



Génération de Trajectoires "On-line" pour Machine à Commande Numérique Ouverte dans un Environnement de Programmation Avancée STEP-NC

Vincent Simoes

► To cite this version:

Vincent Simoes. Génération de Trajectoires "On-line" pour Machine à Commande Numérique Ouverte dans un Environnement de Programmation Avancée STEP-NC. Génie mécanique [physics.class-ph]. Ecole Centrale de Nantes (ECN), 2015. Français. NNT: . tel-01242946

HAL Id: tel-01242946

<https://hal.science/tel-01242946>

Submitted on 14 Dec 2015

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.

Thèse de Doctorat

Vincent SIMOES

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Ecole Centrale de Nantes
sous le label de L'Université Nantes Angers Le Mans*

École doctorale : Sciences Pour l'Ingénieur, Géosciences, Architecture

Discipline : Sciences pour l'ingénieur

Unité de recherche : *École Centrale de Nantes, Institut de Recherche en Communications et
Cybernétique de Nantes (IRCCyN), UMR CNRS 6597*

Soutenue le 10/11/2015

GENERATION DE TRAJECTOIRES «ON-LINE » POUR MACHINE A COMMANDE NUMERIQUE OUVERTE DANS UN ENVIRONNEMENT DE PROGRAMMATION AVANCEE STEP-NC

JURY

Président :	Claire LARTIGUE , Professeur des Universités, LURPA, IUT de Cachan
Rapporteurs :	Emmanuel DUC , Professeur des Universités, Institut Pascal, IFMA Pierre LAGARRIGUE , Professeur des Universités, Institut Clément Ader, Centre universitaire Jean-François Champollion
Invité :	Kelvin HAMILTON , Ingénieur R&D, Delcam, Royaume-Uni
Directeur de Thèse :	Jean-Yves HASCOËT , Professeur des Universités, IRCCyN, Ecole Centrale Nantes
Co-encadrant de Thèse :	Matthieu RAUCH , Maître de Conférences, IRCCyN, Ecole Centrale Nantes

*« Ce qui fait l'homme,
c'est sa grande faculté d'adaptation. »*

Socrate

Remerciements

Je tiens à remercier tout d'abord mes encadrants pour leurs contributions et investissements ayant permis l'aboutissement de cette thèse. Mes plus sincères remerciements à Jean-Yves Hascoët, pour toute la confiance qu'il m'a accordée, en m'associant à des projets universitaires, industriels et communautaires. Il a su trouver le bon équilibre entre exigences, rigueurs et soutiens pour que je puisse donner le meilleur de moi-même. Un merci tout particulier pour ses qualités humaines, ses conseils, son efficacité, sa disponibilité et son écoute. Je remercie aussi grandement Matthieu Rauch, pour toutes ses méthodes transmises, sa disponibilité, son implication et son aide au quotidien.

Les travaux présentés dans ce mémoire ont été réalisés au sein de l'Institut de Recherche en Communications et Cybernétique de Nantes, dirigé par Michel Malabre. Je tiens à le remercier ainsi que l'ensemble de son équipe pour m'avoir accueilli au sein du laboratoire.

Mes remerciements vont également aux membres du jury. Tout d'abord à Claire Lartigue qui m'a fait l'honneur d'en avoir assuré la présidence. J'ai eu la chance de réaliser mon mémoire de master sous sa direction et je la remercie encore de m'avoir accordé sa confiance et ses recommandations pour débiter cette thèse à Nantes. Merci à Emmanuel Duc et Pierre Lagarrigue d'avoir accepté de rapporter mon travail et pour m'avoir apporté des retours très constructifs sur mon manuscrit. Merci à Kelvin de s'être déplacé depuis Amsterdam pour assister à ma soutenance et pour tous ces échanges professionnels et amicaux qui m'ont beaucoup apporté.

Cette thèse n'aurait pas eu la même consistance sans les aides précieuses de Gilles Carabin et Thierry Broquin qui ont grandement contribué à la mise en œuvre des expérimentations de mes travaux.

Je remercie aussi l'ensemble des membres de l'équipe MO2P avec qui j'ai eu le plaisir de partager et d'échanger durant ces trois années, ainsi que les personnes du « quatrième étage » et des équipes IS3P et Robotique. En particulier mes collègues David Noël, Raphael Poirée, Jad Matar, Rolland Renier, Anthony Jubien, Matthieu Quantin et Victor Godreau.

Je tiens également à remercier l'ensemble du personnel du département IPSI de l'Ecole Centrale de Nantes et en particulier l'équipe enseignante avec qui j'ai passé d'agréables moments. Merci à : Stéphane Tichadou, Olivier Legoff, Christian Le Gonidec, Jérôme Friant, Pierre Deplanche, Catherine Michel, Florent Laroche et Jean-Marc Benguigui.

Je tiens à remercier toutes les personnes qui ont contribué à l'Association des Etudiants en Doctorat : Maïté, Lauriane, Marie, Courtney, Marc, Benoit, Fabien. Un merci particulier à Annie Barreau de la Direction de la Recherche.

Un grand merci aux amis qui ont participé à mon quotidien dans cette magnifique ville nantaise : Paulo, Romain, Alexandra, Boris, Nico, Pierre, Francesc, Marie R., Marie C., Corentin, Laurent. Merci à monsieur l'artiste italien Bertrand « Sitnorelki ». Merci aux partenaires de Boxe Française. Je tiens aussi à remercier chaleureusement Catherine pour tout son soutien. Et je ne peux pas résister à l'envie de reprendre ces quelques mots d'Alain Damasio pour remercier cet ami. Merci à toi Olivier, merci « au porte-avion d'écoute et d'amitié impeccable qui tient dans l'armature de tes épaules ».

Pour finir, je remercie l'ensemble de ma famille et en particulier mes parents, à qui je suis fier de dédier cette thèse.

Table des matières

Remerciements.....	5
Table des illustrations.....	11
Abréviations.....	13
Introduction.....	17
Contexte	17
Objectifs visés	18
Plan de lecture de la thèse	18
Chapitre 1 – Évolutions de la chaîne numérique de fabrication	23
1.1. Chaîne numérique de fabrication classique : modèle courant et problématiques associées.....	24
1.1.1. Modèle courant de la chaîne numérique.....	24
1.1.2. La commande numérique au sein de la chaîne numérique.....	24
1.1.3. Problématiques de la chaîne numérique classique	26
1.2. Evolutions de la chaîne numérique et de la programmation des MOCN.....	27
1.2.1. Evolutions liées aux échanges de données au sein de la chaîne numérique ...	28
1.2.1.1. Evolutions des suites CAO/FAO	28
1.2.1.2. Code G : un format non adapté à la description de trajectoires complexes	28
1.2.1.3. Le standard STEP-NC pour étendre les possibilités d'échange de données .	30
1.2.2. Evolutions de la méthode de programmation : vers une programmation avancée du process de fabrication.....	31
1.2.2.1. Méthodes de programmation prédictives	31
1.2.2.2. Méthodes de programmation correctives	32
1.2.2.3. Intérêt des méthodes de programmation correctives	33
1.2.2.4. Nécessité de compensations géométriques	35
1.2.2.5. Principe de l'usinage adaptatif temps réel.....	36
1.2.2.6. Approche de programmation basée sur STEP-NC.....	37
1.3. Synthèse des limitations des CNs propriétaires	38
1.4. Axes de recherche.....	38

1.5. Conclusion	40
Chapitre 2 – Possibilités et intérêts de la CNO	43
2.1. Motivations et objectifs d’une nouvelle CN	43
2.1.1. Constats et objectifs	43
2.1.2. XPAIM au sein du projet européen FoFdration – Concept de Smart Machine Controller	44
2.2. Commande numérique ouverte : définition et projets mondiaux.....	45
2.2.1. Définition d’une commande numérique ouverte	46
2.2.2. Etat de l’art de différents projets d’architecture ouverte	49
2.3. LinuxCNC : définitions et intégration au sein de la XPAIM.....	51
2.3.1. LinuxCNC	51
2.3.1.1. Historique et architecture générale de LinuxCNC.....	51
2.3.1.2. Les principaux composants de LinuxCNC.....	52
2.3.2. Cohabitation de CN et intégration d’une CNO au sein de la XPAIM.....	54
2.3.3. Analyse des performances de la XPAIM et fonctionnalités avancées	57
2.3.3.1. Utilisation « classique » de LinuxCNC	57
2.3.3.2. Intégration de fonctionnalités avancées.....	63
2.4. Conclusion	64
Chapitre 3 – Vers une nouvelle approche de génération de trajectoires en ligne : Compensation temps réel au sein de XPAIM.....	69
3.1. Actions correctives en temps réel : Frontière d’étude et démarche de mise en œuvre sur XPAIM	70
3.1.1. Principe de la génération de trajectoires en ligne et frontière d’étude au sein de XPAIM	70
3.1.2. Démarche expérimentale de démonstration d’actions correctives.....	72
3.2. Développement du montage expérimental de démonstration d’actions correctives	73
3.2.1. Développement matériel du montage expérimental.....	73
a. Quelles contraintes mécaniques doit supporter le montage ?	74

b.	Quelles vitesses voulons-nous atteindre ?	75
c.	Cinématique du montage expérimental	77
3.2.2.	Développement logiciel du montage expérimental	78
a.	Fonctionnement de HAL.....	79
b.	Configuration HAL au sein de XPAIM	80
3.2.3.	Conclusion de la démarche de mise en place du montage expérimental d'actions correctives	86
3.3.	Démonstration d'actions correctives : expérimentations et résultats.....	87
3.3.1.	Démonstration de compensations en ligne : des résultats bien visibles.....	88
3.3.2.	Démonstration de compensations en ligne : cas d'un usinage dans des conditions industrielles	89
3.3.3.	Démonstration de compensations en ligne : des améliorations possibles	98
3.4.	Conclusion.....	99
Chapitre 4 –	Adéquation de XPAIM avec l'usine du futur	103
4.1.	Concept XPAIM : Un champ d'application étendu	104
4.1.1.	Les différents niveaux d'information observables en temps réel sur XPAIM et possibilités associées	104
4.1.2.	Illustration de configuration avancée au sein de LinuxCNC	106
4.2.	Usine du futur : Concepts généraux et applications.....	108
4.2.1.	Usine du futur dans sa globalité.....	108
4.2.2.	XMIS : Pour une meilleure gestion des données au sein de l'usine du futur .	110
4.2.3.	Vers une CN étendue aux nombreuses applications	112
4.3.	Conclusion.....	115
	Conclusion et perspectives.....	119
	Synthèse	119
	Perspectives.....	121
	Références bibliographiques	123

Table des illustrations

Figure 1 - Plan de lecture de la thèse	19
Figure 2 - Modèle courant de la chaîne numérique de fabrication	24
Figure 3 - La structure d'une commande numérique au sein d'une MOCN	25
Figure 4 - Evolution de la programmation des MOCN [Newman07]	27
Figure 5 - Modèles classiques de systèmes de conception (extraits et adaptés de [Laguionie10])	28
Figure 6 - Défauts liés à l'interpolation linéaire pour l'usinage de surfaces complexes [Siemens04]	29
Figure 7 - Chaîne numérique basé sur le standard STEP-NC [Laguionie10]	30
Figure 8 - Modèle de programmation avancée de MOCN par méthode prédictive [Rauch07]	32
Figure 9 - Optimisation de l'avance programmée après simulation de l'avance réelle [Rauch07]	32
Figure 10 - Modèle de programmation avancée de MOCN par méthode corrective [Rauch07]	33
Figure 11 - Contrôle adaptatif d'un procédé [Simoes14]	34
Figure 12 - Principe de la méthode de programmation ICAM [Rauch07]	35
Figure 13 - Vue d'ensemble de l'approche SPAIM pour une programmation avancée et intelligente [Laguionie10]	37
Figure 14 - Axes de recherche au sein de la chaîne numérique	39
Figure 15 - Partenaires du projet européen FoFdation	44
Figure 16 - Vision globale du concept de SMC complétée de [Rauch2014]	45
Figure 17 - Illustration PhoneBlocks (2013) et photos d'un smartphone [Ara15]	46
Figure 18 - Décomposition des fonctionnalités d'une CN [Pritschow et al. 2001]	47
Figure 19 - Critères d'ouverture d'une CN [Pritschow et al., 2001]	48
Figure 20 - Comparaison de différentes architectures de CNO [FoFdation11]	49
Figure 21 - Illustrations simplifiées des différents critères de choix de CNO	50
Figure 22 - Structure du composant EMCOT [Staraveski2009]	52
Figure 23 - Capture d'écran d'un exemple d'interface graphique (ligne de commande à gauche, visualisation des trajectoires au centre et éléments de surveillance à droite)	53
Figure 24 - Architecture du système LinuxCNC [Staraveski2009]	54
Figure 25 - Vue Globale de la cohabitation des différentes CN	55
Figure 26 - Connexions des différents logiciels et matériels à la machine Sabre	56
Figure 27 - Vue globale de l'armoire de la XPAIM	56
Figure 28 - Capture d'écran de différentes trajectoires sous LinuxCNC (Lock, Square and Line)	58
Figure 29 - Positions des axes X et Y (exécution d'un carré de 200mmx200mm)	58
Figure 30 - Vitesses des axes X et Y (exécution d'un carré de 200mmx200mm)	59
Figure 31 - Accélération des axes X et Y (exécution d'un carré de 200mmx200mm)	59
Figure 32 - Comparaison réel/simulé : Vitesse d'avance (exécution d'un zig-zag sur X en arrêt précis)	59
Figure 33 - Comportement cinématique en simulation : profil d'accélération	60
Figure 34 - Capture d'écran de l'outil oscilloscope de Fidia	60
Figure 35 - Positions acquises sous FIDIA en X et Y (carrée de 200mmx200mm)	61
Figure 36 - Exemple d'acquisition pour un programme identique (carré 200mmx200mm) sous LinuxCNC et Fidia.	61
Figure 37 - Résultats d'essais Ballbar, pour F=1000mm/min, Rayon du cercle 10mm, sous FIDIA (à gauche) et LinuxCNC (à droite)	62
Figure 38 - Ensemble de photos de la plateforme IRCCyN en fonctionnement	62
Figure 39 - Pièce 4-vagues (à gauche), CAO de la 4ème vague sous Rhino (au milieu) et trajectoires sous LinuxCNC	63
Figure 40 - Simulation sur une vague en interpolation linéaire et au format NURBS	63
Figure 41 - Usinage d'une trajectoire B-spline sur la plateforme Sabre	64
Figure 42 - Synthèse des méthodes de programmation avancée usuelles sur CN propriétaire	70
Figure 43 - Schéma de principe de l'approche de génération de trajectoires en ligne au sein de XPAIM	71
Figure 44 - Schéma de principe du montage expérimental au sein de la MOCN	72
Figure 45 - Etapes de développements envisagées du montage expérimental	73
Figure 46 - Trace suivant Y dû au temps de cycle (accélération infinie)	76
Figure 47 - Trace suivant Y dû au temps de rattrapage (accélération 1m/s ²)	76
Figure 48 - Compensation de perturbation transversale durant une finition	77

Figure 49 - Schéma de fonctionnement du montage expérimental	77
Figure 50 - Vue d'ensemble du montage expérimental de perturbation	78
Figure 51 - Architecture de LinuxCNC avec zone temps réel [Staroveski2009]	79
Figure 52 - Différents éléments de HAL	80
Figure 53 - Configuration de l'asservissement de l'axe X de la plateforme CNO	81
Figure 54 - Illustration du câblage logiciel de 3 trois axes asservis	82
Figure 55 - Illustration de la configuration simulée du déplacement d'un axe X.....	83
Figure 56 - Ajout d'éléments permettant une compensation temps réel au sein de la configuration d'un axe asservi	84
Figure 57 - Visualisation de la sortie PID (Vitesse) à l'oscilloscope lors de l'entrée manuelle d'une perturbation fictive de 0,5mm	85
Figure 58 - Acquisition de la position du codeur de l'axe X et trace hypothétique sur la pièce avec compensation (en violet) et sans compensation (en rouge)	85
Figure 59 - Acquisitions de la position des codeurs et vitesse associée	86
Figure 60 - Etapes de développements effectués du montage expérimental	87
Figure 61 - Montage au sein de la plateforme XPAIM avec système de manipulation hors machine.....	88
Figure 62 - Réglage de l'échelle de conversion des codeurs.....	88
Figure 63 - Effet visible de la compensation : carré de 100mmx100mm perturbé-non compensé (a) et perturbé-compensé (b)	89
Figure 64 - Profil de perturbations appliqué durant l'usinage en position fixe de l'outil	90
Figure 65 - Usinage perturbé-compensé à gauche et usinage non perturbé à droite	90
Figure 66 - Acquisition des positions outils durant la perturbation de la position outil fixe	91
Figure 67 - Mesure des profils de démonstration sur MMT	91
Figure 68 - Affichage des différents profils obtenus : sans perturbations (en vert), avec perturbations (mesure MMT en bleu et acquisitions en ligne en rouge)	92
Figure 69 - Ecart du profil acquis (en rouge) et mesuré (en bleu) par rapport au profil idéal (ligne horizontale recalée à 0).....	92
Figure 70 - Usinage perturbé-compensé (face a) et usinage non perturbé (face b)	93
Figure 71 - Profil des perturbations (a) et aperçu du profil supposé sans compensation (b).....	93
Figure 72 - Comparaison des profils dans le cas d'un usinage sans perturbations (en vert) et avec perturbations (en rouge).....	94
Figure 73 - Mise en évidence de traces liées aux perturbations (accentuation de facteur 100)	94
Figure 74 - Illustration des résultats d'écarts en fonction du profil de perturbation sur une portion du profil	95
Figure 75 - Mise en évidence de traces liées aux perturbations ayant une vitesse inférieure à 500 mm/min.....	95
Figure 76 - Visualisation des différents profils d'usinage (mesure en bleu, acquisition en rouge, référence en vert).....	96
Figure 77 - Similarité entre profil mesuré et profil acquis (compensation suivant X)	96
Figure 78 - Similarité entre profil mesuré et profil acquis (compensation suivant Y).....	97
Figure 79 - Contribution de ce chapitre dans une vision plus globale faisant intervenir le langage STEP-NC.....	99
Figure 80 - Illustrations des défauts observables et compensables en temps réels	104
Figure 81- Capitalisation, prédiction et observation du process au sein de la chaîne numérique	104
Figure 82 – Différents niveaux d'information exploitables	105
Figure 83 - Vue d'ensemble des possibilités de XPAIM.....	106
Figure 84 – Configuration HAL avancée de l'axe X de la plateforme XPAIM	107
Figure 85 - Ensemble des informations liés à la fabrication d'un produit [Simoes15]	109
Figure 86 - XMIS au centre des acteurs de l'usine du futur [Simoes15]	110
Figure 87 - Echange de données au sein de XMIS [Laguionie11]	110
Figure 88 – Illustration d'un concept d'usine globalisé basé sur XMIS : « Build Anywhere » [Simoes15]	111
Figure 89 - Chaîne numérique de fabrication au sein de XMIS [Simoes15]	112
Figure 90 - Vue d'ensemble des applications de la CN étendue [Laguionie10]	113
Figure 91 - Concept MPS qui combine des modules de production pour s'adapter à la pièce à produire [Friedrich14].....	113
Figure 92 - Concept de fabrication multi-process à l'aide de STEP-NC [Laguionie10]	114
Figure 93 - Vue globale des outils de la plateforme SPAIM	114

Abréviations

CAO Conception Assistée par Ordinateur

CAPP Computer Aided Process Planning

CFAO Conception et Fabrication Assistées par Ordinateur

CIM Computer Integrated Manufacturing

CL Cutter Location

CN Commande Numérique

CNO Commande Numérique Ouverte

CRM Customer Relationship Management

DCN Directeur de Commande Numérique

ERP Enterprise Resource Planning

FAO Fabrication Assistée par Ordinateur

ICAM Intelligent Computer Aided Manufacturing

ICT Information and Communication Technologies

IHM Interface Homme-Machine

IRCCyN Institut de Recherche en Communications et Cybernétique de Nantes

MMT Machine à Mesure Tridimensionnelle

MO2P Modélisation et Optimisation des Process de Production

MOCN Machine-Outil à Commande Numérique

MRP Manufacturing Resource Planning

NCK Numerical Control Kernel

NIST National Institute for Standardization

NURBS Non Uniform Rational Basis Splines

PAU Process Analysis Unit

PCU Process Control Unit

PDM Product Data Management

PDU Process Diagnosis Unit

PLC Programmable Logic Control

PLM Product Lifecycle Management

PPU Project Production Unit

PU Planning Unit

SCM Supply Chain Management

SPAIM STEP NC Platform for Advanced and Intelligent Manufacturing

STEP NC STEP compliant Numerical Control

STEP Standard for the Exchange of Product data model

XCAD eXtended Computer Aided Design ou CAO étendue

XCNC eXtended Computer Numerical Control ou CN étendue

XMIS eXtended Manufacturing Integrated System

XPAIM eXtended Plateform for Advanced and Intelligent Manufacturing

Introduction

Introduction

Contexte

L'industrie manufacturière joue un rôle prépondérant sur le niveau de vie et la croissance économique d'un pays. Le marché actuel est caractérisé par une forte compétitivité et un besoin toujours croissant de produits manufacturés, qui sont plus variés et ont des temps de cycle de vie de plus en plus courts. Pour faire face à cette demande, tout en restant concurrentiel, la solution retenue par certains industriels est de délocaliser leur production afin d'employer une main d'œuvre à bas coût.

