder à de nombreuses informations sur la commande CNC (corrections d'outil, position des axes, alarmes, génération de codes G et protection de programme) directement depuis un programme CNC. Ces choses sont impossibles avec des techniques de programmation de code G normales uniquement.

II-OPERATIONS ARITHMETIQUES

Les opérations figurant dans le tableau ci-dessous peuvent porter sur des variables. L'expression à droite du signe '=' peut contenir des constantes et/ou des variables combinées par une fonction ou un opérateur. Les parenthèses, chez FANUC, sont réservées pour les commentaires. Lors de l'utilisation de fonctions complexes dans l'expression de droite, il conviendra d'utiliser les crochets ouvert/ fermé '[' ']' à la place.

Exemple 1 : pour écrire que le contenu de la variable numéro 100 est égale au contenu de la variable numéro 500 divisée par le contenu de la variable numéro 502 à laquelle il faut rajouter 1, on écrira : $\#100 = \#500 / [\#502 + 1]$ [12]

Les variables #i, #j et #k dans une expression sont à remplacer par un numéro de variable valide. Les variables #j et #k peuvent être remplacées par une constante.

Fonction	Format	Remarques
Définition	$\#i = \#j$	
Addition	$\#i = \#j + \#k$	Pensez à la priorité des opérateurs : multiplication, division, addition et soustraction
Soustraction	$\#i = \#j - \#k$	
Multiplication	$\#i = \#j * \#k$	
Division	$\#i = \#j / \#k$	
Sinus	$\#i = \text{SIN}[\#j]$	Un angle est spécifié en degré décimal. Un angle de 90°30'15" est représenté comme 90.504166667°
Arc sinus	$\#i = \text{ASIN}[\#j / \#k]$	
Cosinus	$\#i = \text{COS}[\#j]$	
Arc cosinus	$\#i = \text{ACOS}[\#j / \#k]$	
Tangente	$\#i = \text{TAN}[\#j]$	
Arc tangente	$\#i = \text{ATAN}[\#j / \#k]$	
Racine carré	$\#i = \text{SQRT}[\#j]$	
Valeur absolue	$\#i = \text{ABSOLU}[\#j]$	
Arrondi au plus près	$\#i = \text{ROUND}[\#j]$	
Arrondi inférieur	$\#i = \text{FIX}[\#j]$	
Arrondi supérieur	$\#i = \text{FUP}[\#j]$	
Logarithme naturel	$\#i = \text{LN}[\#j]$	
Fonction exponentielle	$\#i = \text{EXP}[\#j]$	

N° de variable	Type de variable	Fonction
#0	Variable locale toujours à 0	Cette variable est toujours zéro. Aucune valeur ne peut être attribuée à cette variable.
#1 à #33	Variables locales	Les variables locales peuvent uniquement être utilisés dans une macro pour conserver des données comme les résultats d'opérations. Lors de la mise hors tension, les variables locales sont réinitialisées. Lorsqu'une macro est appelée, des lettres sont attribuées aux variables locales.
#100 – #149 (#199) #500 – #531 (#999)	Variables communes	Les variables communes peuvent être réparties entre différents programmes macro. Lors de la mise hors tension, les variables #100 à #149 sont réinitialisées. Les variables #500 à #531 conservent les données même lors de la mise hors tension. En option, les variables #150 à #199 et #532 à #999 sont également disponibles.
#1000_	Variables système	Les variables du système sont utilisées pour lire et écrire une variété d'articles de données CN comme la position actuelle et les valeurs de compensation d'outil.

III-OPERATIONS LOGIQUES

Les opérations logiques servent surtout à faire des tests pour des branchements conditionnels, pour l'affectation d'une valeur à une variable ou pour des répétitions.

Les différentes opérations logiques : **[12]**

Opérateur	Signification
EQ	Egal à
NE	Différent de
GT	Supérieur à
GE	Supérieur ou égal à
LT	Inférieur à
LE	Inférieur ou égal à

Affectation :

```
IF[#1GT12]THEN#1=12
```

Si la valeur de la variable #1 est plus grande que 12, alors la valeur de la variable #1 est égale à 12; sinon, la variable garde sa valeur.

Branchement conditionnel :

```
IF [#2EQ5] GOTO20
```

Si la valeur de la variable #2 est égale à 5, alors je vais à la ligne N20; sinon je continue à la ligne en dessous.

Répétition : instruction WHILE

Lorsque l'on veut répéter une séquence d'usinage plusieurs fois, on a notre disposition plusieurs méthodes. La première est celle citée plus haut (le branchement conditionnel).

La deuxième est l'instruction WHILE.

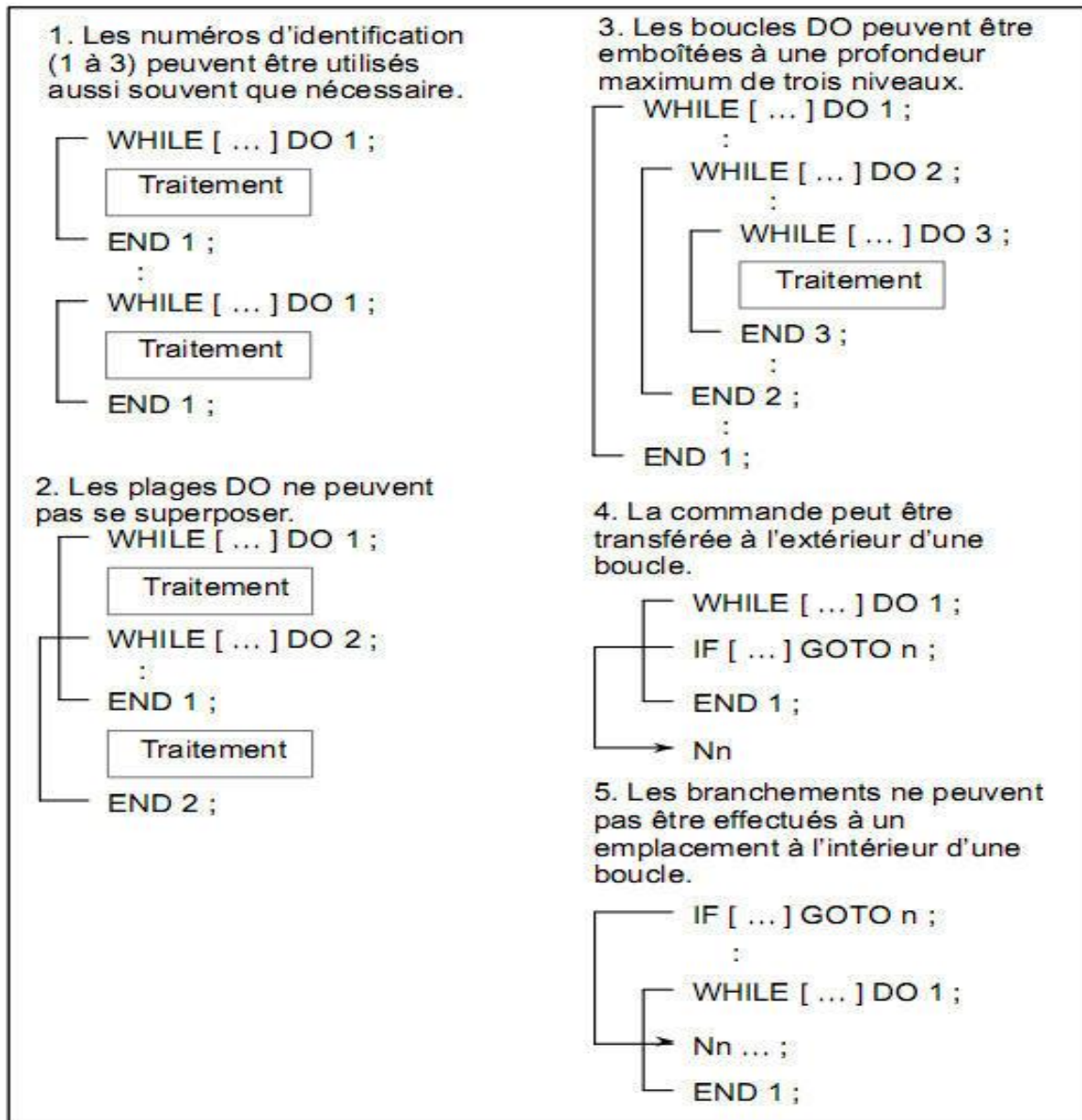
```
WHILE [#1LT10]DO1  
...  
...  
#1=#1+1  
END1
```

Une chose importante dans ce type de boucle, c'est qu'il faut toujours faire évoluer une variable utilisée dans le test : ici #1

Cette boucle signifie : Tant que la valeur de la variable #1 est plus petite que 10 alors on fait jusqu'à END1.

Lorsque le test devient faux, on saute après le END1 pour continuer le programme.

Il est possible d'imbriquer jusqu'à trois boucles WHILE.



Par exemple, si on a l'intention d'utiliser le même sous programme pour une ébauche et une finition et que l'on doit mettre le correcteur de rayon avec la prise de correction (c'est le cas pour les FANUC 0M) on pourrait très bien paramétrer le correcteur !

Programme principal	
<pre> T1 M6 (OUTIL NUMERO 1) #100=1 (DEFINITION DE LA VARIABLE AVEC VALEUR = 1) M98 P1234 F300 (APPEL DU SOUS PROGRAMME) T2 M6 (OUTIL NUMERO 2) #100=2 (DEFINITION DE LA VARIABLE AVEC VALEUR = 2) M98 P1234 F450 (APPEL DU SOUS PROGRAMME) </pre>	
Sous programme	
<pre> O1234 (SOUS PROGRAMME DE CONTOURNAGE) G1 Z-5. G41 X-2. Y0 D#100 (UTILISATION DE VARIABLE POUR LE CORRECTEUR DE RAYON) X50. (La première fois, appel du correcteur D1) Y-50. (La deuxième fois, appel du correcteur D2) X0 Y2. G40 X-10. G0 Z10. M99 </pre>	

Prenons un exemple et faisons les programmes :

<i>Travail à effectuer :</i>	à Surfaçage d'une plaque AU4G de 300mm de long par 250mm de large.
<i>Matériel à disposition :</i>	à <ul style="list-style-type: none"> • Un CU avec armoire FANUC • Le matériel en adéquation pour le bridage (mors crampons) • Fraise à surfacer Ø63 avec 5 plaquettes

Réalisation d'un programme standard : (on considère l'origine en haut à gauche)

```
O0001(SURFACAGE)
N10 M6 T1(APPEL FRAISE SURFACER)
N20 G0 G90 G54 X-40 Y-20 M3 S2500
N30 G43 H1 Z5 M8
N40 G1 Z0 F1875(0.15/DENT)
N50 X340
N60 Y-70(RECOUVREMENT 50MM)
N70 X-40
N80 Y-120
N90 X340
N100 Y-170
N110 X-40
N120 Y-220
N130 X340
N140 G0 Z5 M9
N150 G53 Y0 Z0
N160 M30
```

Si nous devons faire plusieurs passes sur Z (par exemple une ébauche et une finition), soit nous recopions les lignes N40 à N130 après la ligne N140 (ceci, sans oublier de repositionner la fraise à X-40 Y-20) soit nous mettons les lignes N50 à N140 dans un sous programme que nous appellerons après la mise en position et la descente.