Afin de contrer ce phénomène, l'Europe s'est lancée dans une dynamique de recherche et d'innovation à travers de nombreux financements publics et privés autour de l'usine du futur. L'objectif établi est de capitaliser les avancées technologiques à la fois matérielles et logicielles de ces dernières années, ainsi que d'innover sur les processus de fabrication, afin de rendre la production plus flexible, réactive et moins couteuse.

Dans ce contexte, la chaîne numérique de fabrication, allant de la conception d'un produit à sa réalisation, joue un rôle prépondérant. Au sein de cette chaîne, de nombreux progrès technologiques sont observables concernant la structure des machines, les suites de logiciels CAO/FAO, ainsi que sur la mise en œuvre de procédés innovants, mais relativement peu concernent la programmation destinée aux machines-outils à commande numérique (MOCN). Le contexte industriel actuel est caractérisé par le monopole d'utilisation d'une chaîne numérique cloisonnée et hétérogène CAO/FAO/CN qui laisse peu de possibilités d'optimisation de fabrication. La commande numérique qui est au cœur de cette chaîne n'est bien souvent qu'une « simple exécutante ». Une volonté forte est alors accordée à l'extension du rôle de la CN au sein de l'usine dite du futur.

Objectifs visés

Les travaux présentés dans ce manuscrit s'inscrivent dans le contexte de la programmation des machines-outils à commande numérique. Notre objectif est d'explorer et mettre en œuvre une nouvelle méthode de génération de trajectoires en ligne afin d'améliorer la productivité et la conformité des pièces manufacturées au sein de l'usine du futur. Pour cela, nous identifions différentes contributions portant sur :

- ✓ le développement d'une méthode de génération de trajectoires permettant de compléter les méthodes de programmations avancées actuelles (prédictives et correctives),
- ✓ la mise en œuvre d'une plateforme intelligente de fabrication, basée sur une Commande Numérique Ouverte, en alternative aux CNs propriétaires,
- ✓ les perspectives d'évolution des méthodes de programmation au sein de l'usine du futur.

L'ensemble des contributions visent à donner un rôle plus décisionnel à la commande numérique et se veut applicable à tout type de procédé.

Plan de lecture de la thèse

Au cours du premier chapitre, nous présentons les composants des modèles usuels de la chaîne numérique de fabrication, en particulier la commande numérique. En réponse aux problématiques liées à une chaîne numérique classique, nous détaillons les principales évolutions des méthodes de programmation des MOCNs. Nous montrons un manque au sein des méthodes de programmation correctives, permettant une réelle fabrication adaptative. Ce manque est dû aux limitations de la commande numérique dont nous présentons une synthèse. Ceci nous amène alors à définir les différents axes de recherche développés dans le cadre de nos travaux.

Le deuxième chapitre nous permet de définir une alternative aux CNs dites propriétaires. Nous définissons tout d'abord un ensemble d'objectifs à remplir à l'aide d'une plateforme avancée développée dans le cadre de nos travaux et nommée XPAIM (eXtended Platform for Advanced and Intelligent Manufacturing). Nous faisons le choix de développer une Commande Numérique Ouverte (CNO) afin de répondre à ces objectifs. Les différents critères définissant une CNO sont alors présentés ainsi qu'une synthèse des projets existants. Cette synthèse débouche sur le choix d'une CNO : LinuxCNC, qui servira de base de développement dans le cadre de nos travaux. A la suite d'une description de LinuxCNC, nous présentons son intégration au sein d'une machine de type industriel. Enfin nous démontrons son entière fonctionnalité et les possibilités d'algorithmes avancés.

Le troisième chapitre est axé sur les possibilités et la mise en œuvre d’actions correctives en temps réel au sein de la plateforme XPAIM. Nous présentons tout d’abord une nouvelle approche de génération de trajectoires en ligne. Sur les bases de cette approche générique, une démarche expérimentale de démonstration de compensation de perturbations en cours d’usinage est définie. Après une étude du développement matériel et logiciel nécessaire, les résultats de mise en œuvre de cette démonstration sont présentés.

Le dernier chapitre démontre l’adéquation de nos travaux avec le contexte de l’usine du futur. Nous présentons tout d’abord un champ d’applications élargi qui découle de nos travaux au sein de la plateforme XPAIM. Ces applications ne sont possibles qu’avec une bonne gestion collaborative de l’ensemble des données d’une usine. Cette notion d’échange de données est au cœur des courants de pensée de l’usine du futur, dont nous établissons une synthèse. Nous montrons enfin que nos développements ont leur place au sein de l’ensemble des modules présents dans une CN étendue. Ce plan de lecture est synthétisé sur la Figure 1.

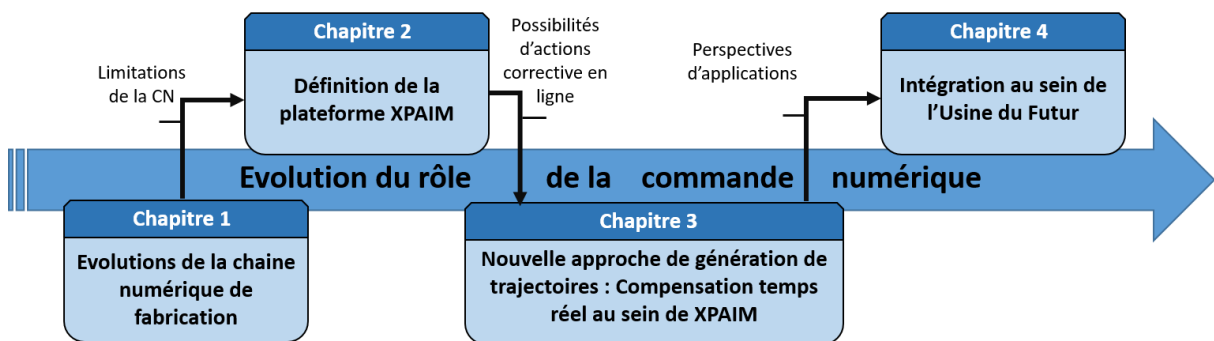


Figure 1 - Plan de lecture de la thèse

Chapitre 1

Évolutions de la chaîne numérique de fabrication

Chapitre 1 – Évolutions de la chaîne numérique de fabrication

Ce premier chapitre a pour objectif de poser le cadre de nos travaux et de définir les axes de recherche qui sont présentés par la suite dans ce manuscrit. Nous présentons dans un premier temps la chaîne numérique de fabrication classiquement utilisée dans l'industrie. Nous détaillerons les composants et les échanges d'informations au sein du modèle courant en nous focalisant particulièrement sur le rôle de la commande numérique (CN). Nous soulignerons les problèmes inhérents à la chaîne numérique classique dans laquelle la CN est une simple exécutante.

Afin de répondre à des besoins de productivité croissants ainsi qu'à l'avancée technologique à la fois matérielle et logicielle, l'interaction des composants de la chaîne numérique et les méthodes de programmation ont évoluées. Nous présentons quelques évolutions liées aux échanges de données ainsi que des nouvelles méthodes de programmation avancées (prédictive et corrective). Ces méthodes permettent d'enrichir la définition purement géométrique du parcours d'usinage et donc d'améliorer les chances de conformité de la pièce, mais elles restent encore limitées dans la prise en compte d'informations au cours de l'usinage et l'adaptation du parcours.

Nous présentons alors une alternative aux approches conventionnelles de programmation pouvant apporter une certaine flexibilité au processus de réalisation d'une pièce : la fabrication adaptative. Cependant, cette méthode s'avère encore difficile à mettre en place, essentiellement dû aux limitations de la CN.

À la suite de cela, nous faisons donc une synthèse des limitations des CN classiques limitant les évolutions de la chaîne numérique. Après avoir présenté les différentes évolutions de la chaîne numérique et les verrous liés, la dernière partie de ce chapitre est destinée à définir les axes de recherches présentés dans ce manuscrit.

1.1. Chaîne numérique de fabrication classique : modèle courant et problématiques associées

1.1.1. Modèle courant de la chaîne numérique

La chaîne numérique de fabrication réunit l'ensemble des étapes de réalisation d'un produit, allant de l'idée à la production. La suite des outils numériques au sein de cette chaîne peut être représentée suivant le modèle courant représenté sur la Figure 2.



Figure 2 - Modèle courant de la chaîne numérique de fabrication

En reprenant l'ordre de description de ce modèle, nous avons tout d'abord l'étape de conception, qui à l'aide de logiciel de Conception Assistée par Ordinateur (CAO) permet de modéliser la maquette numérique du produit à fabriquer. Cette maquette est ensuite utilisée pour construire les parcours de fabrication et définir les paramètres opératoires à l'aide d'un logiciel de Fabrication Assistée par Ordinateur (FAO). Ce logiciel peut lire le fichier de définition de la maquette CAO après conversion d'un format natif (propre au logiciel de CAO) ou standard (IGES, SET, STEP...) dans un format adapté. Le logiciel FAO envoie ensuite un fichier de localisations des points de passage de l'outil CL (Cutter Location) à un post-processeur. Celui-ci, traduit le fichier CL dans le langage de programmation de la majorité des machines-outils qui est le code G (ISO 6983). Ce fichier est alors interprétable par la commande numérique qui pilote les mouvements des axes de la machine. Ce dernier maillon étant essentiel, nous le présentons de manière plus approfondie ci-après.

1.1.2. La commande numérique au sein de la chaîne numérique

La commande numérique pilote de très nombreuses technologies de fabrication et sert de lien entre la partie numérique et physique du processus de fabrication : la CN est en quelque sorte l'interface numérique/électromécanique d'une MOCN [Laguionie11]. La fonction principale d'une CN est de contrôler les déplacements des axes de la machine. Cette fonction fait intervenir plusieurs composants internes à la CN, mais aussi externes, en lien avec la MOCN et son environnement (Figure 3).

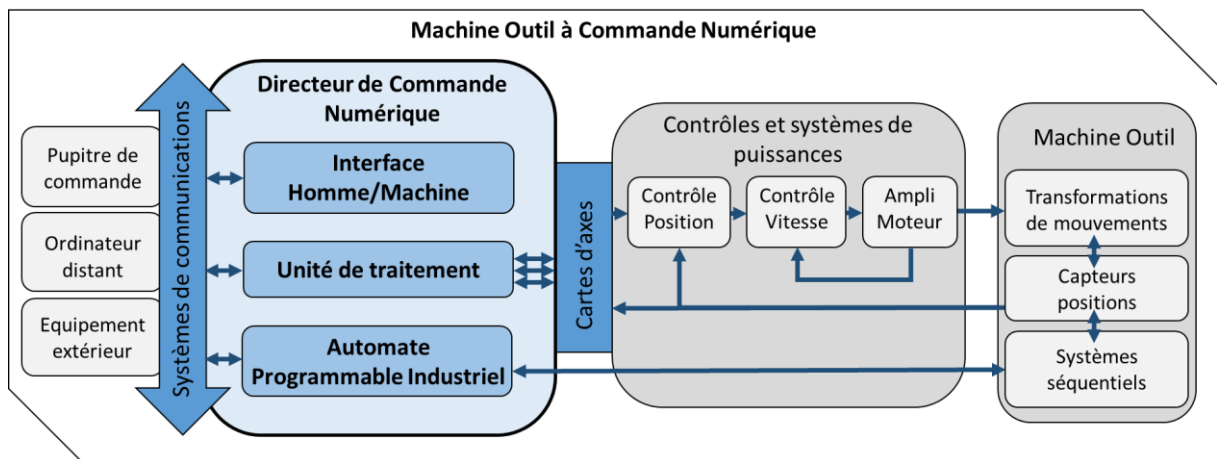


Figure 3 - La structure d'une commande numérique au sein d'une MOCN

Les composants principaux de la CN sont réunis au sein du Directeur de Commande Numérique (DCN). Précisons que l'appellation de CN fait référence à ce qui englobe la programmation de commande numérique liée à une architecture machine, tandis que le terme de DCN fait référence aux composantes physiques de la CN. Les fonctions principales des différents composants sont les suivantes :

- Unité de traitement : Elle est composée d'un microprocesseur et de mémoires. La fonction principale est d'interpréter un programme et de générer des consignes de positions (génération/interpolation de trajectoires).
- Interface homme/machine (IHM) : Généralement un pupitre de commande composé d'un clavier et d'un écran. Cette interface destinée à l'opérateur permet de sélectionner et modifier les programmes, saisir les paramètres de réglages et définir les modes d'utilisation de la machine.
- Automate Programmable Industriel (API) : Il permet de gérer les fonctions annexes de la MOCN relevant de l'automatique séquentielle : Changement d'outil, lubrification, sens de rotation de la broche, signaux d'arrêt d'urgence, ouverture des portes, sécurités, etc...

En lien direct avec ces composants, nous retrouvons différents systèmes de communication (carte de communication PC, protocoles standard de transmission type Profibus, CAN Bus...) qui permettent d'échanger des informations avec le matériel utilisable par l'opérateur ainsi qu'entre chaque composant du DCN. Le lien entre l'unité de traitement et les systèmes de puissances de la machine-outil est réalisé à l'aide de cartes d'axes. Celles-ci font parties intégrantes de l'asservissement numérique de la position des axes de la machines, qui réunit les fonctions d'amplifications, de comparaisons et de corrections. A la

suite des étapes de contrôles, une consigne de tension est envoyée aux moteurs qui déplacent les axes de la machine-outil.

Connaissant chacun de ses composants, nous allons maintenant aborder les problématiques de la chaîne numérique dite classique.

1.1.3. Problématiques de la chaîne numérique classique

Tout d'abord, la chaîne numérique classique a pour défaut d'avoir une structure d'échange de données unidirectionnelle. Toutes les données liées au produit naviguent de la conception à la production avec bien souvent aucun retour possible. Si une modification du parcours est opérée au pied de la machine, elle ne sera pas répercutée sur le modèle FAO de la pièce. Cela peut être le cas aussi entre FAO et CAO. Ceci pose donc un problème lorsqu'une amélioration de la trajectoire est identifiée en aval, il est nécessaire de revenir en amont et de modifier les fichiers en question. Le retour d'expérience et la capitalisation d'observations faites en cours de production sont difficilement possibles. En lien avec ce problème, les données qui transitent au sein de la chaîne numérique sont hétérogènes. Bien que des formats d'échange standardisés existent entre chaque module, cela n'empêche pas l'utilisation de format natif aux logiciels de CAO et FAO, permettant d'avoir plus de fonctionnalités. Des conversions de données sont alors nécessaires entre chaque étape et introduisent des écarts ainsi que des pertes d'informations entre la maquette numérique et l'usinage de la pièce [Wang06]. De plus, un même fichier de programme peut difficilement être utilisé sur différentes machines, car un post-processeur propre à un couple CN/MOCN est nécessaire pour traduire les données de positions d'outil venant de la FAO [Hardwick06]. Ce post-processeur marque une rupture au sein de la chaîne numérique. Des évolutions liées aux échanges de données et au langage de programmation code-G sont présentées à la suite de ce chapitre (section 1.2.1).

Les autres problématiques de la chaîne numérique classique sont liées à sa méthode de programmation actuelle. Ces méthodes se basent seulement sur des données géométriques (trajectoires, décalages outils ...) et cinématiques (interpolations, contraintes machines en vitesses et accélérations ...). Il n'y a pas de lien avec le process et le comportement réel de la machine. Seules les compétences et l'expérience du programmeur sont prises à parties pour améliorer le temps de fabrication, le coût de production et la conformité de la pièce. Dans l'ensemble, la commande numérique est réduite au rôle de simple exécutante. Nous verrons dans la suite de ce chapitre les principales évolutions (section 1.2.2).

1.2. Evolutions de la chaîne numérique et de la programmation des MOCN

Nous avons vu que la chaîne numérique de fabrication classique est sujette à de nombreuses problématiques. Fort heureusement, celle-ci est en évolution constante depuis les premières MOCN de la fin des années 1950, comme nous pouvons le voir sur la figure 4.

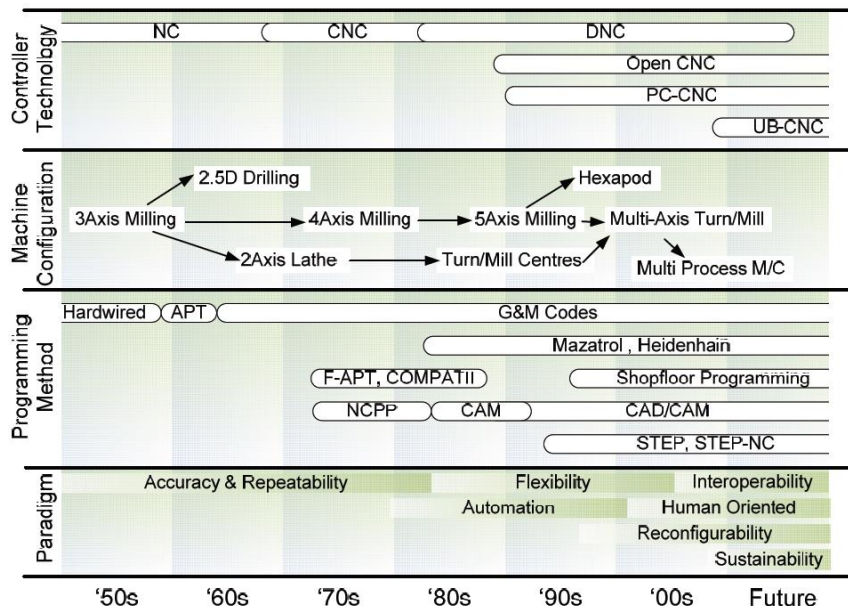


Figure 4 - Evolution de la programmation des MOCN [Newman07]

La complexification des structures des machines a évolué avec l'augmentation des capacités informatiques des outils de conceptions et des contrôleurs. Cependant les méthodes de programmation avancée sont toujours peu implantées dans l'industrie (le code-G est ancré depuis les années 1960) et les différents paradigmes (d'interopérabilité, de durabilité, ...) intrinsèquement conceptuels restent encore peu exploités. Nous allons nous focaliser sur les évolutions liées aux échanges de données au sein de la chaîne numérique ainsi que celles liées aux méthodes de programmation.

1.2.1. Evolutions liées aux échanges de données au sein de la chaîne numérique

1.2.1.1. Evolutions des suites CAO/FAO

Pour pallier le problème d'échange de données entre les différents logiciels de CAO et FAO, les fournisseurs de logiciels proposent des systèmes intégrés CAO/FAO permettant un fonctionnement centralisé autour de fichiers aux formats propriétaires. Pour compléter ce fonctionnement, il existe aussi un modèle collaboratif de conception utilisant un format neutre d'échange (IGES, STEP, ...) qui permet de résoudre les problèmes d'incompatibilité entre systèmes concurrentiels [Xu05] [Laguionie10]. Ceci peut être un atout concernant les projets de grande envergure. Le fonctionnement global et la complémentarité des deux systèmes sont illustrés sur la Figure 5.

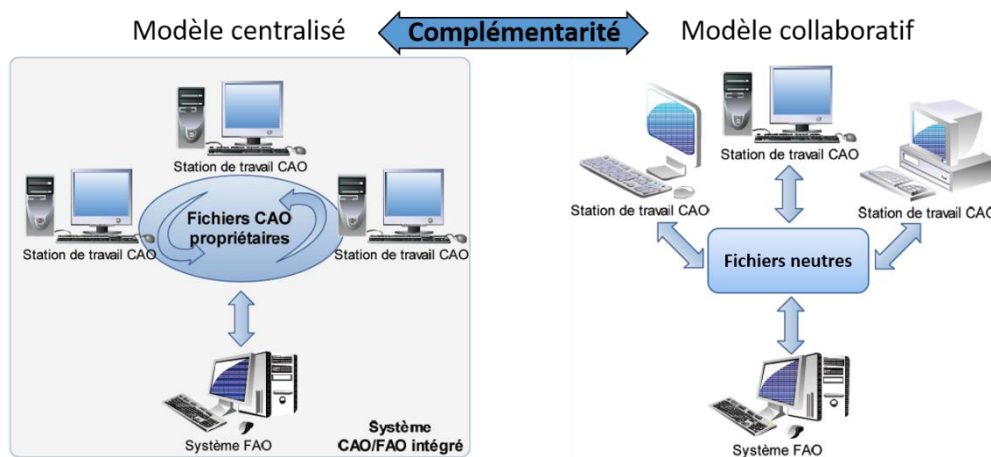


Figure 5 - Modèles classiques de systèmes de conception (extraits et adaptés de [Laguionie10])

Ces systèmes intégrés estompent les frontières (et donc l'hétérogénéité) des premiers maillons de la chaîne numérique. A titre d'exemple, Siemens peut maintenant intégrer l'ensemble des maillons de la chaîne de fabrication depuis le rachat de l'éditeur FAO NX CAM en 2007. Malgré les efforts d'homogénéisation des données, à la suite des étapes CAO/FAO, un post-processeur marque une nouvelle coupure afin de transcrire les trajectoires en code G.

1.2.1.2. Code G : un format non adapté à la description de trajectoires complexes

Le code G est un langage développé initialement par l'*Electronic Industries Alliance* (EIA) au milieu des années 1960 et a été normalisé depuis 1980 (ISO 6983). Ce langage est basé sur une description de bas niveau des actions à effectuer par la machine. Concernant les trajectoires, une succession de points de passage est définie dans le repère de coordonnées

de la machine. Initialement, seules des interpolations linéaires et circulaires sont possibles. En plus des problèmes de rupture déjà évoqués au sein de la chaîne numérique (dû à la nécessité d'un post-processeur), la description des trajectoires élémentaire entraîne des problèmes d'approximations lors de la programmation de trajectoires courbes. En effet, dans le cas de surfaces complexes, une discrétisation est déjà opérée entre le modèle CAO et les trajectoires FAO, entraînant alors une facettisation. Ensuite, pour une meilleure précision, la discrétisation doit être fine, et le nombre de segments à exécuter au sein de la CN devient conséquent, pouvant entraîner alors des problèmes de vibrations (Figure 6).

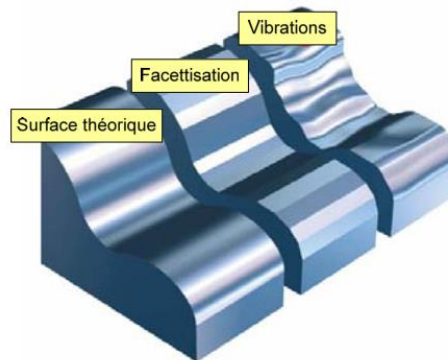


Figure 6 - Défauts liés à l'interpolation linéaire pour l'usinage de surfaces complexes [Siemens04]

Afin d'éviter ces défauts, certaines CN ont évolué et autorisent la programmation de trajectoires complexes à l'aide de formats de description plus poussés, tels que l'interpolation polynomiale, les splines, ou encore les NURBS (*Non-Uniform Rational Basis Splines* [Piegl95]). Ces formats sont basés sur des modèles mathématiques permettant d'avoir une courbe de consigne d'accélération continue [Langeron2004] et donc au final d'avoir une surface complexe conforme, avec un meilleur aspect et un temps d'exécution plus court. Les avantages sont nombreux mais ces formats d'exécution ne sont pas encore disponibles sur toutes les CN et une maîtrise de compétences est nécessaire [Duc02][Affouard04][Langeron04].

Notons aussi que grâce à l'utilisation de calculateurs plus performants, la plupart des CN récentes proposent des fonctions de compressions regroupant les blocs de programmes d'interpolations linéaires en un bloc d'une trajectoire redéfinie en splines [Siemens04]. Ceci améliore grandement le comportement cinématique de la machine [Pateloup05] car la trajectoire peut être exécutée plus rapidement avec moins d'à-coups et de ralentissements d'axes. Le choix est même proposé de privilégier le temps d'exécution, l'état de surface, ou la fluidité de déplacement des axes.

Finalement, les DCN actuels présentent une large palette de fonctions d'optimisation du comportement cinématique des MOCN. Les commandes numériques et le format de description se sont adaptés à l'évolution des technologies (performances cinématiques des machines) et la morphologie des pièces à obtenir. Mais ces évolutions restent encore trop limitées par le formalisme du langage code G. De plus, les fonctionnalités avancées développées sont bien souvent destinées à une CN propriétaire.

Nous constatons donc que le code G est un langage de plus en plus désuet et reste un maillon faible de la chaîne numérique. Ces lacunes ont conduit au développement d'un nouveau standard d'échange de données, appelé STEP-NC.

1.2.1.3. Le standard STEP-NC pour étendre les possibilités d'échange de données

L'étude de la chaîne numérique classique a permis de mettre en avant les principales lacunes du code G [Laguionie10]. Afin de pallier celles-ci et d'étendre les possibilités d'échange des données, le standard STEP-NC reprend les bases du standard STEP pour les adapter à la production sur MOCN [Suh03]. Une des principales avancées de ce langage est de permettre une programmation basée entités : le programme décrit « quoi faire » plutôt que « comment faire » [Kramer06]. Cela permet alors une meilleure prise en compte des possibilités d'un couple process/machine, et peut donner un rôle plus important à la CN. Ceci permet d'avoir un flux bidirectionnel des données au sein de la chaîne numérique et une meilleure capitalisation de l'expérience [Suh02]. L'impact de ce langage sur les échanges de données au sein de la chaîne numérique est synthétisé sur la Figure 7.

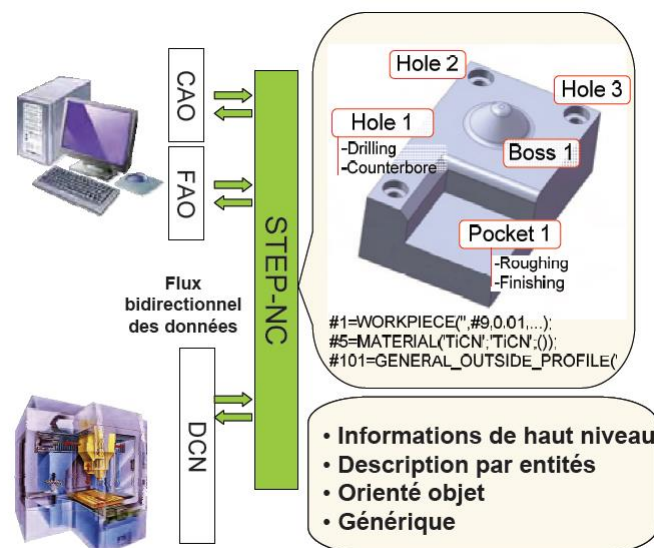


Figure 7 - Chaîne numérique basé sur le standard STEP-NC [Laguionie10]

Les pertes d'information sont minimisées car le STEP-NC offre un modèle complet de données liées à la production. Nous voyons sur cette figure que le post-processeur est supprimé et entraîne maintenant des possibilités de feedback entre la machine et les suites de conception.

Concernant les problématiques d'échange d'informations, le format STEP-NC offre de grandes possibilités d'évolutions. Nous verrons par la suite que ce format modifie aussi l'approche de programmation actuelle dont nous allons maintenant présenter les principales évolutions.

1.2.2. Evolutions de la méthode de programmation : vers une programmation avancée du process de fabrication

L'évolution des maillons de la chaîne numérique et des échanges d'informations est dépendante de l'évolution de la programmation du parcours d'usinage. Les méthodes de programmation avancée tendent à prendre en compte le comportement réel du processus de fabrication, que cela soit avant (méthodes prédictives), pendant (méthodes correctives) ou, dans un cas moins favorable, après (méthodes curatives) la réalisation de la pièce. Nous allons présenter les principales avancées de ces méthodes de programmation et les éventuelles limitations bloquant leurs intégrations au sein des industries manufacturières.

1.2.2.1. Méthodes de programmation prédictives

Nous avons vu que la programmation de la chaîne numérique classique se base exclusivement sur des contraintes géométriques et cinématiques mais ne prend pas en compte les données du process, la structure de la machine, ainsi que les défauts liés à l'ensemble outil/pièce/montage. Nous savons par exemple que le parcours le plus court n'est pas nécessairement le plus productif, du fait de la structure de la machine et des ralentissements des axes aux discontinuités géométriques [Pateloup2005]. Afin d'obtenir les meilleurs résultats de production, il est nécessaire de compléter les compétences et l'expérience du programmeur à travers l'utilisation de différents outils de prédiction.

Les méthodes prédictives sont basées sur l'utilisation d'un ensemble d'outils de simulation du comportement réel de la MOCN dans le but d'optimiser la trajectoire et les conditions de coupe définies en FAO [Terrier05] [Tapie07]. Une vue de ce modèle de programmation est présentée sur la Figure 8.

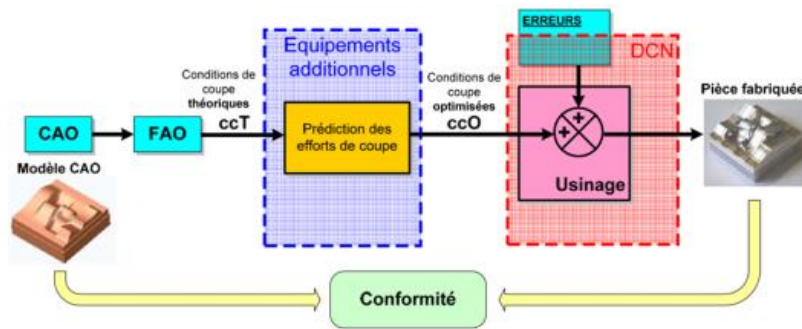


Figure 8 - Modèle de programmation avancée de MOCN par méthode prédictive [Rauch07]

Nous pouvons citer des travaux de prédiction des efforts de coupe lors de l'usinage pour compenser la flexion d'outil [Dessein97][Seo98][Li2004] ou encore des travaux liés à l'optimisation de vitesse d'avance en fonction du comportement réel de la MOCN [Hascoet00] [Hascoet06] [Dugas03] (Figure 9).

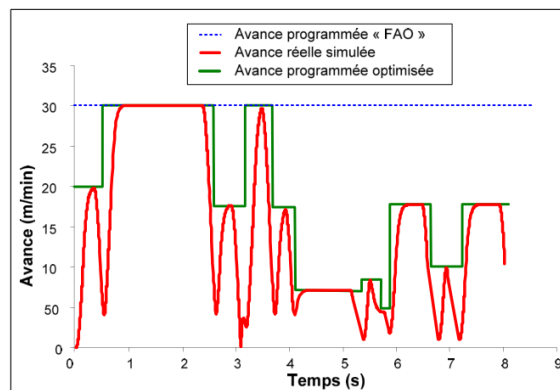


Figure 9 - Optimisation de l'avance programmée après simulation de l'avance réelle [Rauch07]

Ces méthodes apportent un gain de productivité et de conformité de la pièce, mais elles sont encore trop sujettes à des erreurs de modélisation. En effet, les phénomènes mis en jeu durant l'usinage et le comportement dynamique de la machine sont parfois imprévisibles avec les moyens actuels. De plus, le caractère fermé des CN empêche d'avoir toutes les informations nécessaires aux simulations. Il en résulte alors des approximations qui nécessitent des vérifications et ajustements au pied de la machine. Ces lacunes peuvent être compensées par l'utilisation de méthodes dites correctives.

1.2.2.2. Méthodes de programmation correctives

Contrairement aux méthodes précédentes, qui effectue une optimisation de la trajectoire hors-ligne et avant l'exécution du programme, les méthodes de programmation correctives permettent une prise en compte de perturbations durant l'usinage de la pièce et l'optimisation en ligne. Ce modèle peut être représenté de la manière suivante (Figure 10).

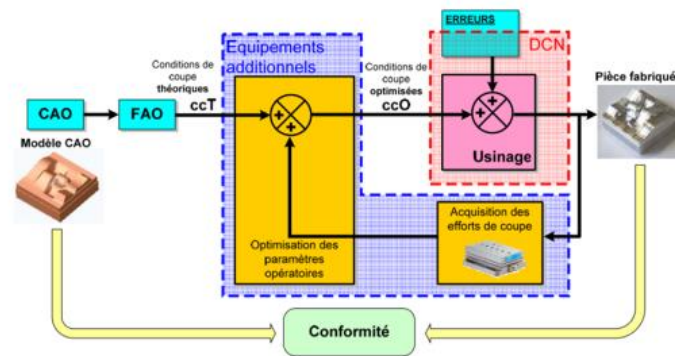


Figure 10 - Modèle de programmation avancée de MOCN par méthode corrective [Rauch07]

Les travaux dans ce domaine étant plus rares, nous allons rentrer dans le détail de ces méthodes afin d'identifier divers manques.

1.2.2.3. Intérêt des méthodes de programmation correctives

Les méthodes de programmation correctives se basent sur le concept de contrôle adaptatif déjà utilisé dans de nombreux domaines dès les années 1960. En effet, celui-ci est d'abord appliqué sur des systèmes à boucle fermée simple (de type échangeur de chaleur, refroidisseur) puis sur des applications plus complexes (autopilotage d'avions et de missiles dès 1977).

Dans le domaine de la fabrication, afin d'expliquer l'intérêt du contrôle adaptatif, et donc des méthodes de programmation correctives, une analogie est faite entre le problème d'optimisation des paramètres du process (trajectoires, paramètres opératoires, ...) et l'action d'un opérateur au pied de la machine [Simoes14] :

- L'opérateur mobilise ses sens pour observer une opération de fabrication.
- A partir de son ressenti, il interprète le bon déroulement de l'opération.
- Si besoin, il modifie les paramètres de la machine pour augmenter ses performances.

Les paramètres qui gèrent le procédé de fabrication sont donc fixés au préalable de manière prudente, ou à la suite d'une programmation prédictive, puis optimisés en fonction de l'expérience de l'opérateur, ce qui donne des gains de productivité (déjà de 5 à 10% au début des années 1980 [Koren83]).

L'intérêt du contrôle adaptatif est donc de compléter et d'automatiser les actions de l'opérateur expérimenté, et l'analogie donne donc simplement :

- Utilisation d'un capteur pour le ressenti (Identification)
- Un modèle pour le bon déroulement de l'opération (Décision)

- Calcul des nouveaux paramètres et intégration (Modification).

L'ensemble de ces étapes est représenté au sein du processus de fabrication sur la Figure 11.

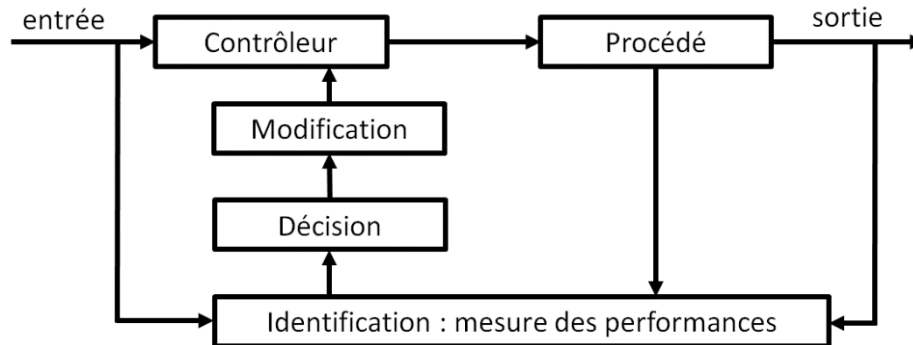


Figure 11 - Contrôle adaptatif d'un procédé [Simoes14]

L'état de l'art permet d'identifier et de synthétiser l'ensemble des informations observées jusqu'à maintenant lors d'un procédé de fabrication [Koren83] [Liu89] [Watanabe83] [Hudson82] [Yang02] :

- Les déformations élastiques de la pièce et de l'outil
- Les déformations dues aux effets thermiques
- Les vibrations
- L'usure d'outil
- Les perturbations périodiques (jeu, éléments rotatifs non équilibrés)
- Les efforts (pression en perçage, couple en tournage/fraisage)

Par le biais des informations relevées et par la connaissance du comportement du procédé de fabrication, de plus en plus pointue au fil des années, des modèles de description de la dynamique du process sont enrichis et intégrés dans des boucles de contrôle.

L'un des problèmes des méthodes de programmation correctives est qu'elles nécessitent bien souvent des équipements de mesure additionnels qui sont intrusifs et difficiles à mettre en place au sein de la MOCN. Une des solutions est alors d'utiliser les données déjà accessibles au sein de la CN. Dans ce cadre, nous pouvons citer la méthode dite ICAM (Intelligent Computer Aided Manufacturing) qui vise à dépasser les limites associées aux méthodes d'optimisation exposées plus haut en étant générique et simple d'emploi, tout en restant performante [Hascoet06][Rauch07] (Figure 12).

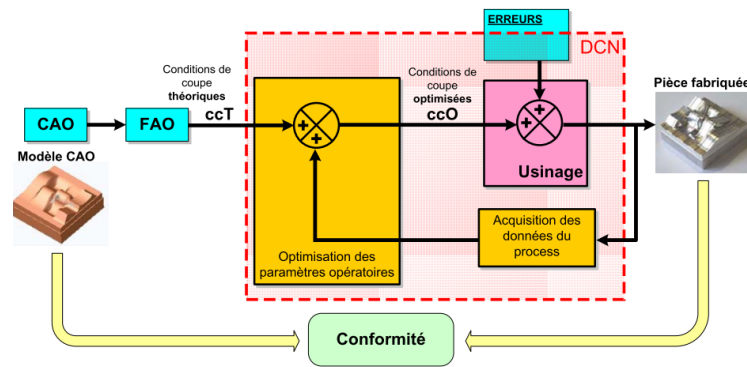


Figure 12 - Principe de la méthode de programmation ICAM [Rauch07]

Cette méthode permet une estimation des efforts de coupe durant le process par une lecture des données de couples et de géométrie de la MOCN données par la CN. Une optimisation est alors effectuée sur les paramètres opératoires de l'usinage. Cette méthode est d'ailleurs aussi étendue à d'autres paramètres et d'autres procédés, tel que le formage incrémental [Rauch09].

Bien souvent dans le domaine de la production, les paramètres modifiés pour optimiser la production se concentrent sur les consignes/gains des moteurs d'axes, ainsi que sur les paramètres liés au procédé (par exemple vitesse d'avance dans le cas de l'usinage). Les différents travaux d'adaptation durant un process ont permis d'avoir un gain de productivité et de conformité [Liu89]. Un champ reste cependant encore peu exploré concernant les modifications de plus haut niveau, en particulier liées à la trajectoire générée par la FAO et à la géométrie réelle de la pièce à fabriquer. Alors que le caractère imprévisible de position et de déformation de la pièce est très important, nous allons voir qu'il est nécessaire d'opérer des compensations géométriques en ligne.

1.2.2.4. Nécessité de compensations géométriques

Il existe un certain nombre d'écarts introduits durant la chaîne de fabrication qui compliquent l'adéquation des trajectoires d'usinage définies à l'aide de maquette numérique sur la pièce réelle au sein de la MOCN. Tout d'abord, en plus des écarts dus aux échanges de données depuis la CAO, il existe des écarts commis durant les opérations précédant l'usinage étudié qui modifie donc la géométrie initiale de la pièce [Tichadou04]. La position initiale de la pièce peut aussi varier selon la qualité du porte-pièce, la configuration de bridage et les efforts de bridage [Wang05]. La trajectoire et les paramètres opératoires doivent donc être adaptés pour obtenir une pièce conforme. Cette optimisation peut cependant être faite hors-ligne, mais cela au prix d'une immobilisation de la machine entraînant alors une perte de productivité.

De plus, il existe aussi des écarts dit de réalisation [Prévost11] qui sont dus, d'une part, aux défauts d'assemblage, de guidage et de vibrations de la structure de la machine, et, d'une autre part, à la déformation de l'outil ou de la pièce dues aux sollicitations de l'usinage ou aux contraintes résiduelles de la pièce usinée [Chen12].

Les méthodes de compensations en ligne sont encore rares pour ces défauts. Nous pouvons citer les travaux de Zhao [Zhao08] qui s'intéresse au contrôle en ligne (évitant l'utilisation de machine à mesure tridimensionnelle) et qui modifie la trajectoire à l'aide des avantages du format STEP-NC (évitant de remonter à la conception), mais seulement après l'usinage d'une première pièce. C'est en quelque sorte une méthode corrective dite curative (après occurrence d'un défaut).