Recopie de bloc	Appel de sous programme
O0001(SURFACAGE)	
N10 M6 T1(APPEL FRAISE SURFACER)	
N20 G0 G90 G54 X-40 Y-20 M3 S2500	
N30 G43 H1 Z5 M8	O0001(SURFACAGE)
N40 G1 Z0.5 F1875(0.15/DENT)	N10 M6 T1(APPEL FRAISE SURFACER)
N50 X340	N20 G0 G90 G54 X-40 Y-20 M3 S2500
N60 Y-70(RECOUVREMENT 50MM)	N30 G43 H1 Z5 M8
N70 X-40	N40 G1 Z0.5 F1875(0.15/DENT)
N80 Y-120	N50 M98 P2
N90 X340	N60 X-40 Y-20
N100 Y-170	N70 G1 Z0 F1875
N110 X-40	N80 M98 P2
N120 Y-220	N90 G53 Y0 Z0 M9
N130 X340	N100 M30
N140 G0 Z10	
N150 X-40 Y-20	O0002(SOUS PROGRAMME)
N160 G1 Z0 F1875	N10 X340
N170 X340	N20 Y-70(RECOUVREMENT 50MM)
N180 Y-70(RECOUVREMENT 50MM)	N30 X-40
N190 X-40	N40 Y-120
N200 Y-120	N50 X340
N210 X340	N60 Y-170
N220 Y-170	N70 X-40
N230 X-40	N80 Y-220
N240 Y-220	N90 X340
N250 X340	N100 G0 Z10
N260 G0 Z10 M9	N110 M99

Maintenant, plusieurs petits problèmes se posent à nous :

Si il faut faire plus que 2 passes en Z ? Réponse : soit recopie des blocs, soit encore des appels de sous programme. Si la pièce change de dimension ? Réponse : retoucher les blocs en conséquence (plus facile avec le sous programme).

Maintenant, entrons dans le vif du sujet : écrire un programme paramétré !

De quoi avons nous besoin ?

- de la longueur de pièce (qui peut changée)
- de la largeur (qui peut changée aussi)
- de l'épaisseur de matière à enlever (en partant du principe que le zéro pièce est toujours sur le dessus)

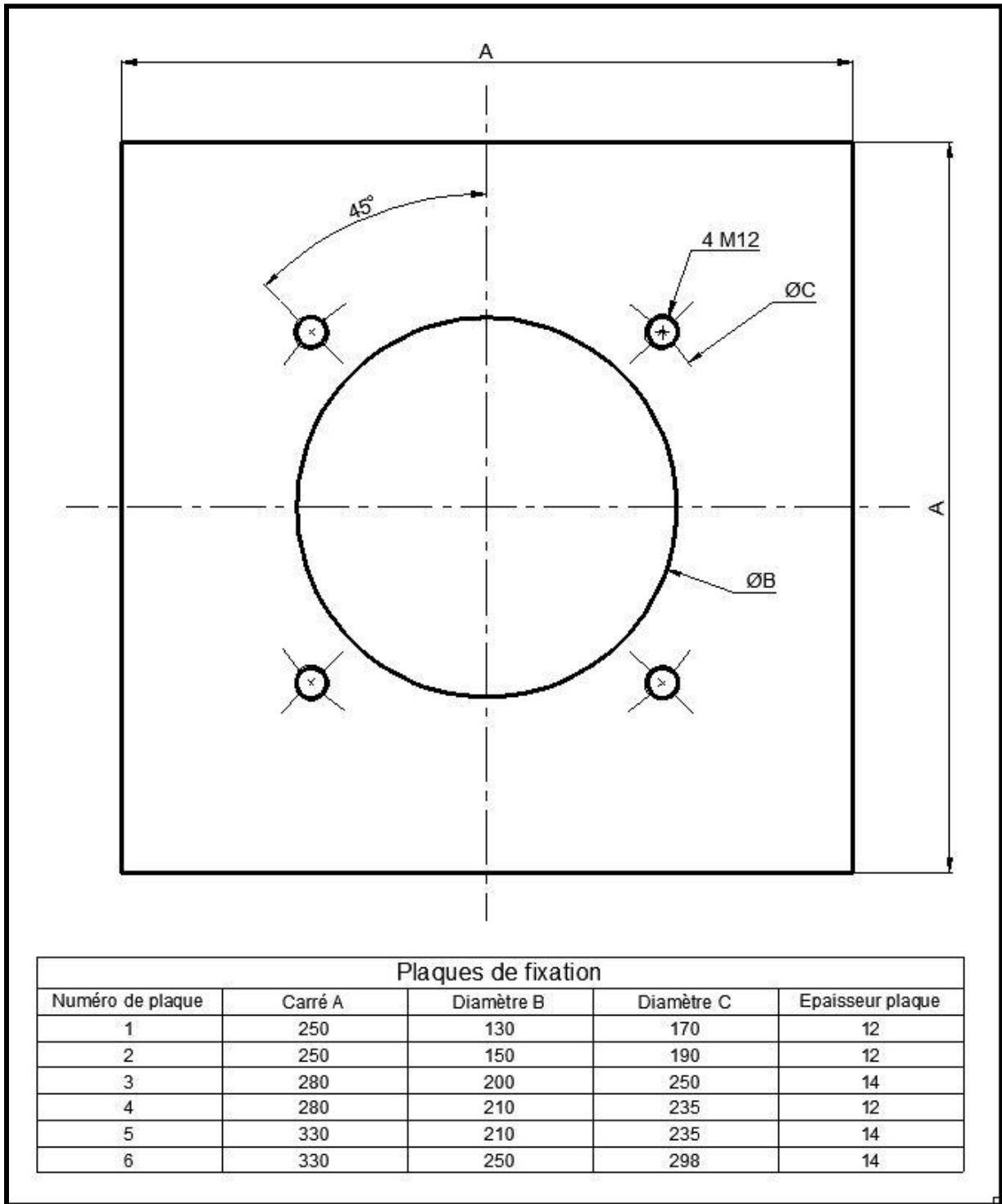
Pour enlever toute la matière, nous aurons donc besoin de deux boucles :

- une boucle pour surfacer toute la pièce.
- une boucle pour les passes en profondeur.

On écrit le code de cette façon :

En français	En langage FANUC
Numero de programme	O0001(SURFACAGE PARAMETRE)
Appel de l'outil	N10 T1 M6(FRAISE A SURFACER D63)
Positionnement	N20 G0 G90 G54 X-40 Y-20 M3 S2500
Longueur = 300	N30 #500=300
Largeur = 250	N40 #501=250
Epaisseur = 3	N50 #502=3
Appel correcteur de longueur et positionnement Z	N60 G43 H1 Z[#502+2] M8
TANT QUE Epaisseur plus grand que 0 faire	N70 WHILE[#502 GT 0]D01
Epaisseur = Epaisseur - 2.5	N80 #502=#502-2.5
Si Epaisseur plus petit que 0 alors Epaisseur = 0	N90 IF[#502 LT 0] THEN #502=0
Position_Y = 20	N100 #100=20
Déplacement_X = Longueur + 40 + 40	N110 #101=#500+40+40
Aller à X-40 Y-20 en rapide	N120 G0 G90 X-40 Y-20
Descendre à Z Epaisseur	N130 G1 Z#502 F1875
TANT QUE Position_Y est plus petit que (Largeur - 30)	N140 WHILE[#100 LT [#501-30]]D02
Aller à Y Position_Y en absolu	N150 G90 Y-#100
Position_Y = Position_Y + 50	N160 #100=#100+50
Faire un déplacement relatif en X de Déplacement_X	N170 G91 X#101
Inverser le sens de Déplacement_X	N180 #101=-#101
FIN TANT QUE Position_Y	N190 END2
Remonter en Z à Epaisseur + 2	N200 G0 G90 Z[#502+2]
FIN TANT QUE Epaisseur	N210 END1
Retour origine Y Z	N220 G0 G53 Y0 Z0 M9
Fin programme	N230 M30

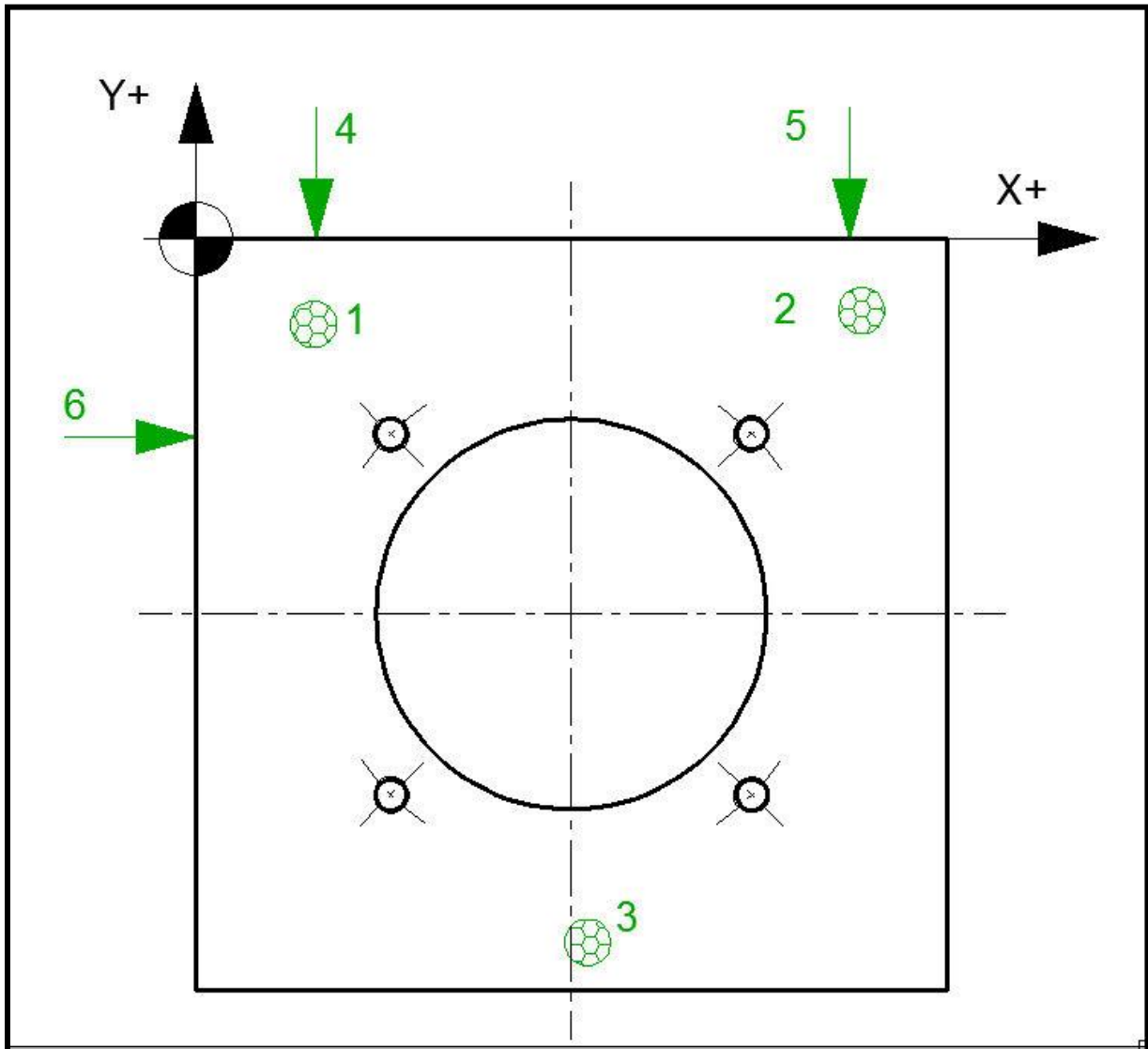
Exemple n=02



Pour la réalisation de cette pièce, nous devons faire la gamme d'usinage. Celle-ci est très simple :

P	SP	Op	Désignation	Outil/Machine	CTRL	Vc
100	110	111	Control du débit fourni par le client		Pied à coulisse	
200			Fraisage CN			
	210		Usinage complet	ROSILIO C450II		
		211	Perçage D20	Foret D20	Pied à coulisse	35 m/mn
		212	Usinage diamètre B	Fraise à plaquette D20	Pied à coulisse	180 m/mn
		213	Pointage des trous M12	Centreur 14/90°	Réglet	35 m/mn
		214	Perçage D10.25	Foret D10.3	Pied à coulisse	35 m/mn
		215	Taroudage M12	Taraud M12	Tampon fileté double M12	8 m/mn
300	310	311	Contrôle			
400	410	411	Stockage/livraison			

Une fois la gamme terminée, nous allons nous pencher sur le contrat de phase de la sous-phase 210. Celui ci fera apparaître la pièce en l'état, l'isostatisme, le travail à faire et l'origine du programme.



A partir de maintenant, il ne reste plus qu'à passer à l'écriture du programme proprement dit en respectant la structure expliquée dans la partie "Introduction". N'oubliez pas de toujours bien commenter vos programmes. Quand on les reprend plus tard, ça évite de se demander à quoi ils servent.

%

:1234 (**PLAQUES PARAMETREES**)

#500 = 1 (NUMERO PLAQUE A FAIRE)

GOTO#500 (ICI ON VA A LA PLAQUE EN REFERENCE)

N1(PLAQUE NUM 1)

#501 = 250(TAILLE PLAQUE A)

#502 = 130(ALESAGE B)

#503 = 170(DIAMETRE PERCAGE C)

#510 = 12(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N2(PLAQUE NUM 2)

#501 = 250(TAILLE PLAQUE A)

#502 = 150(ALESAGE B)

#503 = 190(DIAMETRE PERCAGE C)

#510 = 12(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N3(PLAQUE NUM 3)

#501 = 280(TAILLE PLAQUE A)

#502 = 200(ALESAGE B)

#503 = 250(DIAMETRE PERCAGE C)

#510 = 14(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N4(PLAQUE NUM 4)

#501 = 280(TAILLE PLAQUE A)

#502 = 210(ALESAGE B)

#503 = 235(DIAMETRE PERCAGE C)

#510 = 12(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N5(PLAQUE NUM 5)

#501 = 330(TAILLE PLAQUE A)

#502 = 210(ALESAGE B)

#503 = 235(DIAMETRE PERCAGE C)

#510 = 14(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N6 (PLAQUE NUM 6)

#501 = 330(TAILLE PLAQUE A)

#502 = 250(ALESAGE B)

#503 = 298(DIAMETRE PERCAGE C)

#510 = 14(EPAISSEUR PLAQUE)

GOTO100 (APRES AVOIR CHARGER LES VALEURS ON VA A L'USINAGE)

N100

G91 G28 Z0

M6 T1 (FORET DIA 20)

(PERCAGE POUR PASSAGE FRAISE A PLAQUETTE)

G52 X[#501/2] Y-[#501/2] Z#510 (DECALAGE ORIGINE MILIEU PIECE DESSUS)

G0 G90 G54 X[[#502/2]-12] Y0 M3 S560

G43 H1 Z2. M8

G73 Z-20. Q2 F50

G80 Z50. M9

N200 M6 T2 (FRAISE A PLAQUETTE D20)

(CONTOURNAGE DU DIAMETRE B)

G52 X[#501/2] Y-[#501/2] Z#510 (DECALAGE ORIGINE MILIEU PIECE DESSUS)

G0 G90 G54 X[[#502/2]-12] Y0 M3 S2870

G43 H2 Z2. M8

G1 Z-2. F250 (PROFONDEUR DE PASSE)

G41 X[#502/2] D22 F525

G3 I-[#502/2]

G1 G40 X[[#502/2]-12.]

Z-4. (PROFONDEUR DE PASSE)

G41 X[#502/2] F525

G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
Z-6. (PROFONDEUR DE PASSE)
G41 X[#502/2] F525
G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
Z-8. (PROFONDEUR DE PASSE)
G41 X[#502/2] F525
G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
Z-10. (PROFONDEUR DE PASSE)
G41 X[#502/2] F525
G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
Z-12. (PROFONDEUR DE PASSE)
G41 X[#502/2] F525
G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
Z-15. (PROFONDEUR DE PASSE)
G41 X[#502/2] F525
G3 I-[#502/2]
G1 G40 X[[#502/2]-12.]
G0 Z50 M9
G91 G28 Z0
N300 M6 T3 (**CENTREUR 14/90**)
(POINTAGE DES TROUS M12)
G52 X[#501/2] Y-[#501/2] Z#510 (DECALAGE ORIGINE MILIEU PIECE DESSUS)
G0 G90 G54 X0 Y0 M3 S1100
G43 H3 Z50M8
G98 G82 Z-5 P50 F120 R2 K0 (K0 -> LE PERCAGE ATTENDS UNE COORDONNEE)
G65 P4321 X0 Y0 D#503 K4 A45 (APPEL DE MACRO CLIENT DE CALCUL DES POINTS)