Nous constatons alors qu'un gain de productivité est possible en permettant la compensation des défauts en temps réel. Pour cela, il faut mettre en œuvre des méthodes correctives plus performantes que l'on identifiera sous l'appellation de méthode d'usinage adaptatif temps réel.

1.2.2.5. Principe de l'usinage adaptatif temps réel

L'usinage adaptatif consiste à modifier la trajectoire d'usinage d'une pièce afin qu'elle s'adapte à la position exacte et à la morphologie de cette dernière. Pour cela, il est donc nécessaire d'avoir une stratégie de mesure intégrée à la machine [Simpson11]. Jusqu'à maintenant, l'utilisation d'un palpeur monté sur la tête d'usinage est le moyen le plus répandu, mais cela nécessite des vitesses d'approches faibles [Achelker14]. Bien que les moyens de digitalisations sont nombreux [Li04], il en existe encore peu d'adaptés réellement à l'usinage adaptatif.

Les travaux les plus récents, réunissant des aspects de rapidité et de réactivité sont ceux de Poulhaon [Poulhaon15]. Celui-ci utilise un laser disposé sur une partie fixe de la broche outil pour mesurer l'altitude de la pièce au voisinage de la position outil. Les éventuelles erreurs d'altitudes sont alors compensées en modifiant la jauge outil au sein d'une CN propriétaire SIEMENS. Ces travaux sont plus portés sur l'identification et l'obtention de la nouvelle trajectoire que sur l'intégration de cette nouvelle trajectoire au sein de la CN. La méthode d'intégration est encore complexe, dû au caractère fermé de la CN (les informations temps réel sont difficilement accessibles) mais aussi dû encore une fois au format de description du code-G (la description bas niveau n'est pas propice aux modifications).

1.2.2.6. Approche de programmation basée sur STEP-NC

Il est donc encore technologiquement difficile de réaliser un usinage adaptatif à l'aide du standard code G. Il existe donc de nombreuses perspectives de contrôle intelligent en utilisant le langage STEP-NC qui sont illustrées à l'aide d'une plateforme d'implémentation développée au sein de l'IRCCYN et intitulée SPAIM (STEP-NC Platform for Advanced and Intelligent Manufacturing) (Figure 13).

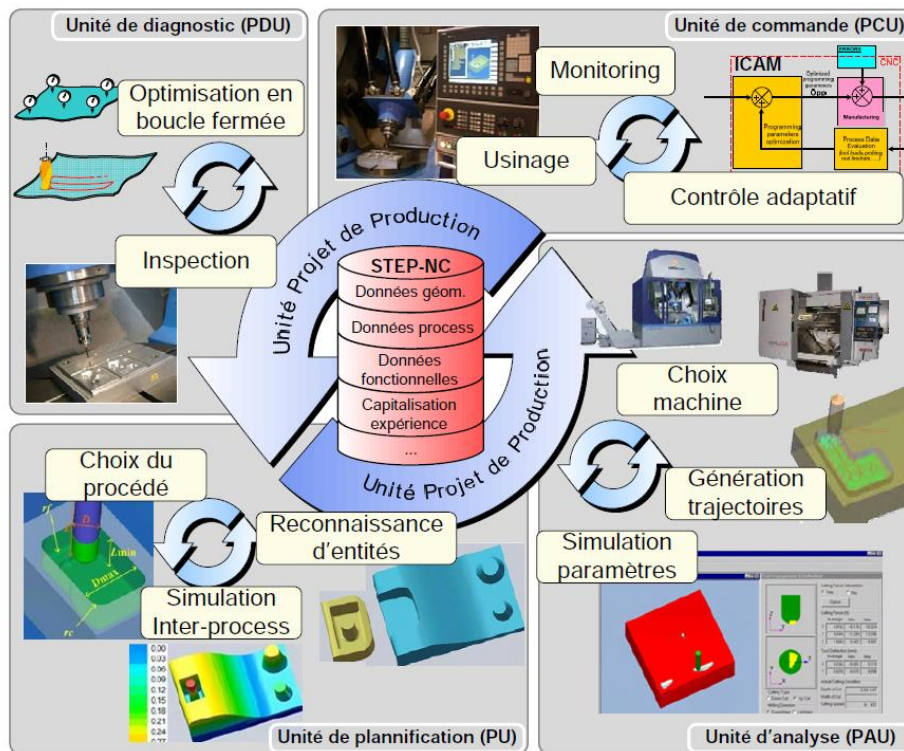


Figure 13 - Vue d'ensemble de l'approche SPAIM pour une programmation avancée et intelligente [Laguionie10]

Cette plateforme fait interagir différentes unités (présentées plus en détails dans le chapitre 4 de ce manuscrit) qui communiquent entre elles par l'intermédiaire du langage STEP-NC. Ces interactions permettent de réaliser l'ensemble des évolutions citées précédemment, mais nécessitent une intégration complète du langage STEP-NC au sein de la CN. Ce qui reste encore difficile sur une CN propriétaire.

A travers les différentes premières sections de ce chapitre, nous avons vu que beaucoup d'évolutions sont possibles au sein de la chaîne numérique, mais certaines restent difficilement exploitables dû aux limitations de la CN propriétaire. Nous allons donc faire une synthèse de ces limitations avant de décrire nos différents axes de recherches.

1.3. Synthèse des limitations des CNs propriétaires

Nous pouvons synthétiser les limitations de la CN propriétaire suivants différents axes liés aux évolutions de la chaîne numérique.

Tout d’abord, un axe de limitation est lié au format de description des trajectoires propre aux CN qui n’a pas suivi l’évolution des formats de description de pièces complexes. Pour satisfaire des besoins de production de plus en plus exigeants, les fournisseurs de CN ont dû s’adapter et ajouter des fonctionnalités supplémentaires (compression, interpolations splines, ...) au standard du code G. Ces fonctionnalités sont encore peu intégrées dans l’industrie et méritent d’être étendues à toutes les CN. De plus, une homogénéisation des données au sein de la chaîne numérique peut être opérée à l’aide du langage STEP-NC, mais celui-ci est difficilement intégrable dans les CN propriétaires actuelles.

Un second axe de limitation est lié aux difficultés de mise en place de méthodes de programmation avancée. La communauté scientifique développe des modèles prédictifs qui souffrent encore de la méconnaissance des algorithmes internes à la CN et du comportement réel de la machine durant la réalisation d’une pièce. De même pour les méthodes correctives dont le potentiel est limité par le manque de possibilité de modification du pilotage temps réel de la MOCN.

En parallèle de ces limitations, il y a aussi des limitations d’ordre économique et pratique, des commandes numériques propriétaires [Xu06]. Tout d’abord, un grand nombre de produits sont incompatibles avec les CN propriétaires (fournisseurs spécifiques et spécifiés). Il existe aussi des difficultés à s’adapter aux fréquences de mises à jour liées aux besoins. Enfin, les services, maintenances et coûts de réparations sont élevés.

Dans le cadre de nos travaux, nous allons répondre à ces limitations à travers différents axes de recherche.

1.4. Axes de recherche

Nous avons vu à travers l’étude de l’évolution de la chaîne numérique que la CN peut jouer un rôle plus important qu’aujourd’hui dans l’élaboration d’une pièce. Celle-ci ne doit pas rester une simple exécutante et doit contribuer à une meilleure flexibilité des méthodes de programmation de trajectoires.

Nous choisissons alors de répondre à la problématique suivante :

Comment générer une trajectoire en ligne qui s'adapte aux conditions réelles de réalisation d'une pièce au sein de la commande numérique ?

Afin de répondre à cette problématique, nous définissons alors différents axes de recherches qui sont étudiés dans le cadre de cette thèse :

- Axe 1 : Lever les limitations des CN propriétaires (chapitre 2)
- Axe 2 : Définir et mettre en œuvre une méthode de programmation avancée corrective et plus particulièrement dans un contexte d'usinage adaptatif (chapitre 3)
- Axe 3 : Etudier l'adéquation de la méthode de génération de trajectoires en ligne au sein de l'usine du futur (chapitre 4)

Le premier axe permet de définir une solution alternative à l'utilisation des CN propriétaires. L'objectif est d'obtenir une plateforme entièrement fonctionnelle, pouvant exploiter toutes les évolutions décrites dans ce chapitre, sans aucune limitation.

Le deuxième axe est focalisé sur les possibilités de génération de trajectoires en ligne grâce à la prise en compte d'informations de perturbation au cours de la réalisation d'une pièce. Les développements effectués se font dans le cas de l'usinage d'une pièce subissant des perturbations simples (linéaires suivant deux axes), mais se veulent généralisables à tout type de perturbations et procédés de fabrication.

Le troisième axe met en avant le champ d'application qui découle de l'observation en temps réel du processus de réalisation. Les informations observées peuvent être directement exploitées pour adapter la trajectoire mais peuvent aussi être capitalisées afin d'améliorer la flexibilité de la production. Chaque axe inclut les développements des précédents axes comme illustré sur la Figure 14.

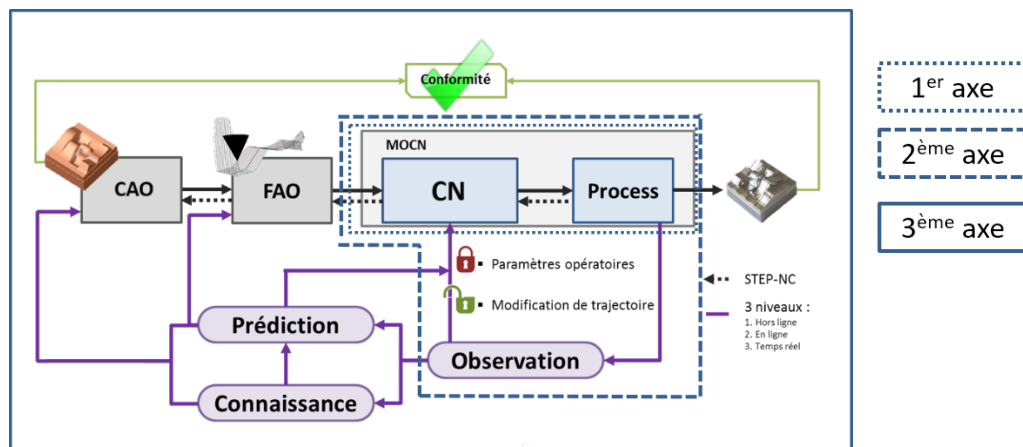


Figure 14 - Axes de recherche au sein de la chaîne numérique

1.5. Conclusion

Ce premier chapitre a permis de présenter les différentes évolutions des méthodes de programmation des MOCN qui permettent de répondre aux problématiques des modèles courants de la chaîne numérique de fabrication.

Un état de l'art des méthodes de programmation prédictives et correctives montre qu'il existe un manque de prise en compte de certaines informations de perturbation au cours de la réalisation d'une pièce. Et quand ces informations sont accessibles en temps réels, il est encore difficile de les exploiter au sein de la commande numérique.

Un besoin de compenser les perturbations en temps réel est alors identifié et nous amène à définir le principe de fabrication adaptative. Nous avons vu que cette approche est encore limitée par des obstacles au sein de la commande numérique. A la suite d'une synthèse de ces limitations, nous avons défini les différents axes de recherche qui sont étudiés dans le cadre de nos travaux et qui répondent à la problématique de génération de trajectoires en ligne s'adaptant aux conditions réelles de réalisation d'une pièce au sein de la commande numérique.

Chapitre 2

Possibilités et intérêts des CNO

Chapitre 2 – Possibilités et intérêts de la CNO

Le premier chapitre de ce manuscrit a permis de montrer que l'évolution de la chaîne numérique de fabrication est freinée par les limitations des CN propriétaires. Il est donc nécessaire d'évoluer vers une CN offrant des possibilités de programmation avancée adaptées à l'industrie d'aujourd'hui et de demain.

Dans ce deuxième chapitre, nous allons présenter en premier lieu nos volontés concernant cette CN du futur (section 2.1.1). Ces exigences nous amènent à considérer l'utilisation de CN Ouvertes (CNO). Après une présentation des critères propres à la CNO (section 2.2.1), nous faisons une synthèse des différents projets existants (section 2.2.2). Les projets sont nombreux mais les mises en œuvre de CNO sur une machine industrielle sont encore rares.

Nous nous focaliserons sur le choix, la description (section 2.3.1) et l'intégration d'une CNO (section 2.3.2) sur une MOCN industrielle. L'entière fonctionnalité de la plateforme résultante, le respect des critères d'une CNO et les possibilités d'algorithmes avancés sont présentés par la suite (sections 2.3.3), ces points faisant partie de nos exigences données en première partie.

2.1. Motivations et objectifs d'une nouvelle CN

2.1.1. Constats et objectifs

Nos travaux se placent dans un contexte mondial et visent à définir et mettre en œuvre une CN offrant des possibilités de programmation avancée adaptées à l'industrie de demain. Le marché des CN est dominé par des grands noms de l'industrie tels que Siemens, Fanuc ou encore Heidenhain.

Notre objectif n'est pas d'essayer de définir et de mettre en œuvre une CN qui dépasse toutes les fonctionnalités existantes, en ne partant d'aucune base. Notre objectif n'est pas non plus de présenter seulement des concepts, car les industries manufacturières doivent être convaincues de nos propositions et cela n'est possible qu'avec une mise en œuvre dans un contexte industriel.

Une de nos volontés est aussi de proposer des solutions génériques, utilisables par tous et à moindre coût. Ceci est difficilement envisageable si nos travaux sont propres à une CN propriétaire en particulier.

De plus, les fournisseurs de CN restent ancrés sur l'utilisation du langage code-G. Plus qu'un défi technologique, il est apparu nécessaire de convaincre les industriels d'évoluer en

démontrant la plus-value accessible d'un langage de haut niveau tel que le STEP-NC [Suh02].

Ces constats nous amènent à partir d'une base de machine industrielle et d'y intégrer nos développements qui respectent les objectifs suivants :

1. Intégrer une CN supplémentaire de nouvelle génération à la machine industrielle utilisée, afin de ne pas limiter les fonctionnalités de la machine à nos seuls développements. Ceci tout en conservant la CN d'origine de la machine (pour une meilleure intégration dans le milieu industriel). Ceci permet aussi une comparaison du comportement des CN sur une même structure mécanique.
2. Intégrer une CNO, car c'est une CN qui respecte des critères (de portabilité, d'interopérabilité, d'extensibilité et d'évolutivité) nécessaires à nos développements et développés par la suite.
3. Montrer l'entière fonctionnalité de cette CNO et illustrer un exemple de fonction avancée
4. Mettre en œuvre la fonctionnalité d'action corrective en ligne
5. Evaluer les fonctionnalités supplémentaires apportées par le langage STEP-NC
6. Evaluer la place de ces développements sur l'évolution de la chaîne numérique

Par la suite, la plateforme en lien avec ces développements sera désignée par l'acronyme XPAIM (eXtended Platform for Advanced and Intelligent Manufacturing).

XPAIM est par ailleurs exploitée dans le cadre du projet européen FoFdration, présenté ci-après, pour lequel a été introduit le concept de SMC (Smart Machine Controller) [Rauch14].

2.1.2. XPAIM au sein du projet européen FoFdration – Concept de Smart Machine Controller

XPAIM est une réalisation concrète correspondant aux objectifs du projet européen FP7 FoFdration (Foundation for the Sustainable Factory Of the Future [FoFdration])[Rauch14]) réunissant 16 partenaires académiques de 8 pays différents (Figure 15).



Figure 15 - Partenaires du projet européen FoFdration

Ce projet a pour but de développer un « Contrôleur de Machine Intelligent (SMC, Smart Machine Controller) qui, en lien avec les besoins industriels, inclut les caractéristiques suivantes :

- Le système apprend par expérience et ses performances sont améliorées par l'utilisation de bases de connaissances et de simulations
- Les conditions de fabrication sont automatiquement sélectionnées pour produire des pièces avec la qualité désirée et le maximum d'efficacité
- Les différents modèles définis, ainsi que les capteurs pouvant intégrer la machine, fonctionnent en synergie afin d'améliorer le process et la précision des modèles
- Un langage de haut niveau est utilisé pour communiquer les besoins, contrôler le process, décrire les composants physiques et stocker un historique

Une vision d'ensemble du concept de SMC est synthétisée sur la Figure 16 [Rauch14].

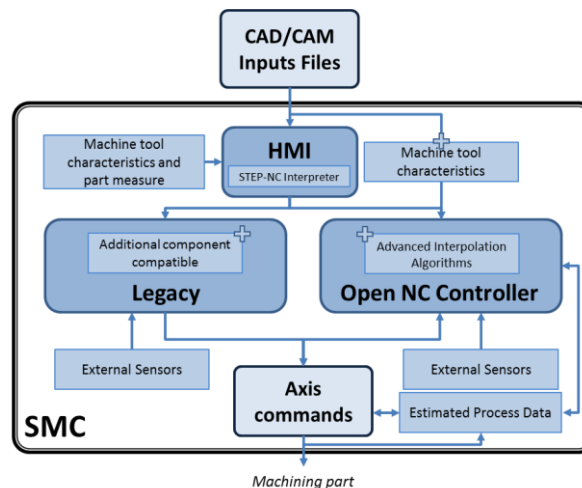


Figure 16 – Vision globale du concept de SMC complétée de [Rauch2014]

Ce concept est mis en œuvre dans le cadre de nos travaux. Les bases et les différents objectifs étant définis, nous allons justifier le choix de la CNO intégrée à la XPAIM. Pour cela, il est nécessaire de définir ce qu'est une CNO et de synthétiser les différents efforts académiques et industriels de développement de systèmes ouverts. Cette synthèse nous permettra d'identifier la CNO adéquate à nos développements sur la XPAIM.

2.2. Commande numérique ouverte : définition et projets mondiaux

Du fait des limitations des CN propriétaires, une relative ouverture des CN est devenue un argument commercial pour certains constructeurs (Siemens, Okuma, Fidia et autres CN avec environnement Windows). Par exemple, l'IHM ou encore l'automate programmable peuvent être modifiés, ou encore les coefficients d'un modèle d'asservissement peuvent

être optimisés (et ceci, bien souvent dans le cadre de l'installation ou d'une maintenance opérée par le constructeur de la CN). Ce type de CN peut être défini comme « semi-ouverte » et ne permet pas d'accéder aux parties temps réel du contrôle de déplacement. Face aux différentes attributions du mot « ouvert » qui peuvent être faites par différents acteurs, il est nécessaire de présenter notre définition de la commande numérique ouverte.

2.2.1. Définition d'une commande numérique ouverte

La définition de la commande numérique ouverte est basée tout d'abord sur celle des systèmes informatiques ouverts. Le comité technique de l'IEEE (Institute of Electrical and Electronics Engineers) à travers la Norme [IEEE95] définit ce qu'est un système ouvert. De cette norme, nous en ressortons notre définition d'un système ouvert. Tout d'abord, un système ouvert permet à des applications correctement mises en œuvre d'être intégrées et de fonctionner sur différentes plateformes de différents fournisseurs. Celles-ci doivent également interagir avec d'autres applications du système ainsi que fournir une interface utilisateur cohérente.

Une des caractéristiques principales des systèmes ouverts est une grande modularité à la fois logicielle et matérielle. Cette modularité peut être illustrée par un projet du domaine de la technologie grand public, initialement nommé Phoneblocks et devenu le Projet Ara de Google (Figure 17).



Figure 17 - Illustration PhoneBlocks (2013) et photos d'un smartphone [Ara15]

Dans ce projet, chaque fonctionnalité d'un smartphone (téléphonie, internet, photographie...) est décomposée dans différents composants modulaires (réseau, écran, appareil photo, processeur, batterie). Ceci permet de personnaliser l'ensemble en fonction du budget de l'utilisateur et du besoin (ex. : en ajoutant une batterie de plus grande capacité pour les grands nomades). Ceci facilite aussi les mises à jour et montées en puissance du système (ex : en changeant de processeur), et bien évidemment a un avantage en terme d'écologie et de durabilité : plutôt que de remplacer tout l'ensemble lorsqu'un élément est défectueux, il suffit de remplacer seulement un module. De plus, un

même module peut être modifié plutôt que remplacé pour avoir des fonctionnalités avancées et s'adapter aux besoins de l'utilisateur final.

Par cette illustration, nous comprenons bien l'intérêt des systèmes ouverts, qui peut être appliqué à tout domaine.

Pour revenir à la CN et aux inconvénients des structures propriétaires, la comparaison peut être illustrée de la manière suivante (Figure 18).