(MACRO CLIENT = O4321)
(DIAMETRE DE PERCAGE = #503)
(NOMBRE DE TROUS = 4)
(ANGLE DE DEPART = 45)
G80 Z50 (ANNULATION DU CYCLE)
(UTILISATION DU CENTREUR POUR EBAVURAGE DIAM B)
G0 X[#502/2]-5] Y0
Z2.
G1 Z-3. F240
G41 X[#502/2] D23
G3 I-[#502/2]
G1 G40 X[#502/2]-5]
G0 Z50.
G91 G28 Z0
N400 M6 T4 (**FORET D10.3**)
G52 X[#501/2] Y-[#501/2] Z#510 (DECALAGE ORIGINE MILIEU PIECE DESSUS)
G0 G90 G54 X0 Y0 M3 S1100
G43 H4 Z50 M8
G98 G73 Z-17 Q2.5 F150 R2 K0 (K0 -> LE PERCAGE ATTENDS UNE COORDONNEE)
G65 P4321 X0 Y0 D#503 K4 A45 (APPEL DE MACRO CLIENT DE CALCUL DES POINTS)
(MACRO CLIENT = O4321)
(DIAMETRE DE PERCAGE = #503)
(NOMBRE DE TROUS = 4)
(ANGLE DE DEPART = 45)
G80 Z50 M9 (ANNULATION DU CYCLE)
G91 G28 Z0
N500 M6 T5 (**TARAUD M12**)
G52 X[#501/2] Y-[#501/2] Z#510 (DECALAGE ORIGINE MILIEU PIECE DESSUS)
G0 G90 G54 X0 Y0
G43 H5 Z50
M29 S200

```
G84 Z-18 R2 F350 K0 (K0 -> LE PERCAGE ATTENDS UNE COORDONNEE)

G65 P4321 X0 Y0 D#503 K4 A45 (APPEL DE MACRO CLIENT DE CALCUL DES POINTS)

(MACRO CLIENT = O4321)

(DIAMETRE DE PERCAGE = #503)

(NOMBRE DE TROUS = 4)

(ANGLE DE DEPART = 45)

G80 Z50. M9 (ANNULATION DU CYCLE)

G91 G28 Y0 Z0

M30 (FIN DE PROGRAMME)

:4321(MACRO CLIENT PERCAGE DE TROUS)

#100 = #7 / 2(RAYON POS PERCAGE)

#101 = 360 / #6(INCREMENT ANGULAIRE)

#102 = #1(ANGLE DE DEPART)

#103 = 1(COMPTEUR DE BOUCLE)

WHILE[#103 LE #6] DO1 (TANT QUE PLUS PETIT QUE LE NOMBRE DE TROUS)

#110 = #24 + [COS[#102] * #100] (POSITION X)

#111 = #25 + [SIN[#102] * #100] (POSITION Y)

X#110 Y#111 (ON VA A LA POSITION)

#102 = #102 + #101 (ON INCREMENTE L'ANGLE)

#103 = #103 + 1 (ON INCREMENTE LE COMPTEUR)

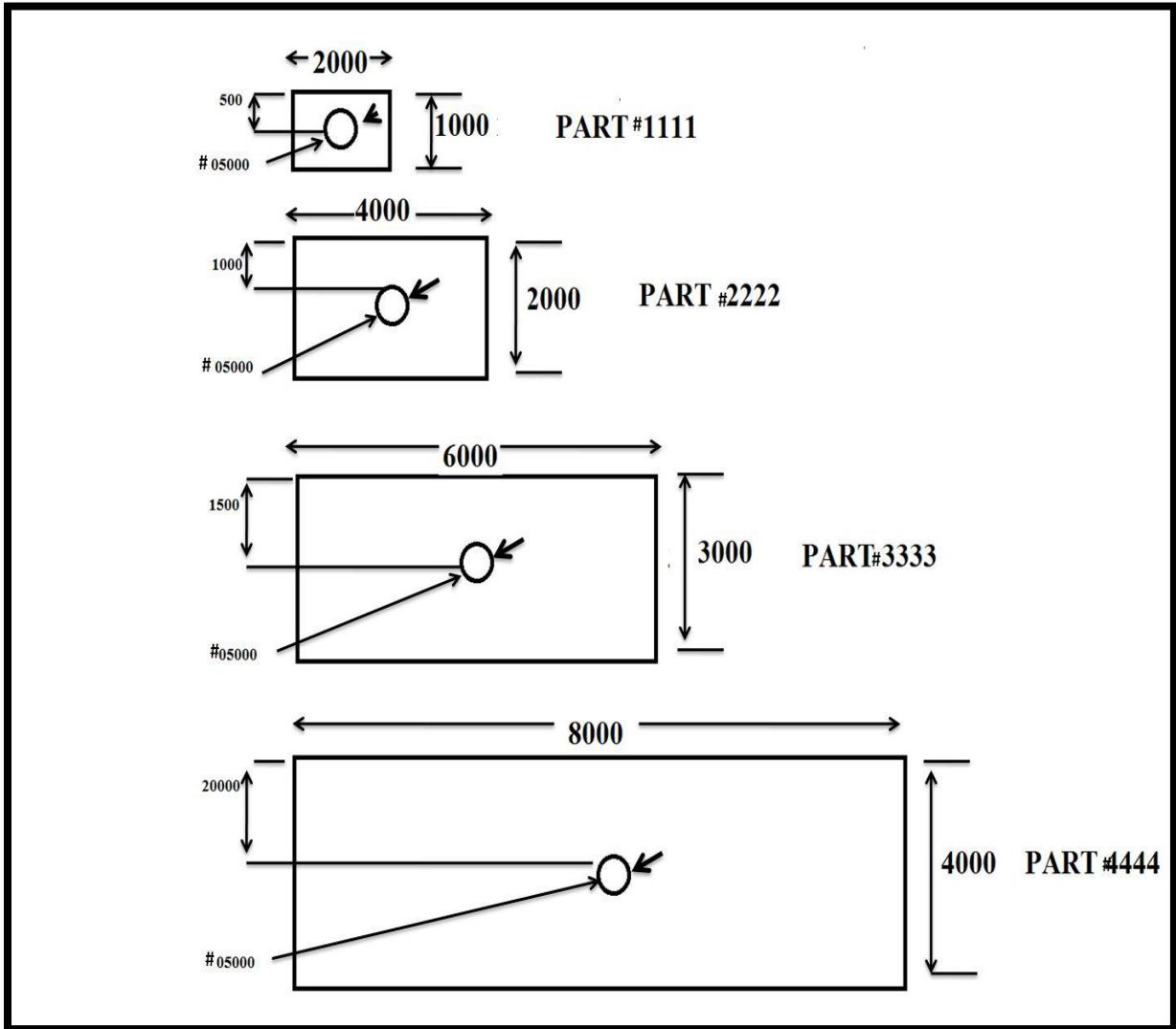
END1

M99

%
```

Exemple 03 :

Exemple simple d'un programme paramétrique pour une série de pièces. [11]



#500=1111
GOTO#500
N1111(PART NO 1111)
#100=2.
#101=1.
#102=#100/2
#103=#101/2
GOTO100

N2222(PART NO 2222)
#100=4.
#101=2.
#102=#100/2
#103=#101/2
GOTO100
N3333(PART NO 3333)
#100=6.
#101=3.
#102=#100/2
#103=#101/2
GOTO100
N4444(PART NO 4444)

#100=8.

```
#101=4.  
#102=#100/2  
#103=#101/2  
GOTO100  
N100  
(MILL OUTSIDE PROFILE)  
G0G17G40G80G94  
G91G28Z0.  
T1M6  
S2000M3  
G90G0G54X-.35Y.35  
G43H1Z1.M8  
G1Z-1.F20.  
G1Y.25F10.  
G1X#100+.25  
G1Y-#101-.25  
G1X-.25  
G1Y.35  
G1X-.35  
G0Z1.  
M5  
M9  
G91G28Z0.  
M1  
N200  
(DRILL)  
G0G17G40G80G94  
G91G28Z0.  
T2M6  
S1000M3  
G90G0G54X#102Y-#103  
G43H2Z1.M8  
G0Z.1  
G81Z-1.R.1F2.  
G80  
G0Z1.  
M5  
M9  
G91G28Z0.  
M30  
%
```

IV-CONCLUSION

Les programmes ainsi écrit comportent pratiquement autant de lignes de code que les faits de l'écrire de façon "standard". Pour une pièce plus petite, il serait même légèrement plus grand. Le gros avantage, c'est qu'en changeant 3 variables, il est possible de surfer n'importe quelle pièce de n'importe quelle dimension et de n'importe quelle surépaisseur. On peut y apporter quelques améliorations (un programme paramétré n'est jamais totalement optimisé) comme le transformer en macro client et appelé par la fonction G65 (G65 U300 V250 W3 F1875).

Donc on peut conclure que :

- La technique de programmation paramétrique étend la fonctionnalité des capacités de génération de machines à commande numérique
- De nombreuses applications basées sur la programmation paramétrique peuvent nous faire gagner du temps, réduire les efforts et minimiser le potentiel d'erreurs coûteuses.
- Bien entendu, ces avantages entraînent des économies de coûts directes. Dans de nombreux cas, le coût de la programmation paramétrique ajouté au contrôle de la machine (s'il ne s'agit pas d'une fonction standard) peut être facilement justifié à la lumière des avantages qu'il peut fournir.
- On applique le programme paramétrique dans
 - Usinage des familles de pièces
 - Nouveaux cycles d'usinage
 - Mouvement le long de courbes complexes

I-INTRODUCTION

Ce chapitre explique en détail les étapes de la simulation et la réalisation d'une pièce tout en s'initiant à créer et à gérer un programme, saisir les déplacements de l'outil, créer des contours quelconques avec le calculateur, réaliser des ébauches et des finitions de contour.

A cet effet on utilise les logiciels dédiés à la simulation d'usinage qui sont le WinCts et le MachMill.

Pour avoir des pièces mécanique à l'aide de logiciel solidworks on réalise la conception et pour la génération de programme paramétrique on utilise les codes FAO comme MasterCam, Camworks et surfcame par exemple, après cette procédure on génère le code G par le programme WinCTS et MachMill.

II-PRESENTATION DE LA PIECE

Réalisation de la pièce sur la CNC type F1 EMCO après la simulation du programme qui est généré par winCts dans le Machmill.

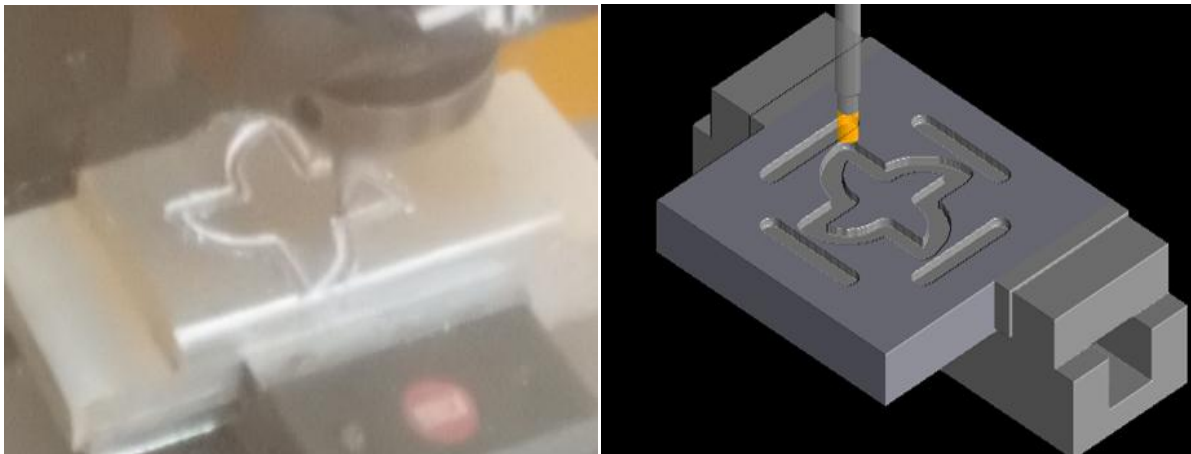


Fig 1 : la pièce réalisée par la machine et simulée par WinCTS

III. LANCEMENT DE PROGRAMME SUR WincTs

Simulation de la pièce sous WinCts

III.1 Gestion et création de programmes

Pour lancé le programme WinCts premièrement il faut faire l'accostage de la pièce par les boutons F1, F11, F11 et F3, on clic sur le bouton de verrouillage de clavier numérique après sur 9, 6, 2 pour voir les coordonnées d'origine de la machine.

Après la mise sous tension de la commande, on retrouve dans l'écran de base

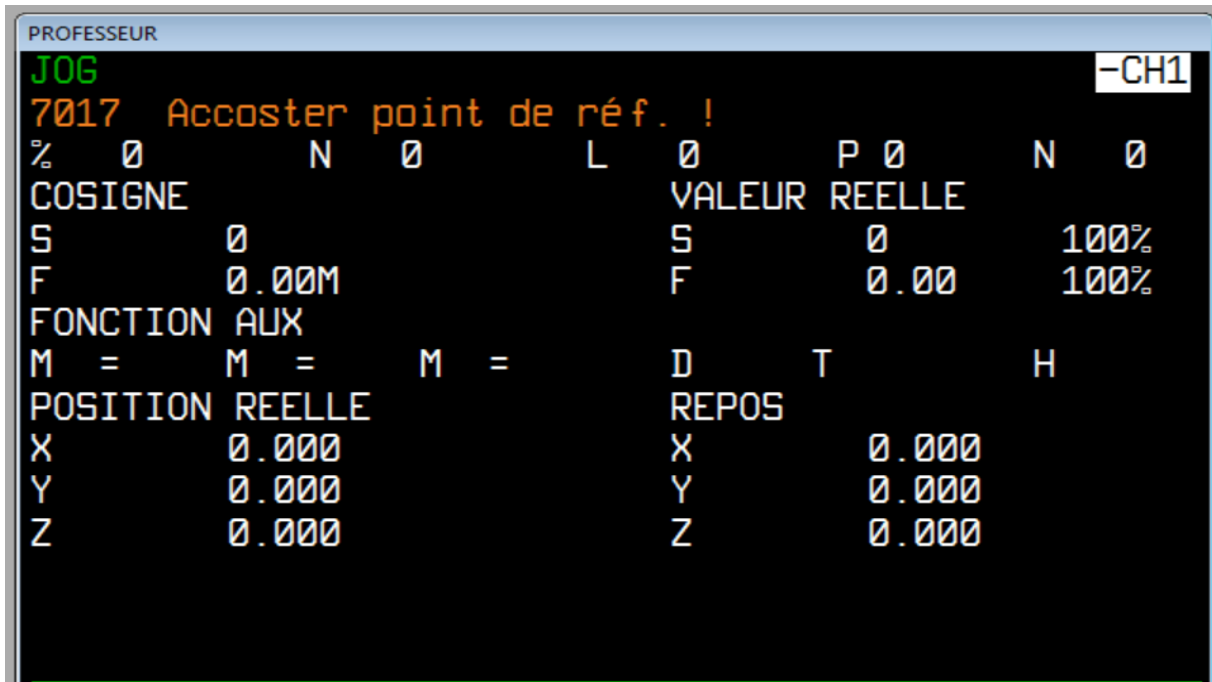
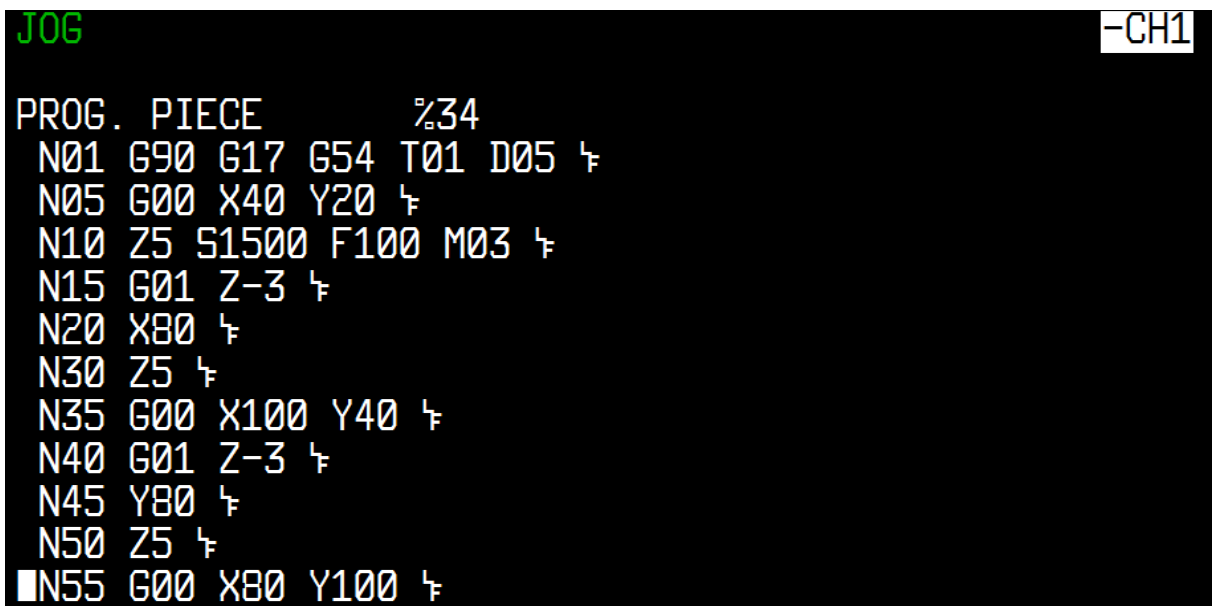


Fig 2: Interface de la fenêtre (lancement de la démarche du programme)

Lancer la commande par F3 ou bien select programme pour lancer le programme. Il faut donner le nom de programme par % suivi par des chiffres exemple (%500) et on commence la saisie du programme.



```
N60 G01 Z-3 𐀀
N65 X40 𐀀
N70 Z5 𐀀
N75 G00 X20 Y80 𐀀
N80 G01 Z-3 𐀀
N85 Y40 𐀀
N90 Z5 𐀀
N100 G00 X50 Y30 𐀀
N110 G01 Z-3 𐀀
N115 G03 X70 Y50 J20 𐀀

N115 G03 X70 Y50 J20 𐀀
N120 G01 X90 𐀀
N130 G03 X70 Y70 I-20 𐀀
N140 G01 Y90 𐀀
N150 G03 X50 Y70 J-20 𐀀
N160 G01 X30 𐀀
N170 G03 X50 Y50 I20 𐀀
N180 G01 Y30 𐀀
N190 Z5 M05 𐀀
N200 M30 𐀀
█

SELECT. F3 | GUIDE- F4 | SIMUL- F5 | PLAN F6 | 3DVIEW F7
PROG. | OPER. | LATION | |
```

Fig 3 : Le déroulement de programme (par le nom %34)

Lorsque en termine la saisie du programme, premièrement on choisi le brut et l'outil.
Pour choisir l'outil à l'aide de bouton F7 ou en sélection 3D VIEW.

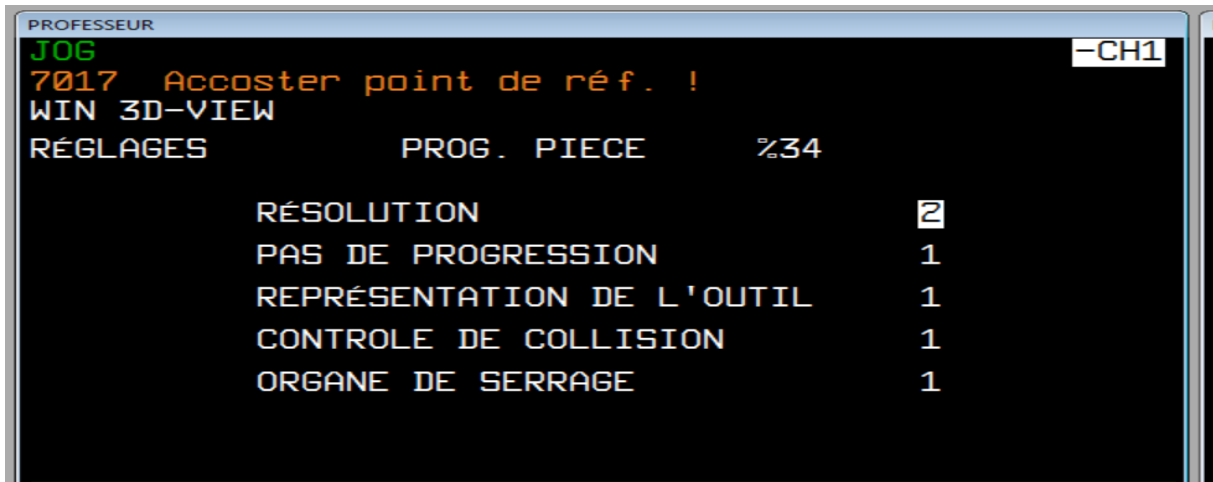


Fig 4 : déclaration des réglages concernant la réalisation

Pour afficher l'outil et le contrôle de collision, il faut les déclarés par des chiffres ; comme il est affiché sur la fig4, après on choisi le brut et l'outil. Après avoir terminer ces procédure on va lancer la simulation par la suite.

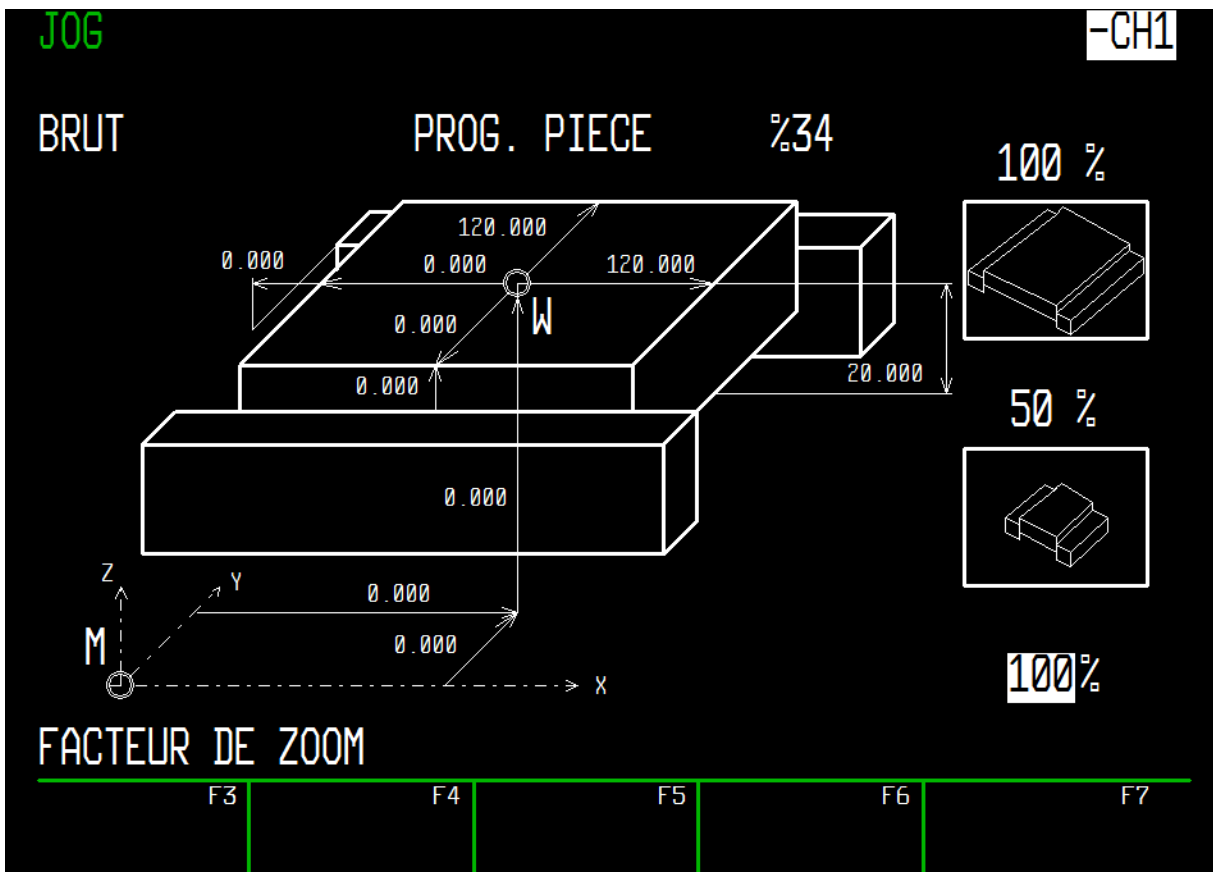


Fig 5 : choix de Brut

On a choisi le brut dans la zone a droite au dessus, par le code G. Ce choix se lance par la commande G54.