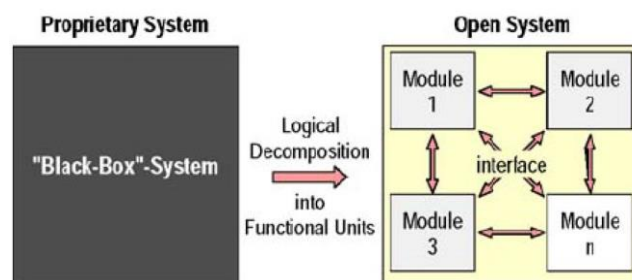


Figure 18 - Décomposition des fonctionnalités d'une CN [Pritschow et al. 2001]

L'architecture fermée est assimilée à une boîte noire, où seules les données d'entrées et de sorties sont connues, tandis que le système ouvert fait apparaître un ensemble de modules plus petits, où chaque données d'échanges est visible. Plus la taille des boites est petite, plus les applications sont nombreuses et plus le système est modulaire [Koren 1988].

De plus, afin que le développement de la CN ouverte soit possible, quelques conditions de mise en œuvre sont nécessaires pour les différents composants. Les fournisseurs doivent être neutres, pour respecter les critères d'extensibilité et d'évolutivité, tout en empêchant tout bénéfice à un fournisseur particulier. Tout développement ne devient pas propriétaire. La communauté de développement doit être active. L'évolution du système doit être contrôlée par les utilisateurs et les fournisseurs. Les besoins immédiats et futurs doivent être identifiés et développés. Il faut respecter une certaine standardisation. Tous les éléments d'échange du système doivent être standardisés par la communauté internationale afin de garantir un vaste réseau de partage et d'intérêt. Enfin l'ensemble doit être librement et ouvertement disponible. Ce principe est équivalent à la désignation d'*open source* (code source ouvert).

Afin de compléter la définition d'un système ouvert, dans le cas d'une commande numérique ouverte, les quatre critères suivants sont définis [Miles 1998] [Rauch 2014] :

- **Portabilité** (*Portability*) :

Capacité d'un module (*Application Module*, AM par la suite) à être transféré et entièrement fonctionnel sur différentes plateformes sans nécessiter de modification.

- **Extensibilité** (*Extendability*) :

Possibilité d'ajouter des AMs au système sans que leurs capacités soient diminuées ou qu'il y ait conflit entre eux.

- **Interopérabilité** (*Interoperability*) :

Possibilité d'interaction entre les différents AMs dans un même système. Ils doivent être capables de travailler ensemble et d'échanger des informations sur un standard normalisé.

- **Evolutivité** (*Scaleability*) :

Possibilité de modifier et adapter la fonctionnalité et la performance des AMs en fonction de la taille du ou des systèmes. Ces critères définissant un système ouvert sont représentés dans la Figure 19.

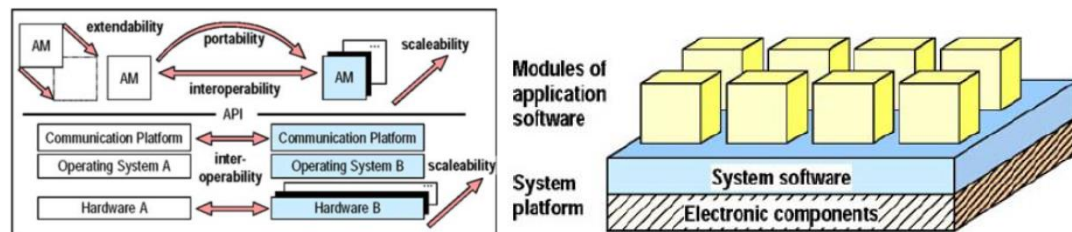


Figure 19 – Critères d'ouverture d'une CN [Pritschow et al., 2001]

Parfois, d'autres critères peuvent apparaître (Interchangeabilité [Laguionie10], Vivacité [Asato02], Transplantable [Yamasaki97]), mais ceux-ci découlent des critères principaux cités précédemment ou du concept même de système ouvert.

En somme, si et seulement si les quatre critères présentés sont bien respectés, nous pouvons attribuer le mot « ouvert » à notre système.

Les différents critères étant définis, nous allons maintenant nous focaliser sur différents projets d'architectures ouvertes, afin d'en extraire une CNO concordant avec nos objectifs.

2.2.2. Etat de l'art de différents projets d'architecture ouverte

Il existe un certain nombre de projets académiques et industriels ayant pour objectif de définir le cadre et/ou la mise en œuvre de nouvelles fonctionnalités des CN. Nous pouvons citer les premiers grands projets internationaux développés aux débuts des années 1990 tels que OSEC [Fujita96][Asato02], HOAM [Altintas94] ou encore OSACA [Nacsa08]. Dans le cadre de nos travaux, notre but n'est pas de rentrer dans les détails de chacun de ses projets, mais plutôt d'évaluer les principaux apports et surtout la possibilité d'exploitation de programmation avancée.

Dans le cadre du projet FoFdration, en exploitant et en complétant des comparaisons existantes tels que [Pritschow01] ou [Nacasa08], nous aboutissons à un tableau (Figure 20) qui illustre clairement la diversité des différentes CNO et aussi la difficulté de faire ressortir la CNO idéale, tant les architectures, méthodes de communications et modules sont variés.

	Types of Open Architectures							
	OSEC	JOP	HOAM-CNC	OSACA	OMAC	FSA	EMC2	FSMC
Operating Platform	Windows only	Windows	Windows and others	Windows and others	Windows and others	Windows and others	Windows/ RT Linux	Windows/ RT Linux
Type of Architectural Framework	Component-based	Wrapper	Client-Server	Client-Server	Component-based	Client-Server/ component-based	Hierarchical, Component based	Client-Server/ component-based
Distributed	No	No, parallel	No, parallel	Yes	Yes	Yes	Yes	Yes
Infrastructure Defined	No	No	No	Yes	No	Yes	Yes	Yes
Level of Modularity	Low	Medium	High	Medium	Medium	Very High	High	High
AM Architecture	Modular	Modular	Modular	Modular building blocks	Modular building blocks	Modular	Modular, blackbox	Modular, blackbox
Named Modules	7	4	6	9	14	7	5, general	11 (16)
Module Description	Undefined	Undefined	Partially defined	Undefined	Defined with FSM	Defined	Defined	Partially defined
Type of API	Wrapper	Generalized Wrapper	Wrapper	Function calls	Object-oriented	Object-oriented	Object-oriented, wrapper	Object-oriented
Advantages (API)	Easy to understand	Vendor neutrality	Level of modularity	Level of modularity	Proxy agents for implementation	High modularity and interconnectivity	Effective HAL	High modularity and interconnectivity
Disadvantages (API)	Not object oriented, too simple	Not fully developed	Partially defined	Variables instead of functions	Complicated API	latent high systems demands	Blackbox model for AMs	Undefined Modules
Programming Language	C/C++	C/C++	C/C++	C/C++	C, C++, Java	C/C++	C/C++	C/C++/Delphi
Communication Platform	Specific DLL's	Undefined/ anything	Undefined/ anything	Hierarchical, OSACA specific	Undefined/ anything	OPC Specification	NML	NML, DLL, OPC Spec.
Current Interest	Very low	Very low	Medium	Low	High	High	High	Very High

Figure 20 - Comparaison de différentes architectures de CNO [FoFdration11]

Dans l'ensemble, ces projets sont très orientés sur la description de l'architecture, du fonctionnement modulaire et des échanges d'informations. Dans le cadre de nos travaux, nous attachons beaucoup d'importance à ce que ces descriptions soient les plus détaillées possible ainsi que les applications soient concrètes et pas seulement conceptuelles. Nous identifions alors un critère dit d'intégration qui traduit la possibilité de mise en œuvre, l'intérêt actuel du projet et la fonctionnalité des différents modules. Sur la Figure 21, nous illustrons une répartition des orientations retenues appliquée à trois projets parmi les plus complets (dans un souci de lisibilité).

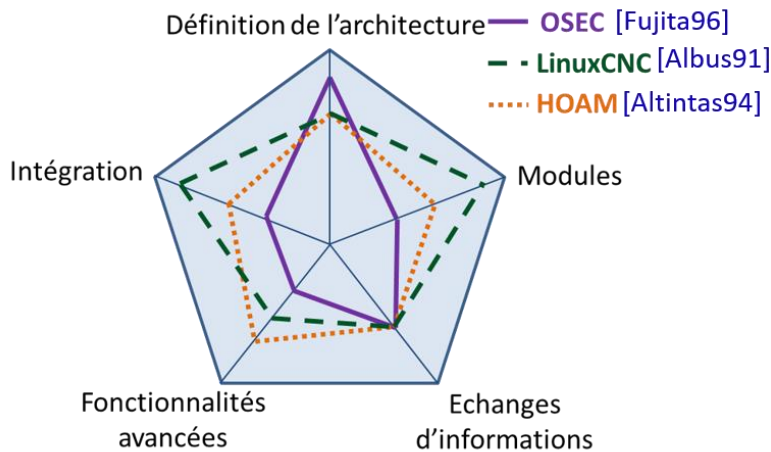


Figure 21 - Illustrations simplifiées des différents critères de choix de CNO

Sur cette figure, nous voyons que l'architecture OSEC est difficilement exploitable, du fait surtout du manque de description des modules (Figure 20). Dans le cas du projet HOAM, une importance a été donnée à l'intégration de fonctionnalités avancées (monitoring, détections de collisions), mais ceci dans un cadre encore trop conceptuel et difficilement intégrable. Enfin, LinuxCNC (anciennement intitulé EMC2) [LinuxCNC], du fait de sa communauté très active, de son intérêt actuel et de son intégration sur de nombreuses machines (de type prototype, développés en interne), montre un fort attrait à l'intégration.

Concernant cet aspect d'intégration, nous pouvons aussi citer le projet PREMIUM-OpenCNC (LURPA), pour lequel l'accent est mis principalement sur les aspects de description de trajectoires et la maîtrise des lois de mouvements. Cette plateforme offre la possibilité d'ajouter de nouveaux algorithmes d'interpolation ou d'asservissement [Beudaert13], mais les problématiques d'échanges d'informations et de descriptions modulaires sont volontairement mises de côté. Bien qu'intégré à une machine industrielle et offrant des gains de productivité et d'aspect de surface (lié à la possibilité d'interpoler directement sur la surface à usiner), le système qui en résulte est difficilement exploitable par l'industrie en l'état. En effet, il n'offre pas toutes les fonctionnalités d'une commande

numérique classique (telles que l'interprétation de programme), ne respecte pas l'ensemble des critères d'un système ouvert tels que l'interopérabilité et n'est donc pas considéré comme une CNO.

Les projets de CNO aboutis, avec des applications réelles dans un contexte d'usine manufacturière sont donc assez rares. En nous focalisant sur des critères d'intégration et de description complète des modules de fonctionnement, qui sont primordiaux pour respecter nos objectifs, notre choix s'oriente vers la CNO LinuxCNC. Les avantages de celle-ci sont présentés de manière plus approfondie ci-après.

2.3. LinuxCNC : définitions et intégration au sein de la XPAIM

Dans cette partie, l'intégration d'une CNO au sein de XPAIM est présentée. Tout d'abord les spécificités du logiciel utilisé LinuxCNC sont illustrées avant de démontrer les performances et l'entière fonctionnalité de la plateforme aux regards des critères définis précédemment.

2.3.1. LinuxCNC

Afin de maîtriser l'intégration de LinuxCNC et exploiter les possibilités de programmation avancée, il est important de bien comprendre son architecture générale.

2.3.1.1. Historique et architecture générale de LinuxCNC

LinuxCNC est basée sur le contrôleur EMC (Enhanced Machine Controller) qui est né à l'initiative du NIST (National Institute of Standards and Technology) [Albus91][Proctor93] dont l'objectif était entre autre d'étudier et de développer une commande numérique complète de machine. La version actuelle de LinuxCNC fonctionne dans l'environnement du système d'exploitation Linux muni d'extensions temps réel. La stabilité et les performances du système d'exploitation sont les principales raisons du choix de cet environnement par la communauté de développement. Le fait que cet environnement soit distribué en open source respecte aussi les prérequis d'un système ouvert (2.2.1).

L'avancée principale de LinuxCNC est l'utilisation d'un Hardware Abstraction Layer (HAL, littéralement, couche d'abstraction matérielle) qui est une couche logicielle centrale. Cette couche permet à des simples modules matériels d'être assemblés et interconnectés pour obtenir par la suite un système complexe et performant. L'atout principal du HAL est sa capacité à traiter des modules qui peuvent être échangés, modifiés ou remplacés, sans nécessairement changer le système complet : le lien avec les critères d'interopérabilité et

de portabilité est alors évident. La possibilité est aussi donnée de tester partiellement ou globalement le système avant son implantation physique réelle. Le HAL permet le transfert de données en temps réel de la CN aux différents matériels de commande et modules logiciels sous-jacents. Il sert donc de cadre au développement et à l'exécution en temps réels des différents modules [HAL Manual]. Sa simplicité d'utilisation sera démontrée dans le chapitre suivant qui met en œuvre une modification du système existant dans le cadre de nos travaux.

L'architecture générale de ce contrôleur fait apparaître quatre composants: un contrôleur de mouvement (EMCMOT), un contrôleur des entrées/sorties analogiques (EMCIO), un exécuteur de tâche réunissant un coordinateur (EMCTASK) et une interface utilisateur (GUI's).

Ces composants ont des fonctions assez proches des composants présentés en section (CN propriétaire Chapitre 1), respectivement, l'unité de contrôle (sans interpréteur), l'automate programmable et l'interface homme/machine (incluant l'interpréteur).

2.3.1.2. Les principaux composants de LinuxCNC

a. Contrôleur de mouvement (EMCMOT)

Ce composant fonctionne de manière cyclique. Il gère les actions de contrôle telles que la planification de trajectoires, la transformation directe et inverse, et le calcul de la sortie désirée des systèmes moteurs. Ce composant fait lien entre l'exécuteur de tâche (interpréteur) et l'interface HAL, qui est reliée aux parties matérielles (cartes d'axes) (Figure 22).

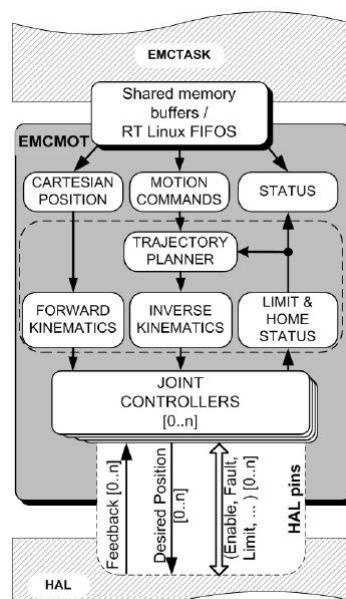


Figure 22 – Structure du composant EMCMOT [Staraveski2009]

b. Contrôleur des entrées sorties analogiques (EMCIO)

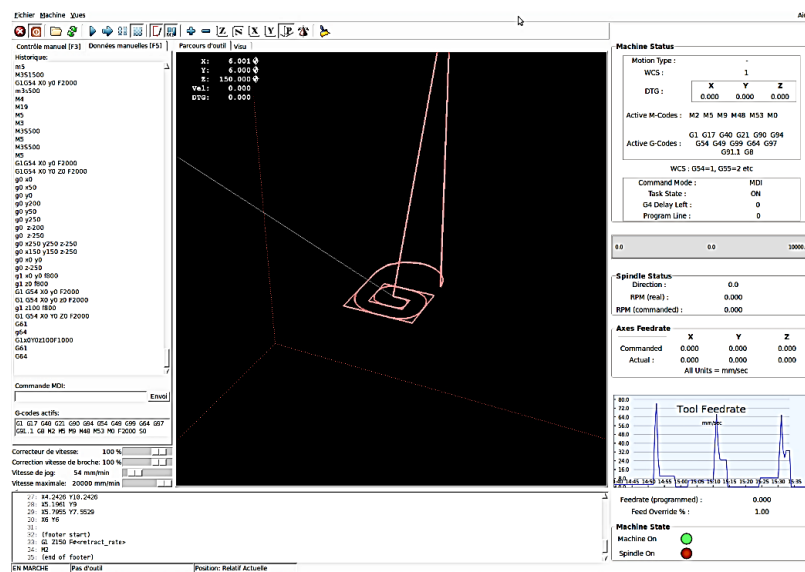
Ce composant, contrairement au précédent, ne fonctionne pas dans un environnement temps réel. Il gère toutes les fonctions non liées aux contrôles des mouvements des axes. Il reprend exactement toutes les fonctions d'un automate programmable. Sa particularité est qu'il peut être composé de plusieurs modules de gestion d'entrées/sorties analogiques, facilitant les modifications de l'ensemble. En étant directement lié à l'interface HAL, il est entièrement adaptable et portable.

c. Exécuteur de tâche (EMCTASK)

Ce module a le rôle de gestionnaire des tâches à effectuer et d'interpréteur du langage de programmation de machine-outil (Code-G) [Marietta1994]. Ce module est hiérarchiquement supérieur aux deux composants précédents car il est en charge de surveiller les activités et statuts de chacun d'entre eux. Ce module reçoit et analyse les commandes venant de l'interface graphique et les traduit en langage NML (Neutrale Messaging Language) [NML][Shackleford00].

d. Interfaces graphique utilisateur (GUI's)

Cette interface donne accès à l'utilisateur à tous les paramètres et options en interaction avec le système. LinuxCNC permet d'avoir une IHM classique liée à la machine (Figure 23) ou une interface déportée. La particularité de ces interfaces est qu'elles sont personnalisables et peuvent permettre l'accès à des réglages d'éléments liés au fonctionnement temps réel. Des fonctions de surveillance et d'acquisition (type oscilloscope) sont aussi possibles.



L'ensemble de ces quatre composants constitue alors l'architecture générale de LinuxCNC visible sur la Figure 24.

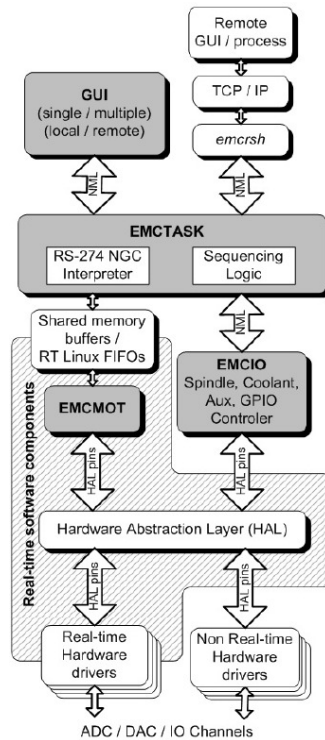


Figure 24 - Architecture du système LinuxCNC [Staroveski2009]

Nous remarquons bien sur cette figure le lien établi par la couche HAL entre les composants EMCOT, EMCIO et les composants matériels. Cette couche facilite grandement l'intégration.

La structure et les composants de LinuxCNC étant maintenant présentés, nous allons désormais détailler les développements effectués pour son intégration au sein de la plateforme XPAIM du laboratoire.

2.3.2. Cohabitation de CN et intégration d'une CNO au sein de la XPAIM

Comme expliqué dans la section 2.1.1, un des objectifs, durant les différentes intégrations sur la base de la XPAIM, est de conserver toutes les fonctionnalités du contrôleur originel NUM, tout en permettant l'intégration de différentes CN invitées telles que LinuxCNC, Fidia et éventuellement d'autres CN de fournisseurs lambda, dans une parfaite cohabitation sur une même machine industrielle Figure 25. Ceci permet aussi de montrer l'influence de chaque contrôleur durant le procédé et de démontrer les potentiels technologiques de chacun (entre autres, lissage de courbe, utilisation de NURBS).

Les différentes implémentations et innovations doivent être utilisables à court terme par l'industrie manufacturière. Cette plateforme doit encourager l'implication des principaux fournisseurs du secteur des machines-outils. La MOCN utilisée comme base pour l'ensemble de ces objectifs est une fraiseuse trois axes Cincinnati Milacron Sabre 750 [Cincinnati] équipée d'une commande numérique NUM 750F.

Pour faciliter le passage d'une CN à une autre au sein de la XPAIM, un bouton matériel a été ajouté et permet de donner le contrôle complet de la machine sur le contrôleur sélectionné (Figure 25 en bas à droite).