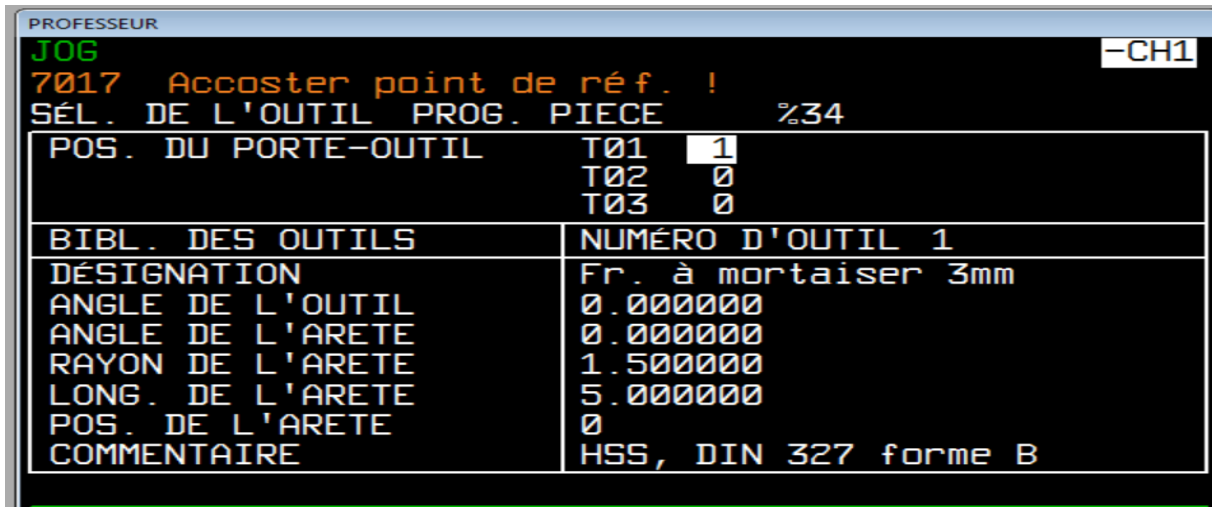


Fig 6 : choix de l'outil

Comme il est indiqué sur la fig 6, l'outil est déclaré par T1 et sa position par D1

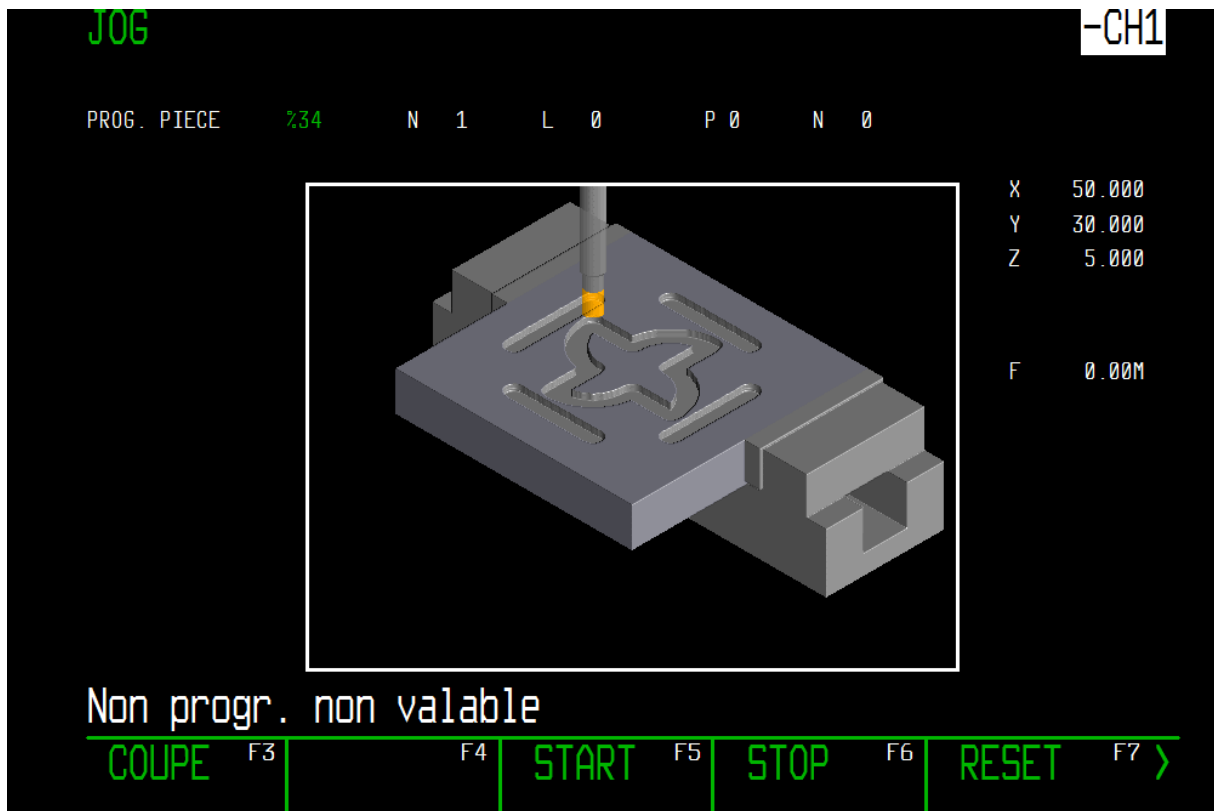


Fig 7 : la pièce simulée par WinCTS

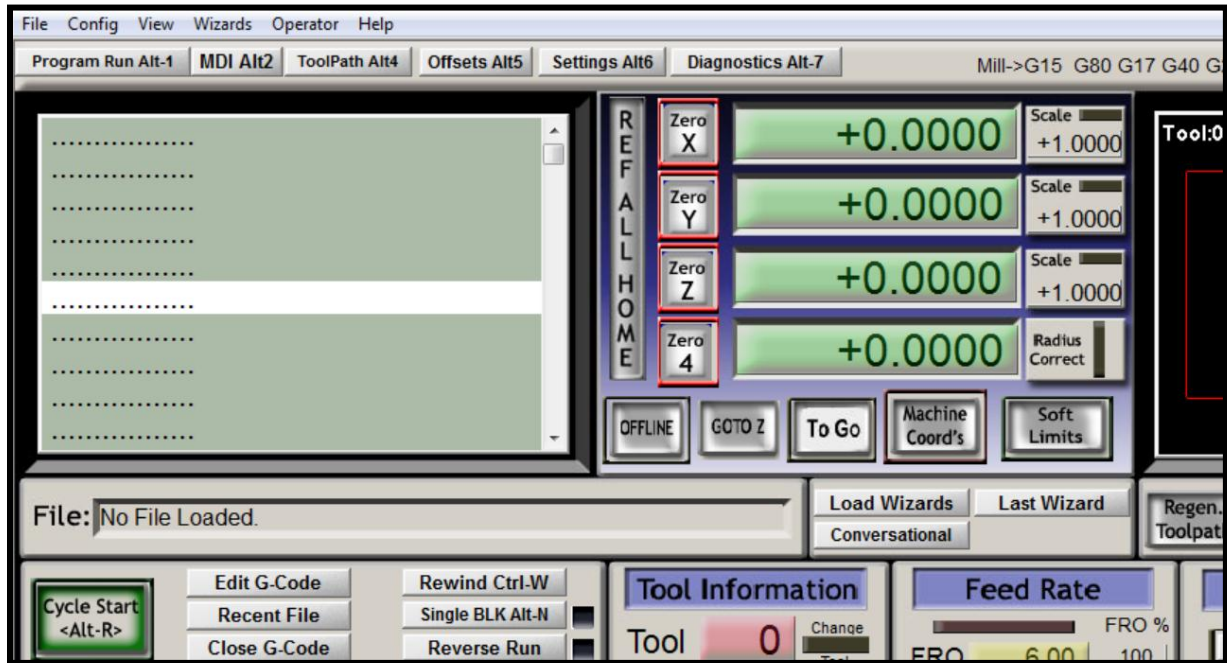
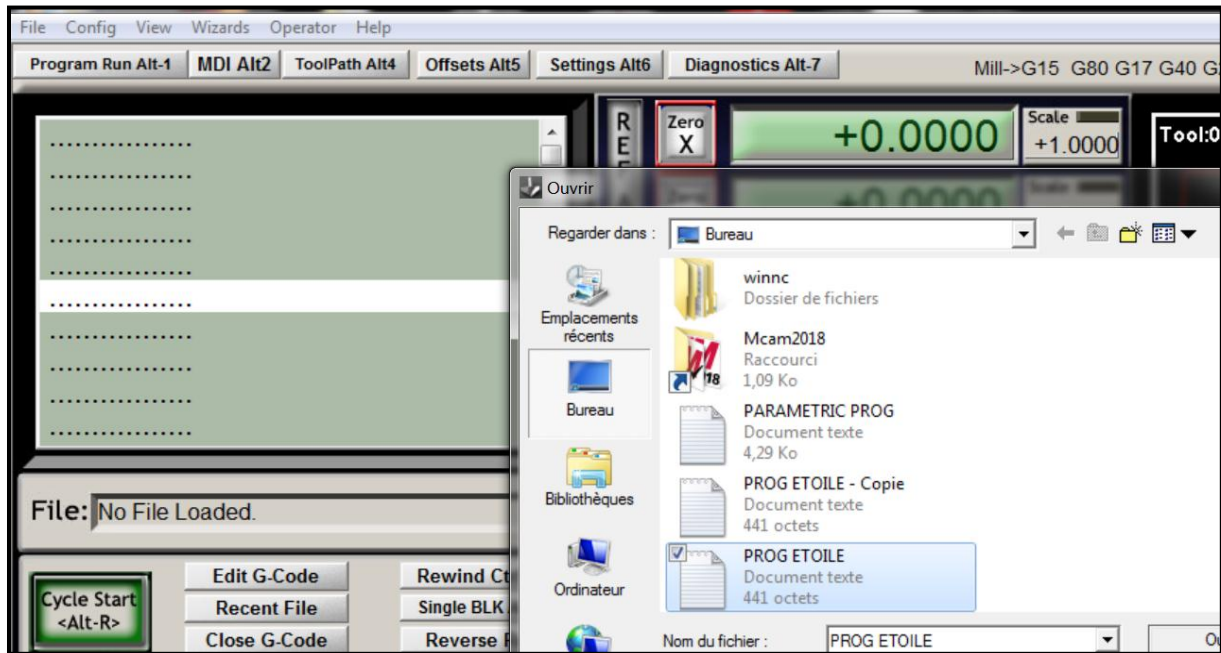


Fig 8 : l'interface du programme Machmill

MACHMILL est un logiciel de commande de la machine.

- Il assure la lecture et le traitement du G-code (le langage qui décrit les opérations à effectuer) en mouvement de la machine.
- L'interface manuelle permet la commande de la machine ainsi que la prise des origines de la pièce.

Pour exécuter le programme Code G par ce logiciel (Machmill), on fait appelle au code G par (file, load code G), sous forme de fichier bloc note.



```
%34
N5 G00 G90 T06 D06
N10 X40 Y20 S1500 F100
N15 Z5 M03
N20 G01 Z-3
N25 X80
N30 Z5
N35 G00 X100 Y40
N40 G01 Z-3
N45 Y80
N50 Z5
N55 G00 X80 Y100
N60 G01 Z-3
N65 X40
N70 Z5
N75 G00 X20 Y80
N80 G01 Z-3
N85 Y40
N90 Z5
N100 G00 X50 Y30
N110 G01 Z-3
N115 G03 X70 Y50 J20
N120 G01 X90
N130 G03 X70 Y70 I-20
N140 G01 Y90
N150 G03 X50 Y70 J-20
N160 G01 X30
N170 G03 X50 Y50 I20
N180 G01 Y30
N190 Z5 M05
N200 M30
```

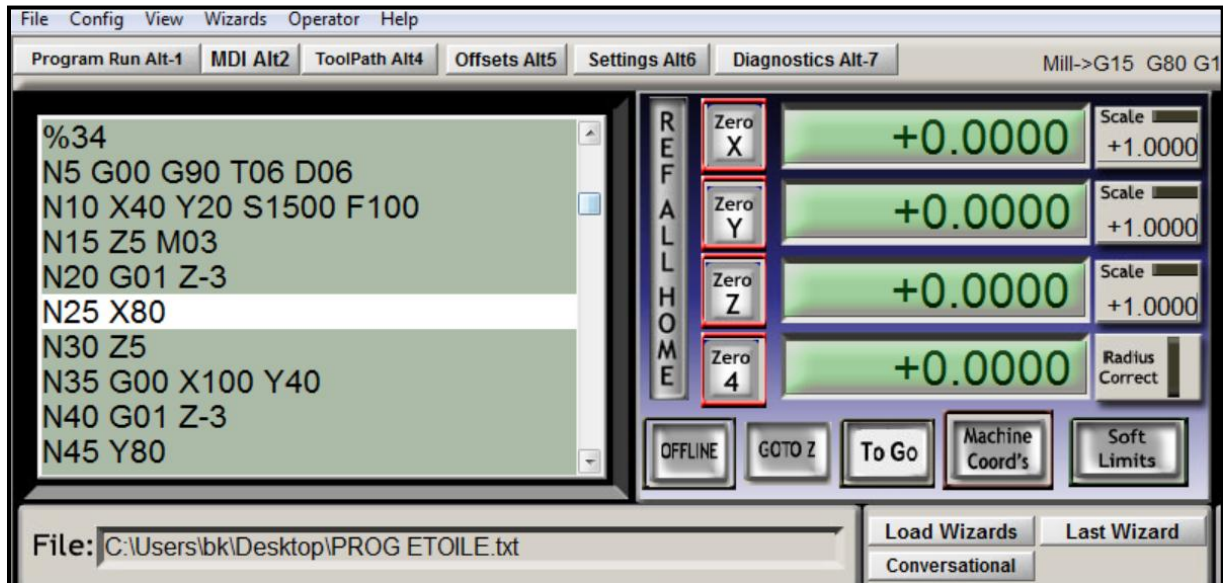


Fig 9 : le programme est sur MachMill



Fig 10: Affichage de la trajectoire à réalisée

A l'aide de la commande tallpath on démarre l'exécution de notre programme.

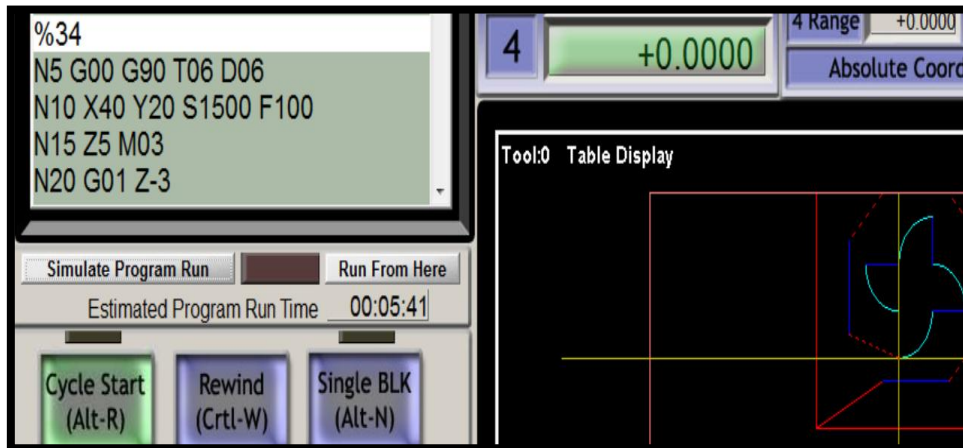


Fig 11: démarrage de l'exécution

Pour lancer l'exécution, il suffit de cliquer sur le bouton vert (Cycle Start ALT-R)

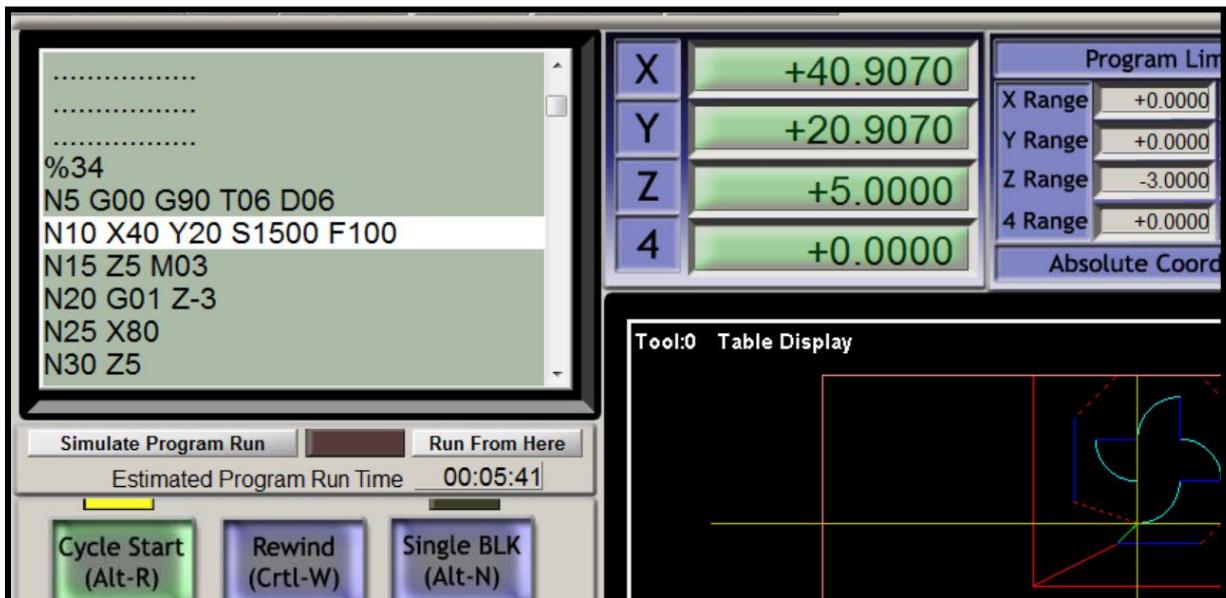
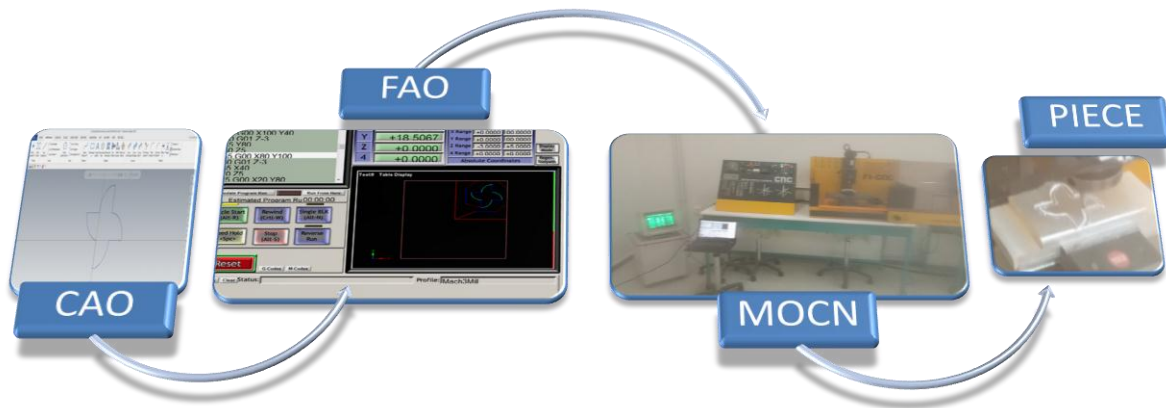


Fig 12 : l'exécution du programme

IV-REALISATION DE LA PIECE SUR LA MACHINE CNC EMCO TYPE F1 :

Ci montre un organigramme qui résume notre réalisation :