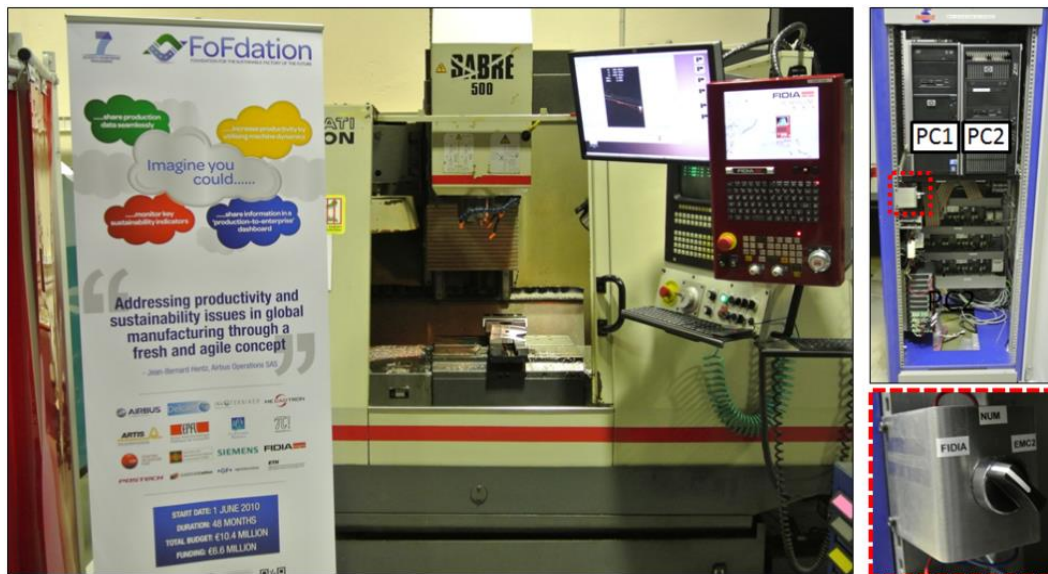
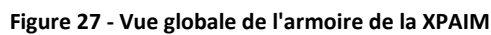


Figure 25 - Vue Globale de la cohabitation des différentes CN

L'implémentation de LinuxCNC passe par l'utilisation de deux ordinateurs de support. Un premier ordinateur (PC1) est équipé du système d'exploitation Linux (Ubuntu), du logiciel LinuxCNC et quelques matériels supplémentaires de contrôle (cartes d'axes, interface d'entrée sortie analogique et lecteurs de codeurs). Cet ordinateur est en charge du contrôle temps réel de la machine et est basé sur un cœur Linux temps réel (RT-Linux kernel). Tandis que le deuxième ordinateur (PC2) tournant sur un système d'exploitation Windows contient tous les logiciels tels que CAO et FAO. C'est sur cette plateforme que l'on retrouve SPAIM [Rauch09] qui permet le contrôle basé STEP-NC. Cet ensemble est visible dans la Figure 26.



Avant de décrire quelques fonctionnalités propres à cette plateforme, nous pouvons refaire un lien vers les différents critères définissant un système ouvert dans le cas particulier de l'utilisation de LinuxCNC.

- **Portabilité** : le système d'exploitation Linux en lui-même, qui est un système ouvert, assure la portabilité.
- **Interopérabilité** : concrètement cela est possible par l'intermédiaire du Profibus et du HAL de LinuxCNC.
- **Extensibilité** : concrètement cela correspond au rajout de SPAIM, de Linux CNC et de différentes fonctionnalités (Nurbs, ...).
- **Evolutivité** : LinuxCNC peut être utilisée sur des machines avec une structure cinématique plus complexe (5 axes) mais aussi destinées à des applications différentes (formage incrémental, fabrication additive, ...).

Afin de satisfaire nos objectifs, plus particulièrement les trois premiers présentés en section 2.1.1 (correspondant à l'intégration et la démonstration de fonctionnalités des CNs ajoutées), nous allons maintenant analyser les performances de la XPAIM.

2.3.3. Analyse des performances de la XPAIM et fonctionnalités avancées

Nous détaillerons dans cette section l'analyse des performances de la XPAIM, tout d'abord dans le cas d'un fonctionnement « classique » (exécution d'un programme) à l'aide de LinuxCNC puis à l'aide de Fidia, qui est la CN propriétaire de nouvelle génération que nous avons choisi d'ajouter.

2.3.3.1. Utilisation « classique » de LinuxCNC

La plateforme étant équipée de plusieurs CN, elle présente l'intérêt de pouvoir comparer les différentes performances de chacune. Le but de cette partie n'est pas de démontrer qu'une CN est plus performante qu'une autre, mais plutôt de montrer que LinuxCNC est une CN entièrement fonctionnelle dans un contexte industriel.

a. Essais d'acquisition sous LinuxCNC

L'IHM de LinuxCNC propose l'acquisition de données par un outil type oscilloscope, mais celui-ci est surtout utile pour des captures très courtes (de l'ordre du centième de secondes). Cet outil sera utilisé et présenté ultérieurement dans le chapitre 3. Dans le cas d'acquisitions plus longues, un composant logiciel échantillonneur (Sampler) HAL est utilisé. La configuration HAL de la machine permet d'activer ce composant, le lier à une fonction temps réel et de définir les signaux à acquérir (avec une fréquence

d'échantillonnage d'environ 910Hz, soit une acquisition toute les 1,1ms en fonctionnement réel).

Dans le cadre de ces premiers essais, une série d'acquisitions a été réalisée avec différents paramètres (vitesses d'avances, trajectoires (Figure 28), mode arrêt précis (temps d'arrêt de 5ms) ou non (G61/G64)).

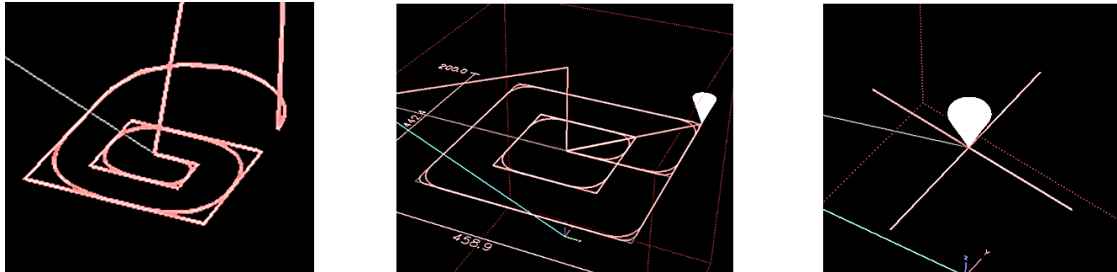


Figure 28 - Capture d'écran de différentes trajectoires sous LinuxCNC (Lock, Square and Line)

Tout d'abord, précisons que LinuxCNC offre la possibilité de pouvoir simuler l'exécution d'une trajectoire en prenant en compte la configuration cinématique de la machine. De plus les limites cinématiques de la machine (Vitesses et Accélérations) peuvent être modifiées aisément au sein de LinuxCNC. Celles-ci ont une influence « logicielle » sur le profil d'interpolation. Enfin, les coefficients d'asservissement (entres autres, facteur P, I et D) peuvent aussi être modifiés. Ces possibilités sont propres aux CN de nouvelle génération.

Les différentes acquisitions (correspondants aux signaux de déplacements de l'outil) nous permettent d'affirmer que LinuxCNC est bien performante. Les positions sont bien respectées (Figure 29). Pas d'écart supérieur à 0,05mm (correspondant à une tolérance de finition classique en fraisage).

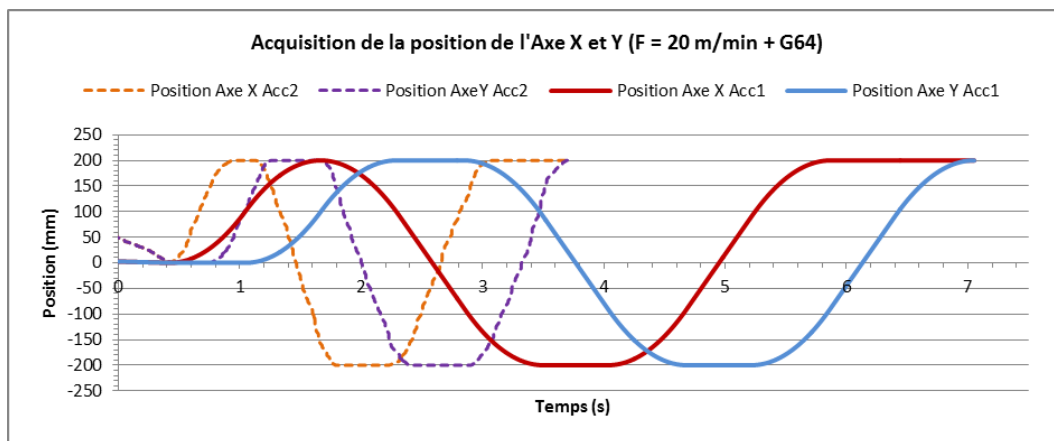


Figure 29 - Positions des axes X et Y (exécution d'un carré de 200mmx200mm)

Les limites cinématiques de la machine sont bien respectées et nous pouvons observer l'influence de la modification des accélérations logicielles de la XPAIM (Figure 30 et Figure 31).

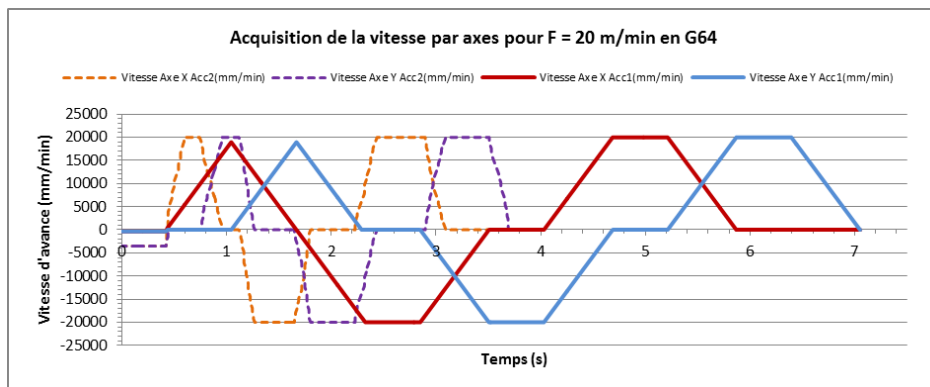


Figure 30 - Vitesses des axes X et Y (exécution d'un carré de 200mmx200mm)

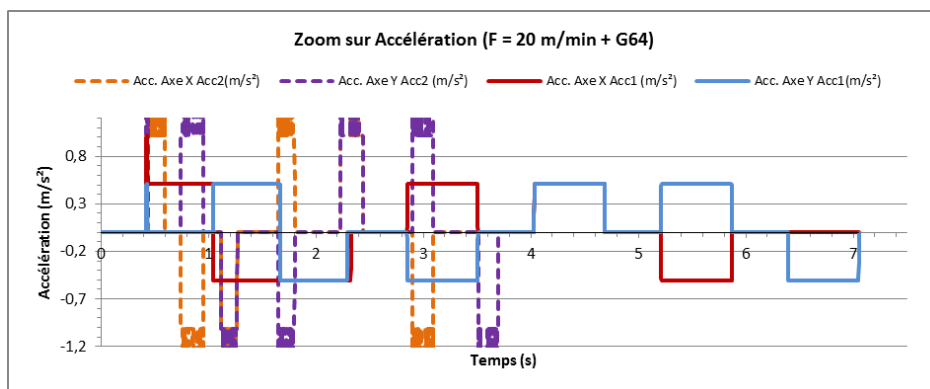


Figure 31 - Accélération des axes X et Y (exécution d'un carré de 200mmx200mm)

Nous observons sur les figures 30 et 31 que le temps de trajet du parcours avec une accélération doublée est plus court (env. 3,7s) que celui avec une accélération simple (env. 7s). Enfin, nous pouvons statuer sur la précision de l'outil de simulation de LinuxCNC, car le temps de simulation estimé est très proche du temps réel (5%) (Figure 32).

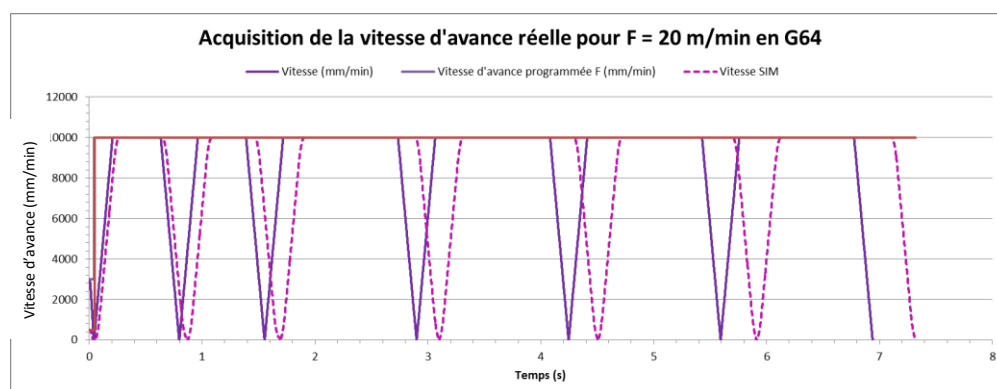


Figure 32 - Comparaison réel/simulé : Vitesse d'avance (exécution d'un zig-zag sur X en arrêt précis)

Cette légère différence peut s'expliquer par la différence d'asservissement et de pilotage des axes. Dans le cas de la simulation, les profils d'accélérations suivent une loi de trapèze (Figure 33), contrairement à la machine qui suit une loi de trapèze en vitesse.

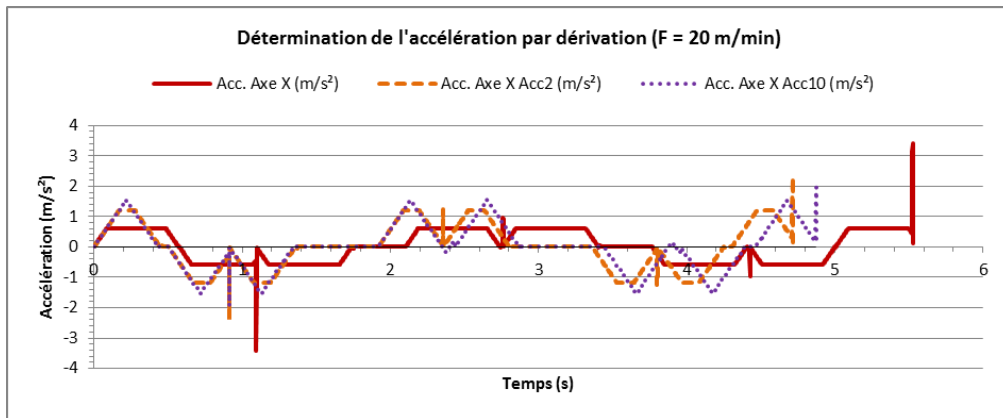


Figure 33 – Comportement cinématique en simulation : profil d'accélération

Ces premiers essais sur XPAIM montrent donc l'entière fonctionnalité de LinuxCNC. Afin d'avoir une référence, nous allons maintenant les comparer à celle d'une CN de nouvelle génération Fidia, installée sur la même machine.

b. Essais d'acquisition sous Fidia

Le choix a été fait de comparer les performances cinématiques de la machine sous FIDIA, plus précisément le modèle CNPC143 très récent, sur lequel il est possible d'utiliser un outil type Oscilloscope (Figure 34).

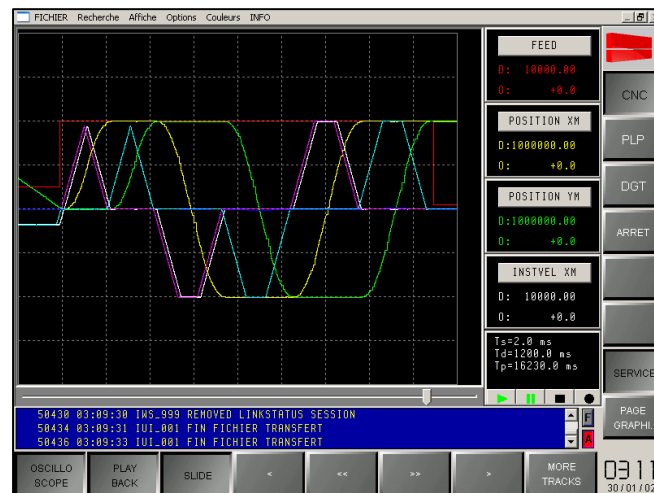


Figure 34 - Capture d'écran de l'outil oscilloscope de Fidia

Il est possible alors d'enregistrer les signaux voulus avec une fréquence d'échantillonnage de 500Hz. Lors de l'utilisation de cette CN, notre premier constat est qu'il n'existe pas de fonction G61/G64 sous FIDIA (le lissage est toujours actif). Pour comparer à LinuxCNC, nous

avons donc ajouté des temporisations de 4,4ms correspondants à un pilotage des trajectoires avec arrêts précis.

Suite aux acquisitions, nous observons que la position présente quelques légères perturbations (Figure 35), éventuellement dues à l'asservissement, mais dans l'ensemble les résultats sont très satisfaisants.

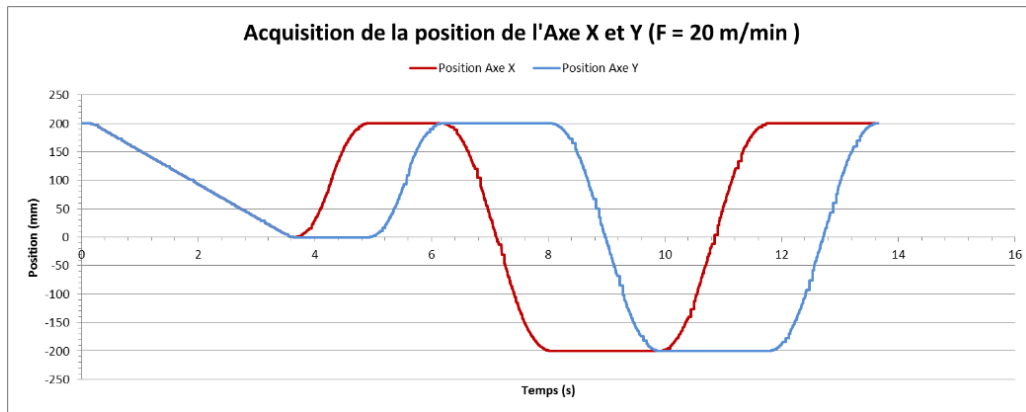


Figure 35 - Positions acquises sous FIDIA en X et Y (carrée de 200mmx200mm)

Il est alors possible de comparer ces acquisitions à celle de LinuxCNC.

c. Comparaison des deux commandes numériques

Nous déduisons des acquisitions quelques résultats comparatifs sur l'exécution et l'utilisation des deux CN pour les trajectoires testées (Figure 36) :

- Temps d'exécution sur LinuxCNC 20% plus rapide que sur FIDIA
- Fonction d'arrêt précis/lissage (G61/64) possible sur LinuxCNC, non existant sur FIDIA

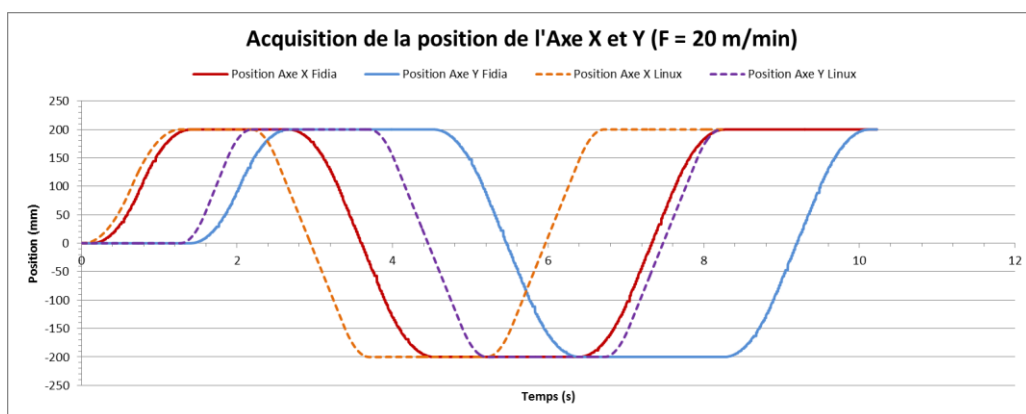


Figure 36 - Exemple d'acquisition pour un programme identique (carré 200mmx200mm) sous LinuxCNC et Fidia.

Les différences de performances sont observables par acquisitions directes des données de positions et vitesses, mais aussi par essais à l'aide d'un système Ballbar télescopique (RENISHAW). L'analyse des résultats montre les différences suivantes (Figure 37) :

- LinuxCNC est plus performante au niveau des jeux à l'inversion (-27% d'erreurs), des pics d'inversion (-82%), de l'équerrage (-21%), de la rectitude sur Y (-14%) et de la circularité (-20%)
- FIDIA est plus performante au niveau des jeux latéraux (-18%), des erreurs cycliques (-23%), de la rectitude sur X (-31%) et de la différence d'échelle (-15%)

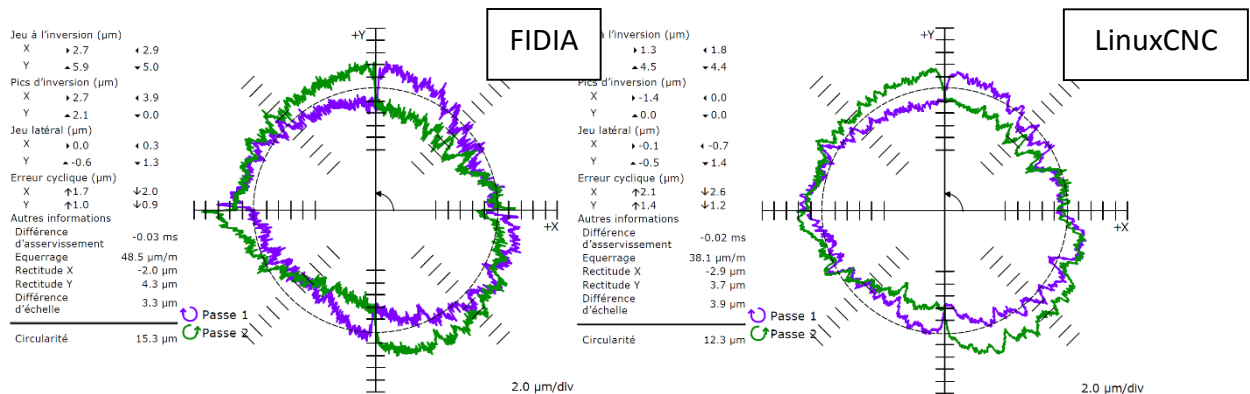


Figure 37 - Résultats d'essais Ballbar, pour F=1000mm/min, Rayon du cercle 10mm, sous FIDIA (à gauche) et LinuxCNC (à droite)

La CN LinuxCNC est donc au moins aussi performante qu'une CN propriétaire et apporte toutes les fonctionnalités usuelles.

De plus, il est aussi possible de modifier directement les caractéristiques machines au sein de la CN, et la visualisation du parcours ainsi que le réglage des décalages (repère pièce) sont peut-être plus intuitif pour certains utilisateurs sur l'interface LinuxCNC.

Une vue d'ensemble de l'utilisation de LinuxCNC est montrée sur la Figure 38



Figure 38 – Ensemble de photos de la plateforme IRCCyN en fonctionnement

2.3.3.2. Intégration de fonctionnalités avancées

LinuxCNC est donc performante et par sa structure ouverte permet d'ajouter des fonctionnalités avancées.

Afin d'illustrer le critère d'extensibilité d'une CN ouverte, une interpolation NURBS, encore peu présente au sein des commandes numériques propriétaires [Langeron04] [Siemens09], a été intégrée au sein de la plateforme de l'IRCCyN. Cette intégration permet d'aller dans le sens de l'homogénéisation de l'ensemble des données de description de la chaîne numérique (liée au premier axe de limitation traitant de la description de trajectoires). Le format de description au sein de la CN est maintenant le même que le format de description du modèle CAO. Ceci est illustré dans le cas de la description de la géométrie de pièce test 4-vagues [Dugas2002] (Figure 39).

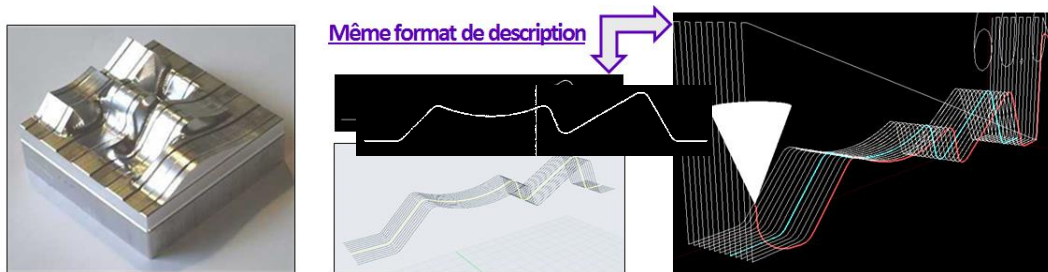


Figure 39 - Pièce 4-vagues (à gauche), CAO de la 4ème vague sous Rhino (au milieu) et trajectoires sous LinuxCNC

L'utilisation de NURBS a un intérêt concernant la précision de la description de trajectoires et la réduction de la taille des fichiers (jusqu'à 20 fois plus léger) [Wang2007]. Mais surtout un intérêt pour la plus grande continuité de la courbe de trajectoire qui en découle, et donc pour le respect de la vitesse d'avance programmée dans le cas de géométrie complexe (Figure 40). La surface usinée qui en résulte est donc moins marquée.

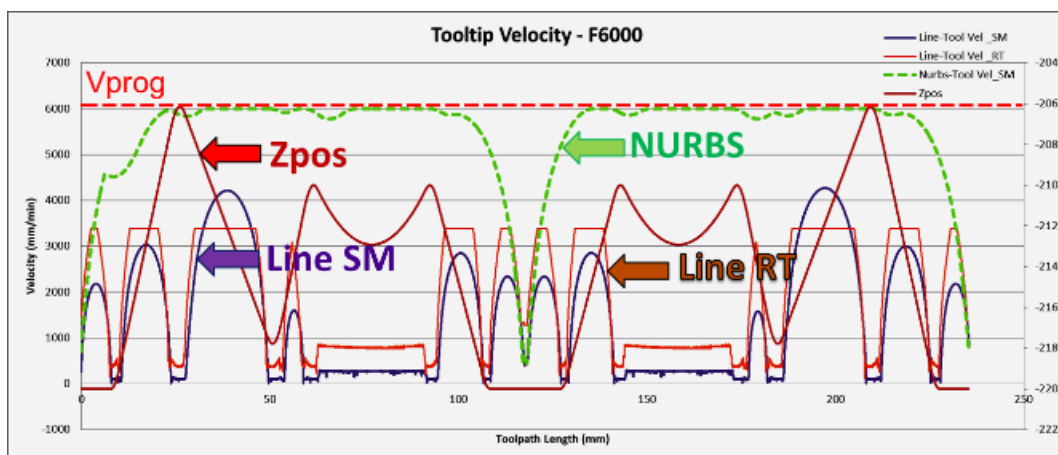


Figure 40 - Simulation sur une vague en interpolation linéaire et au format NURBS

LinuxCNC, de par sa grande communauté, bénéficie d'un large effort de développement. La version LinuxCNC de base (dite « release », version stable) est souvent mise à jour. Mais il est aussi possible d'implanter sous le système d'exploitation des versions en cours de test et développement. Dans notre cas, les développements effectués sur le format NURBS n'étaient pas encore assez aboutis. Nous avons donc fait le choix d'effectuer des tests d'une version de développement orchestré par ARAIS Technology, permettant de réaliser des courbes B-spline (Figure 41).



Figure 41 - Usinage d'une trajectoire B-spline sur la plateforme Sabre

Nous pouvons donc observer par le critère d'extensibilité propre aux CNOs, que la XPAIM peut s'adapter aux besoins de l'industrie et aussi permettre l'intégration de nouvelles fonctionnalités avancées.

2.4. Conclusion

Dans ce chapitre nous avons défini la base d'intégration XPAIM ainsi que nos objectifs de programmation avancée. Nous avons ensuite présenté le concept de commande numérique ouverte et plus particulièrement les différents critères à respecter. Puis nous avons effectué une synthèse et une comparaison des différents projets académiques et industriels. Ceci nous a permis d'identifier le projet LinuxCNC comme étant capable de servir de base CNO à nos développements, car celle-ci bénéficie d'une description complète et d'un intérêt actuel de la communauté de contributeurs. Nous avons détaillé son intégration au sein de la XPAIM et effectué un ensemble d'expérimentations démontrant ses potentiels. L'ensemble des contributions de ce chapitre répondent aux trois premiers objectifs de nos travaux rappelés ci-après :

1. Intégrer une CN supplémentaire de nouvelle génération à une MOCN, tout en conservant la CN d'origine.
2. Intégrer une CNO, car respectant des critères (de portabilité, d'interopérabilité, d'extensibilité et d'évolutivité) qui sont nécessaire à nos développements et qui seront développés par la suite.
3. Montrer l'entière fonctionnalité de cette CNO et illustrer un exemple de fonction avancée.

Dans le prochain chapitre, nous allons nous focaliser sur les possibilités d'actions correctives au sein de XPAIM, lié au cinquième objectif de notre démarche, en mettant en place une méthode de modification de trajectoires en ligne et en temps réel.

Chapitre 3

**Vers une nouvelle approche de génération de
trajectoires en ligne : Compensation temps réel au
sein de XPAIM**

Chapitre 3 – Vers une nouvelle approche de génération de trajectoires en ligne : Compensation temps réel au sein de XPAIM

Actuellement, la plupart des méthodes de programmation avancée visent à obtenir un programme de fabrication optimal et unique pour toute une production. A l'aide d'outils de simulation et de connaissances de plus en plus poussés, les prédictions du comportement réel du process permettent une augmentation de la production. Lorsqu'ils sont maîtrisés, les outils d'inspection en ligne permettent d'adapter les paramètres opératoires aux diverses perturbations possibles durant la fabrication d'une pièce. Cependant, lorsque ces perturbations affectent la position de la pièce, il est encore difficile de modifier la trajectoire avec une réactivité suffisante. Il est souvent nécessaire d'avoir recours à des actions curatives réduisant alors la productivité. Pour citer un exemple, afin de réduire les imprévus liés à la position de la pièce, dans le cas d'usinage de grand volume (ex. fuselage Airbus), la mise en position est très complexe pour éviter les déformations de la pièce. Les coûts sont très élevés pour élaborer les appuis idéaux. Afin de réduire ces coûts, une des solutions est de rendre plus flexible la mise en position et d'avoir la possibilité de compenser en temps réel les déformations de la pièce. C'est donc cette nouvelle approche de génération de trajectoires qui est présentée dans ce chapitre.

Nous avons montré dans le chapitre précédent que la plateforme XPAIM équipée de LinuxCNC respecte tous les critères d'une CN ouverte et est entièrement fonctionnelle et performante. Nous allons à présent nous focaliser sur les possibilités et la mise en œuvre d'actions correctives en temps réel.

Dans le domaine des actions correctives, trois étapes sont essentielles : l'identification (mesures de perturbations), la décision (calculs de nouveaux paramètres) et la modification (actions de compensation). Les champs d'investigation liés à ces étapes étant nombreux, nous définissons au début de ce chapitre notre frontière d'étude (3.1).

Après avoir présenté le principe général de notre nouvelle approche de génération de trajectoires en ligne (3.1.1), nous définissons notre démarche expérimentale permettant de s'affranchir des difficultés des étapes d'identification et de décision, tout en démontrant les possibilités de compenser des perturbations (3.1.2).

Ensuite, notre étude s'articulera autour du développement matériel et logiciel permettant de répondre à la question suivante : Comment prendre en compte des données extérieures à la CN afin de compenser des perturbations durant l'usinage ?

Enfin, ces développements seront validés par la mise en œuvre d'une démonstration de compensation de perturbations de positions d'une pièce au cours d'un usinage au sein de XPAIM.

3.1. Actions correctives en temps réel : Frontière d'étude et démarche de mise en œuvre sur XPAIM

Le principe d'action corrective peut être étudié sous différents angles et regroupe de nombreux axes de recherche tels que ceux concernant les méthodes de mesure ou encore le calcul de surfaces déformées. Il est donc nécessaire de délimiter une frontière d'étude pour se focaliser sur la génération de trajectoires en ligne durant le process de fabrication.

3.1.1. Principe de la génération de trajectoires en ligne et frontière d'étude au sein de XPAIM

Nous avons abordé dans le chapitre 1 l'état de l'art concernant les méthodes de programmation avancée actuelles et les difficultés d'intégration liées aux limitations des CN propriétaires. Nous pouvons alors définir une base des possibilités de programmation avancée usuelles sous forme de synthèse sur la Figure 42.

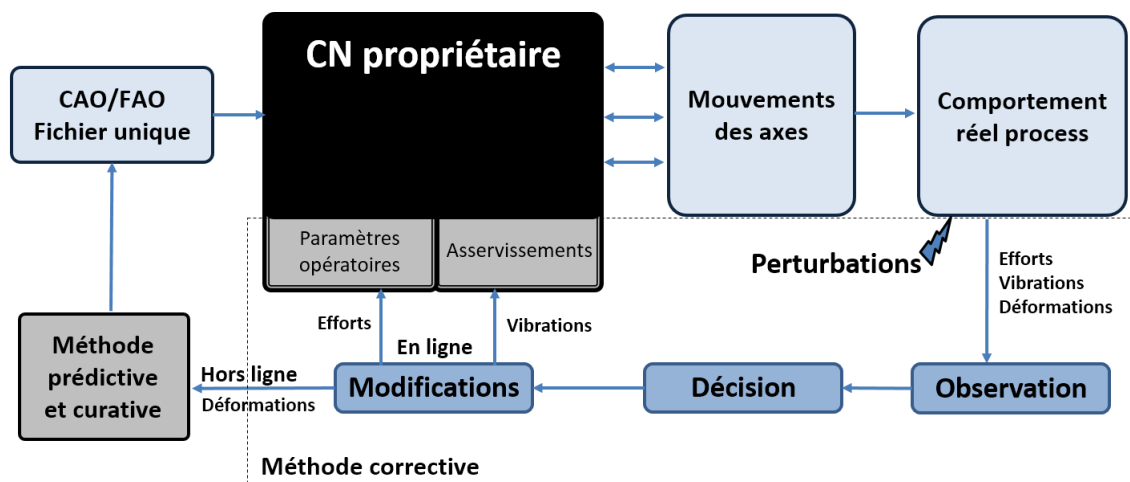


Figure 42 - Synthèse des méthodes de programmation avancée usuelles sur CN propriétaire

Nous allons suivre la description de ce schéma en partant de l'étape d'observation. Cette étape permet d'identifier, à l'aide de fonctions internes à la MOCN ou d'équipements additionnels, les différentes sources de perturbation pouvant influencer sur la conformité de la pièce. Nous considérons trois grandes sources de perturbations qui sont liées aux efforts de coupe, aux vibrations ainsi qu'aux déformations de la pièce ou de l'outil. Ces informations sont ensuite exploitées durant l'étape de décision. Cette étape permet de déterminer si une action est nécessaire pour compenser les perturbations et, dans le cas

échéant, de définir le nouveau modèle de référence de pièce à respecter. En fonction de ce modèle, des modifications sont effectuées et c'est à cette étape que les limitations de la CN propriétaire sont les plus visibles. En effet, des modifications peuvent être effectuées en ligne sur les paramètres opératoires du process (par exemple une modification de la vitesse d'avance en fonction des efforts de coupe [Rauch07]) ou sur le comportement de l'asservissement (afin de compenser les vibrations [Liu89]). Mais concernant des défauts de positionnement, les modifications en ligne sont encore rares (modification de la jauge outil par exemple [Poulhaon13]). Dans les cas plus usuels, afin de compenser ces défauts, il est nécessaire d'effectuer des modifications hors ligne de la trajectoire. Le modèle prédictif est alors mis à jour et la pièce est rectifiée par une action curative. Fort de ce constat, afin d'avoir une meilleure productivité, nous définissons une approche de génération de trajectoires en ligne au sein de la plateforme XPAIM comme illustrée sur la Figure 43.

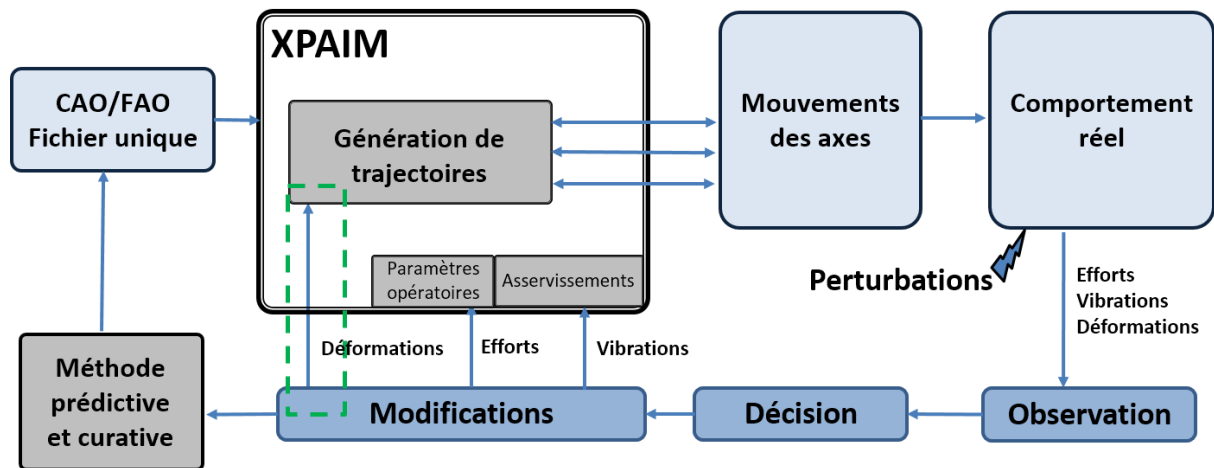


Figure 43 - Schéma de principe de l'approche de génération de trajectoires en ligne au sein de XPAIM

Cette approche est exploitable sur toute CN respectant les critères d'un système ouvert. Du fait du manque de possibilités de modification des CN propriétaires, les travaux concernant l'étape de modification sont moins nombreux que ceux traitant des étapes d'observation et de décision. Autrement dit, beaucoup de traitements sont effectués autour de la question « Comment obtenir une nouvelle trajectoire ? », mais encore peu sur la question « Comment intégrer cette trajectoire dans une MOCN ? ». C'est pour cela que nous focaliserons le plus possible nos efforts sur la résolution de cette question. La frontière de notre étude est délimitée par les éléments situés dans l'encadré (vert et pointillé) de la Figure 43.

Dans ce chapitre, nous introduirons donc une démarche expérimentale permettant d'étudier principalement la question de l'intégration de données extérieures et la modification de la trajectoire au sein de la CN.

3.1.2. Démarche expérimentale de démonstration d'actions correctives

L'objectif principal d'une démonstration d'actions correctives en ligne est d'obtenir une pièce conforme à partir d'un unique programme d'usinage, malgré des perturbations de positions. Pour cela, nous définissons des objectifs intermédiaires à respecter. Tout d'abord, il faut définir un moyen permettant d'introduire des perturbations équivalentes à des modifications globales et locales d'une pièce en cours de fabrication. Ensuite, il faut mesurer ces perturbations, puis déduire la trajectoire compensatrice pour enfin modifier la trajectoire d'usinage au sein de la CN.

Afin de simplifier le problème d'identification et de décision, pour se focaliser sur le problème de modification, nous choisissons de produire des perturbations suivant des axes existants de déplacement au sein de la MOCN, de les mesurer de manière directe et de les compenser par opposition au sein de la CNO.

Concrètement, nous souhaitons exécuter des perturbations suivant les axes X et Y de la MOCN durant l'exécution d'une trajectoire à Z constant, à l'aide d'un montage expérimental (Figure 44).

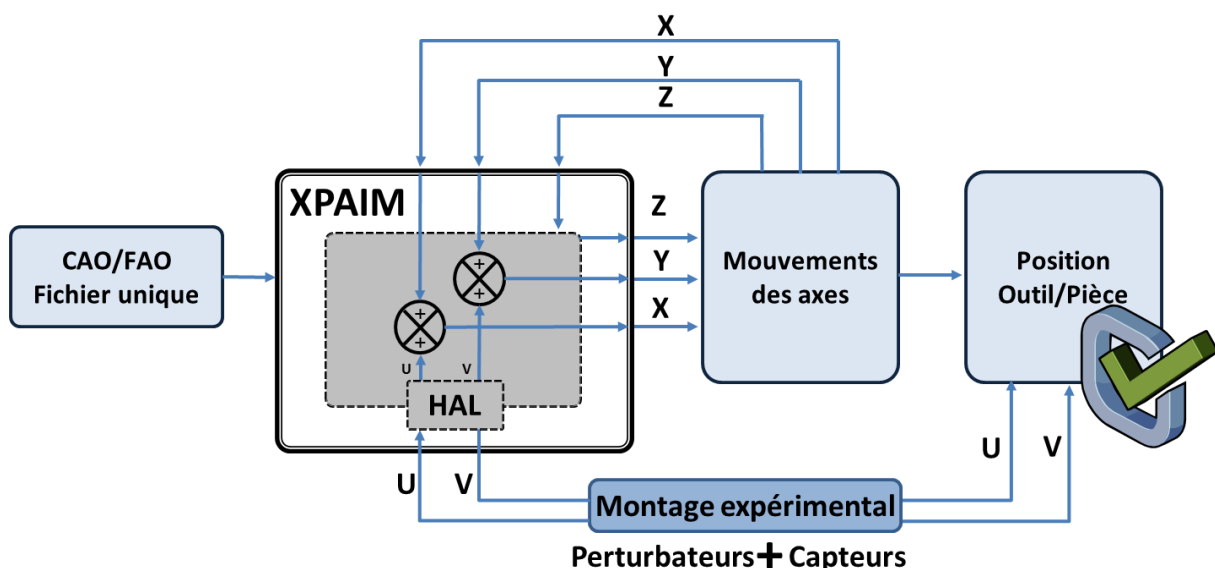


Figure 44 - Schéma de principe du montage expérimental au sein de la MOCN

Le montage expérimental joue alors le rôle à la fois de perturbateur et de capteur, mais non de compensateur, cette action étant laissée à la CNO.