Les images suivantes montrent les différentes étapes que nous avons suivies afin d'obtenir notre pièce finale :

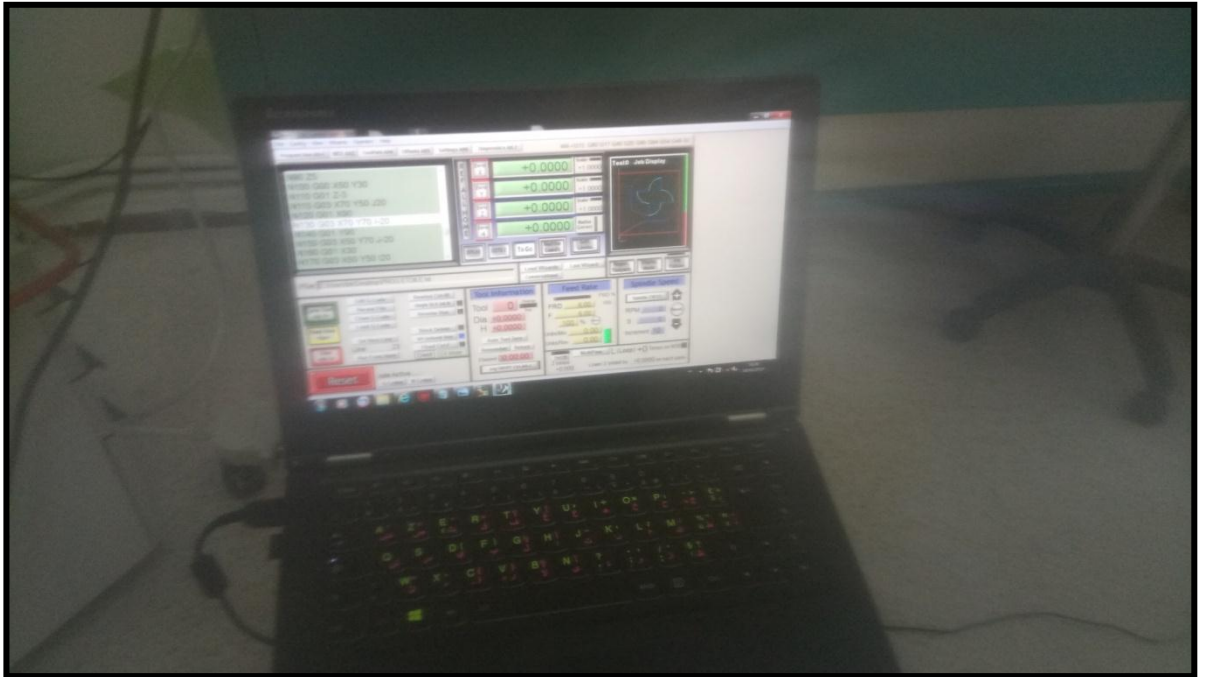
- 1. Liaison du PC -dans lequel est installé le logiciel MachMill- avec notre machine qui porte la pièce :**



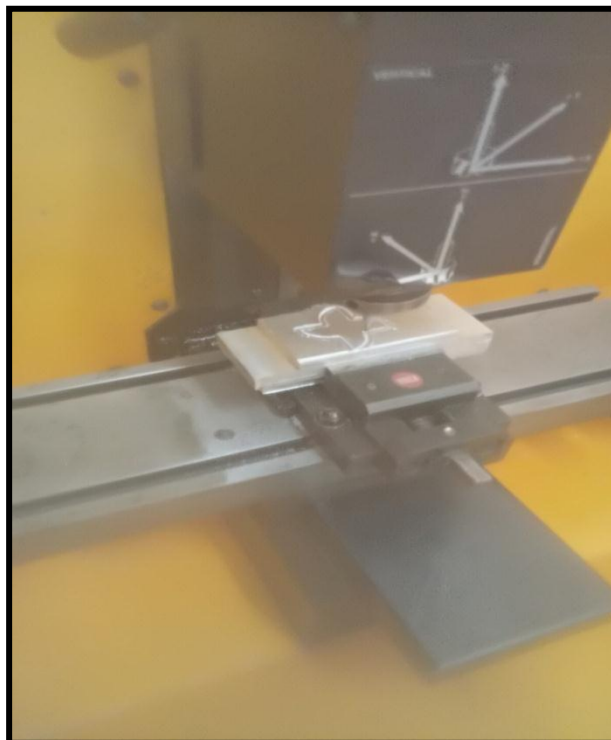
2. Fixation de la pièce brute initiale sur la machine :



3. Exécution et simulation du programme sur le logiciel MachMill :



4. Réalisation de notre pièce sur machine :



5. Obtention de la pièce finalisée :



V- CONCLUSION

WinCTS et MachMILL permettent d'élaborer le programme CN directement à partir du dessin du guidage longitudinal. La programmation ou les modifications d'une pièce. Cette dernière est particulièrement rapide en raison de la structure claire des programmes et grâce aux fonctions intégrées très performantes de création de trajectoires. Ils ont permis aussi de réaliser facilement les contours des pièces les plus complexes. Cette simulation nous a permis d'approfondir nos connaissances dans le domaine de la commande numérique.

Le processus de fabrication des pièces usinées est un processus complexe constitué d'activités numériques et physiques. La commande numérique est au cœur de ce processus, elle fait le lien entre la chaîne numérique allant de la définition de la pièce à fabriquer jusqu'à la création des trajectoires d'usinage et en terminant avec le produit physique aboutissant à l'usinage cet avantage donc de la MCN est son efficacité pour simplifier les tâches, minimiser le temps d'usinage, car le temps est très important (le temps c'est l'argent à la fabrication) et donner un superbe état de surface. La seconde partie est composée de deux grands axes :

- la simulation qui comprend plusieurs étapes et de paramètres donc des réglages s'imposent pour lancer la simulation du logiciel du dessin (WinCTS)..., et puis l'obtention du programme de la pièce à usinée, et enfin l'usinage de la pièce sur la machine.

Le travail qu'on a fait en milieu technique nous a permis de toucher de près le monde numérique appliqué, voir notre pièce dessinée se concrétise en quelques secondes ; c'était il y a un moment donné un rêve maintenant c'est devenu réalité.

Nous espérons que notre étude dans le domaine de la fabrication avancée sera un acquit pour enrichir notre expérience et sera considéré comme une motivation pour les prochaines générations de notre spécialité qu'on espère développera d'avantage cette discipline et pourrons rétablir une liaison entre le PC et la machine EMCO CNC type F1, une étape que nous n'avons pas pu la réaliser.

La perspective est ouverte pour développer des programmes d'usinages plus compétitifs tels que la programmation paramétrique et qui auront pour seul objectif un gain de temps pour diminuer le cout et augmenter la résistance des paramètres de base.

- [1] **Gilles.** Prod'homme Commande numérique des machines-outils DOC B 7 130, Techniques de l'ingénieur, Génie mécanique, Usinage, B7 130, 1995.
- [2] **RAHOU MOHAMED** Module : Atelier i et ii deuxième année EPST, Ecole préparatoire en sciences et techniques Tlemcen, 2013.
- [3] **E. Duc E. Lefur** machines-outils a commande numérique structure, modélisation et réglage, Préparation à l'agrégation de génie mécanique, 16 Septembre 1997.
- [4] **D. Duret**, « Simulation de gamme d'usinage », revue de l'ingénieur et du technicien de l'enseignement technique, n 229, 1981, pp. 34-37.
- [5] **D. Gelin, M. Vincent** « Eléments des fabrications », Edition marketing, paris, Mars 1995.
- [6] SIEMENS, SINUMERIK 840D/810D/FM-NC Notions de base Manuel de programmation documentation utilisateur Edition 12.97.
- [7] **Y. BOUTIN.** Intégration des cycles d'usinage avancés des machines-outils à commande numérique aux logiciels de FAO. ÉCOLE DE TECHNOLOGIE SUPÉRIEURE UNIVERSITÉ DU QUÉBEC. JANVIER 2002.
- [8] **SITE** : www.ange-softs.com, “ http://www.angsofts.com/programme_cnc_usinage.php “
- [9] **S. BENLAHMIDI et H. AOUCI**, Fabrication Assistée par Ordinateur. TP Production (Première année) Ecole nationale Supérieure de Technologie ROUIBA, 2013.
- [10] **Hillaire, R.** (2000). Whatever Happened to Open Controls? *Manufacturing Engineering* 124(6), pp. 80-88.
- [11] **M. Lunch** , *Parametric programming for computer numerical control machine tools and touch probes* (society of Manufacturing Engineers, 1997).
- [12] **G. Nikiel**, *Programming of CNC machine tools on the example of the Sinumerik 810D/840D, control system* (ATH University of Bielska-Biala 2004).

Résumé

L'usinage sur machine à commande numérique passe, entre autre, par l'amélioration de la programmation de celles-ci. Dans ce travail, nous avons approchés les notions fondamentales des machines-outils à commande numérique MOCN et leurs classifications et les commandes les plus utilisées. Ensuite une étude de programmation paramétrique puis une présentation des logiciels *WINCTS et MACHMILL* a été abordée pour simuler l'usinage d'une pièce. Ce travail a pour but de faire une simulation d'usinage d'une pièce sous les logiciels *WINCTS et MACHMILL* puis sa réalisation sur machine F1 CNC EMCO. L'utilisation des logiciels dédiés à la simulation d'usinage (*WINCTS et MACHMILL*) permettront la réalisation d'une pièce tout en s'initiant à créer et à gérer un programme, saisir les déplacements de l'outil, créer des contours quelconques avec le calculateur, réaliser des ébauches et des finitions de contour, et le filetage.

Mots clés : MOCN, *WINCTS, MACHMILL, CNC EMCO F1*, programmation paramétrique, G-CODE , pièce.

summary:

Numerically controlled machining involves, among other things, improving the programming of these machines. In this work, we have approached the fundamentals of MOCN numerical control machine tools and their classifications and the most used controls. Then a parametric programming study and a presentation of *WINCTS and MACHMILL* software was discussed to simulate the machining of a part. This work aims to make a machining simulation of a part under *WINCTS and MACHMILL* software and its realization on F1 CNC EMCO machine. The use of the software dedicated to the machining simulation (*WINCTS and MACHMILL*) will allow the realization of a part while being initiated to create and to manage a program, to seize the movements of the tool, to create any contours with the calculator, make blanks and contour finishes, and threading.

Key words: MOCN, *WINCTS, MACHMILL, CNC EMCO F1*, parametric programming, G-CODE, part.