Afin de mettre en place ce montage expérimental, des développements logiciels et matériels sont nécessaires, comme illustrés sur la Figure 45.

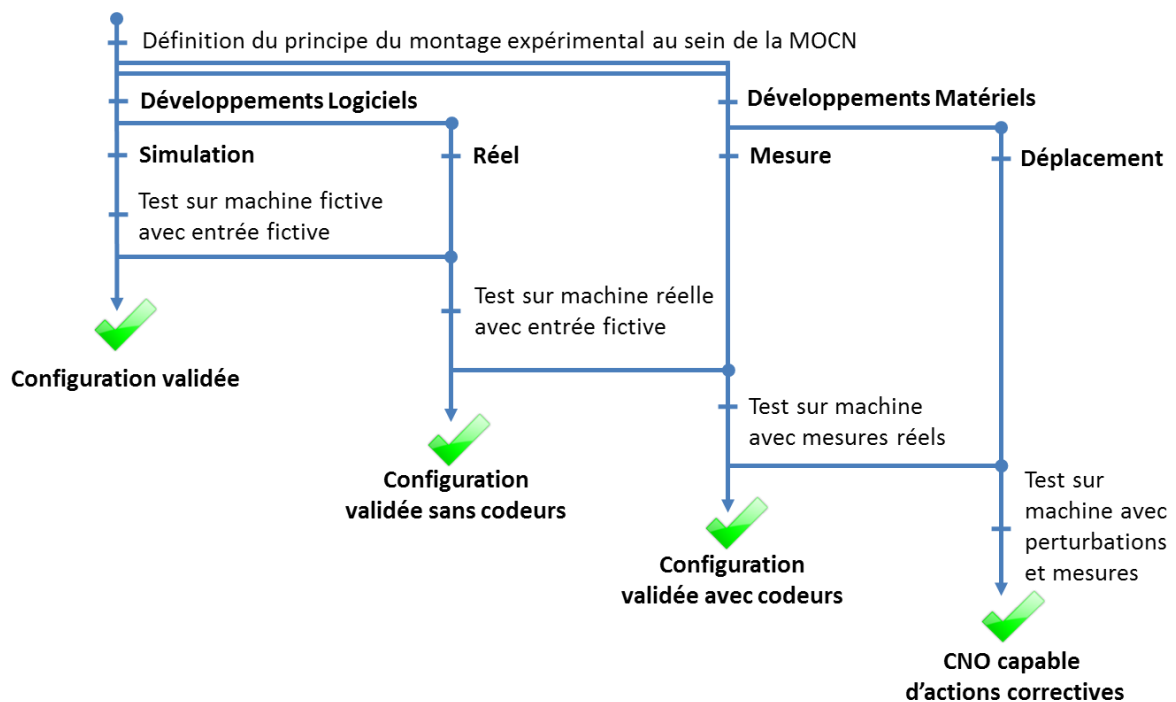


Figure 45 - Etapes de développements envisagées du montage expérimental

Bien qu'il y ait une dépendance entre les éléments matériels et logiciels, ces développements peuvent s'effectuer en parallèle. Les différents tests suivent un ordre chronologique, de la validation d'une configuration fictive avec entrées de perturbations fictives, aux essais d'usage avec perturbations et mesures réelles.

3.2. Développement du montage expérimental de démonstration d'actions correctives

Nous allons présenter dans cette partie les différents développements, tout d'abord matériels, ensuite logiciels, permettant de démontrer la possibilité d'actions correctives au sein de XPAIM. Ces développements font appel aux fonctionnalités propres à un système ouvert (définies dans le chapitre précédent).

3.2.1. Développement matériel du montage expérimental

Le montage doit permettre de perturber la position d'une pièce durant son usinage au sein de la plateforme XPAIM afin de simuler une modification durant le process de la géométrie

à usiner. Pour cela, celui-ci doit tout d'abord pouvoir mettre et maintenir en position une pièce durant un usinage. Ensuite, le montage doit autoriser des déplacements suivants l'axe X et Y de la MOCN. Enfin, il doit permettre de mesurer et d'envoyer les informations de déplacement à la plateforme XPAIM.

Afin de choisir la meilleure solution remplissant ces objectifs, nous devons répondre à un certain nombre de questions. Quelles contraintes mécaniques doit supporter le montage ? Ou encore, quelles vitesses voulons-nous atteindre ? Nous allons aussi définir par quels moyens la pièce est mis en mouvement et comment mesurer ces déplacements, afin de les exploiter au sein de LinuxCNC.

Nous traitons ces différentes questions dans les développements ci-dessous.

a. Quelles contraintes mécaniques doit supporter le montage ?

Le champ de solutions est volontairement réduit par l'application désirée. En effet, afin de limiter les efforts lors de l'usinage, tout en étant proche d'une application industrielle, nous souhaitons réaliser deux types d'usinage :

- Usinage d'une rainure (de côté 100x100 mm suivant X et Y) dans de la mousse condensée (force de coupe spécifique très faible) pour avoir une démonstration de compensation dans le cadre d'une ébauche.
- Usinage en flanc dans de l'alliage d'aluminium (pénétration radiale de l'outil 0,2mm) pour avoir une démonstration de compensation dans le cadre d'une finition.

L'usinage en ébauche, dans de la mousse condensée, nous permet d'avoir des résultats de compensation bien visibles. Tandis que l'usinage en finition dans de l'alliage d'aluminium est plus proche d'un cas réel (usinage de poche pour une pièce destinée à l'aéronautique par exemple) et démontrera la réactivité de notre approche de génération de trajectoires en ligne.

La structure de notre montage a été dimensionnée pour pouvoir être utilisée dans les cas d'usinage dans de l'alliage d'aluminium. Dans ces situations, les efforts de coupes sont négligeables (rapport de 1 pour 10) si nous utilisons un système de déplacement de type table croisée, en opposition aux systèmes de type portique (souvent moins robustes).

Concernant l'ordre de grandeur des vitesses de déplacements, nous contraignons notre choix à la fois par l'application et la capacité temps réel de XPAIM.

b. Quelles vitesses voulons-nous atteindre ?

L'objectif est de déterminer la vitesse de modification de position de pièce maximale en fonction d'une tolérance d'usinage.

Le cas le plus défavorable est lorsque l'outil suit une trajectoire rectiligne (suivant X par exemple) et que la pièce est perturbée suivant un axe transversal (suivant Y dans ce cas).

Nos données d'entrée prises en compte sont :

- Temps de cycle de la machine : $T_{cy} = 0,01 \text{ s}$ – Ce temps correspond à la période de lecture des consignes de position.
- Tolérance d'usinage : Ébauche $Tu = 0,7 \text{ mm}$, Finition $Tu = 0,05 \text{ mm}$
- Profil de modification de position de la pièce : Accélération infini (Dirac) et vitesse constante
- Accélération maximale sur l'axe Y : $Acc_y = 1 \text{ m/s}^2$

Afin de déterminer la vitesse de déplacement maximale de la pièce V_{max} , telle que la trace laissée sur la pièce Dep_y soit inférieure à la tolérance d'usinage Tu , nous faisons l'hypothèse que le temps de calcul de la nouvelle modification T_m est égal au temps de cycle de notre machine T_{cy} . Ce temps de calcul T_m correspond à la durée avant laquelle la machine est informée d'une éventuelle modification de trajectoires. Cette hypothèse sera validée par la suite.

Un simulateur informatique de résolution analytique et de visualisation des résultats a été développé sous Matlab. L'algorithme développé, sur la base des données d'entrée permet de générer le profil de vitesse de déplacement de la pièce maximale puis de calculer en fonction de ce profil, la vitesse de déplacement de l'axe Y (par itération). Ensuite, celui-ci calcule la trace laissée sur la pièce en intégrant la différence des deux profils précédents. Enfin, l'algorithme trace l'ensemble des résultats et donne le décalage maximum en position suivant Y. Cet outil est l'équivalent d'un simulateur de perturbations et de défauts.

Concernant le calcul de la trace laissée sur la pièce (différence entre le déplacement dû à la perturbation de la pièce et la valeur de compensation sur l'axe Y), il est nécessaire de considérer deux sources de retard de compensation. Une liée au temps de réaction réduit au temps de cycle et l'autre liée au temps de rattrapage corrélé au dynamisme de la machine (accélération de l'axe Y). Nous pouvons séparer ces deux phénomènes. Par exemple, dans le cas de l'ébauche, la Figure 46 illustre le cas où seul le temps de cycle est pris en compte (avec une accélération infinie). La trace laissée sur la pièce dans ce cas est de 0,29 mm avec les valeurs de paramètres données précédemment.

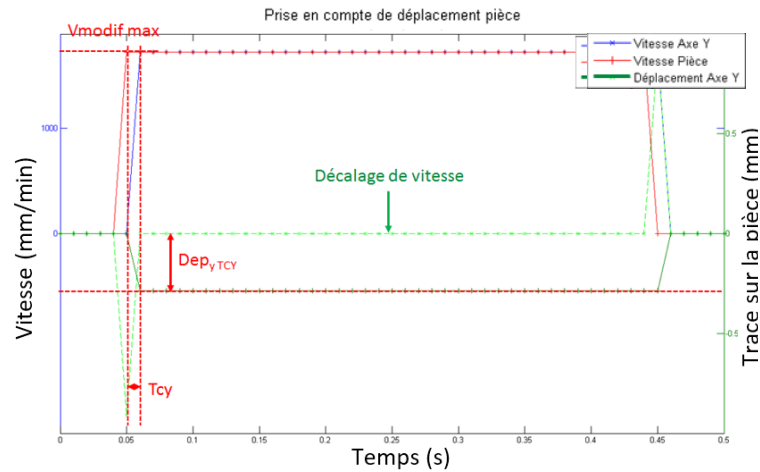


Figure 46 - Trace suivant Y dû au temps de cycle (accélération infinie)

Ensuite, la Figure 47 illustre le cas où la réaction de compensation à la perturbation est immédiate mais l'accélération de l'axe limitée. Nous déterminons alors une trace de 0,41 mm.

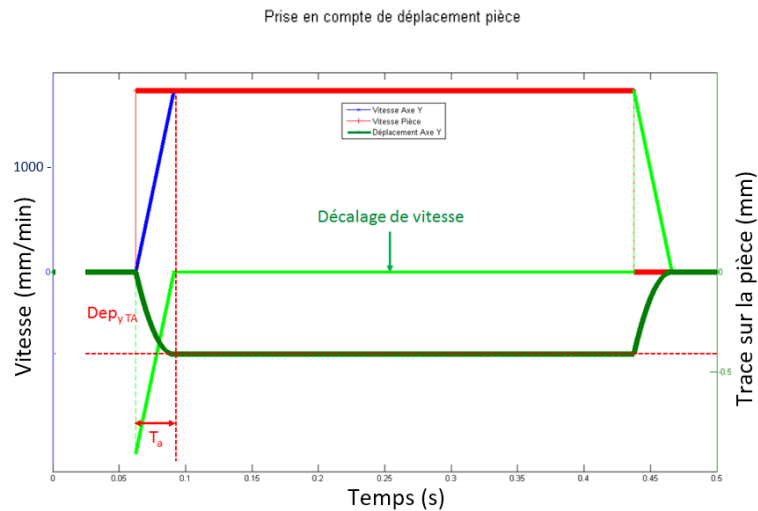


Figure 47 - Trace suivant Y dû au temps de rattrapage (accélération 1m/s^2)

La somme des deux déplacements obtenus donne une valeur de 0,7 mm, qui correspond bien à la tolérance T_u visée, et ceci pour une vitesse de 1700 mm/min.

Dans le cas de la finition, l'influence des deux sources de retard est illustrée sur la Figure 48. Les déplacements étant plus faibles, le temps de réaction a plus d'influence que le temps de rattrapage.

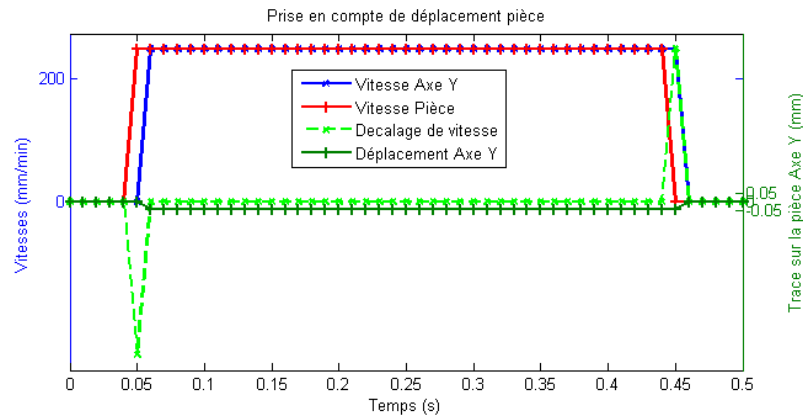


Figure 48 - Compensation de perturbation transversale durant une finition

Pour résumer, les capacités de compensation temps réel de notre plateforme XPAIM permettent selon nos simulations d'avoir une vitesse maximale de perturbation d'environ 1700 mm/min en ébauche et d'environ 250 mm/min en finition.

Dans le cadre de notre développement expérimental, le système de déplacement choisi doit donc permettre d'atteindre une vitesse de déplacement de minimum 1700mm/min.

c. Cinématique du montage expérimental

Le montage doit permettre des déplacements suivants X et Y, dans les deux sens. Afin d'éviter une réversibilité du montage (qui nécessiterait la gestion d'un frein), nous faisons le choix d'un système de déplacement par vis-écrou. Les déplacements seront opérés manuellement à l'aide de manivelles (utiliser des moteurs complexifie le système sans plus-value de démonstration). Ces déplacements seront mesurés à l'aide de codeurs : cette technologie étant déjà employée sur la plateforme XPAIM pour le fonctionnement en configuration non modifiée, la connexion est donc possible par analogie. Afin d'avoir une bonne vision de l'ensemble, la Figure 49 reprend tous ces éléments.

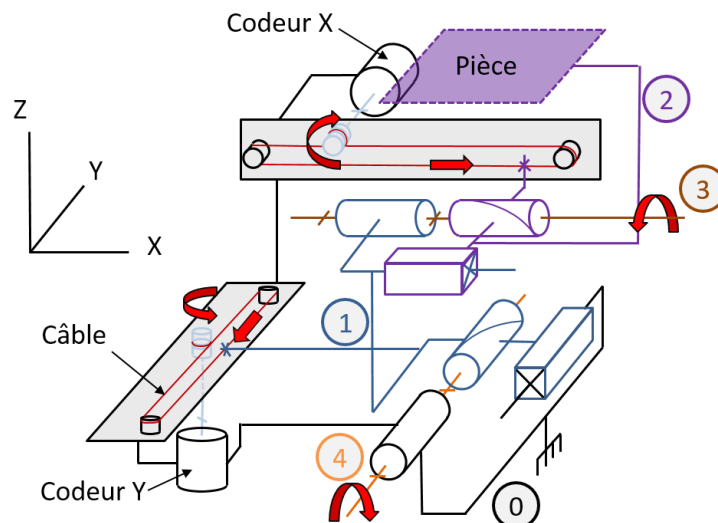


Figure 49 - Schéma de fonctionnement du montage expérimental

Sur ce schéma, l'ensemble {1+2+3+4} correspond à une table croisée classique. Cet ensemble est fixé à la table de la machine {0}. A l'aide de plaques usinées et d'un système de câblage (Figure 50), les codeurs relèvent la position en X et Y des différents éléments mobiles de la table croisée. Ce système permet de compenser la souplesse et le jeu à l'inversion de la table (problèmes rencontrés si les codeurs étaient installés en bout de manivelle par exemple). Plus le système de mesure est proche de l'élément mobile, plus la mesure est précise.



Figure 50 - Vue d'ensemble du montage expérimental de perturbation

Dans cette partie, nous avons mis en place un montage physique dédié permettant de maintenir une pièce et de perturber sa position au sein d'une MOCN. Ce développement est nécessaire dans le cadre de la démonstration d'actions correctives souhaitée. Nous allons maintenant aborder l'intégration de cet ensemble au sein de XPAIM. Ceci permettra de démontrer dans le même temps le caractère modifiable de la CNO et de rentrer dans le détail du fonctionnement de l'interface HAL.

3.2.2. Développement logiciel du montage expérimental

Dans cette partie, nous allons montrer nos développements de la CNO permettant de prendre en compte des perturbations en temps réel.

En étudiant l'architecture de LinuxCNC (Figure 51), diverses possibilités permettant une prise en compte temps réel apparaissent : au sein du module EMCMOT ou au sein de l'interface HAL. Précisons que la prise en compte de modifications ne peut pas être faite en temps réel par le module EMCTASK (interpréteur). La trajectoire d'entrée est donc unique, il n'y a pas de réécriture et de réinterprétation du programme par la CN (le programme reste générique). Nous avons donc choisi une modification liée à la cinématique de la machine.

La première possibilité consiste à modifier les composants décrivant la cinématique directe et inverse de la machine. Ces composants sont modifiables dans les fichiers sources de LinuxCNC dans le langage CC (Canonique Code) [LinuxCNC_Kinematics14] et définissent le passage de l'espace cartésien (lié à la pièce) à l'espace articulaire (lié aux articulations de la machine). Il est donc possible de modifier les relations cinématiques existantes ou d'ajouter des articulations fictives qui influenceraient le comportement de la machine. Cette solution n'a pas été choisie car la modification des fichiers sources nécessitent une recompilation de LinuxCNC sur la machine. De plus, cette dernière solution implique un risque d'avoir un logiciel moins stable mais surtout une impossibilité de passer aisément d'une configuration « normale » à une configuration « compensatrice ».

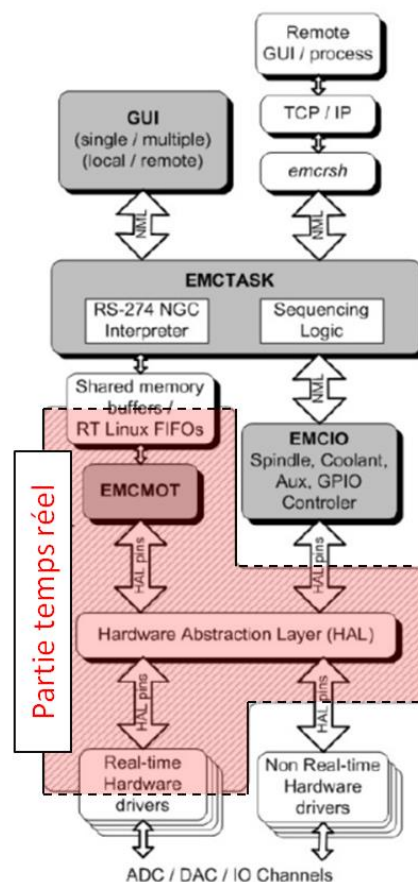


Figure 51 - Architecture de LinuxCNC avec zone temps réel [Staroveski2009]

Notre choix se porte donc sur la modification de la configuration HAL. Avant de rentrer dans le détail des modifications, nous présentons le fonctionnement de HAL succinctement.

a. Fonctionnement de HAL

Hal permet d'étendre le plan de câblage d'une MOCN à l'intérieur du contrôleur [LinuxCNC_HAL14]. Tous les éléments matériels (câbles, connecteurs, ...) ainsi que les

appareils de mesures ont leurs équivalents logiciels dans HAL. Les documentations HAL de la communauté LinuxCNC sont très complètes, mais restent dans l'ensemble très textuelles et peu imagées. Pour bien comprendre son fonctionnement, tous les composants et éléments de configuration de HAL sont représentés sur la Figure 52.

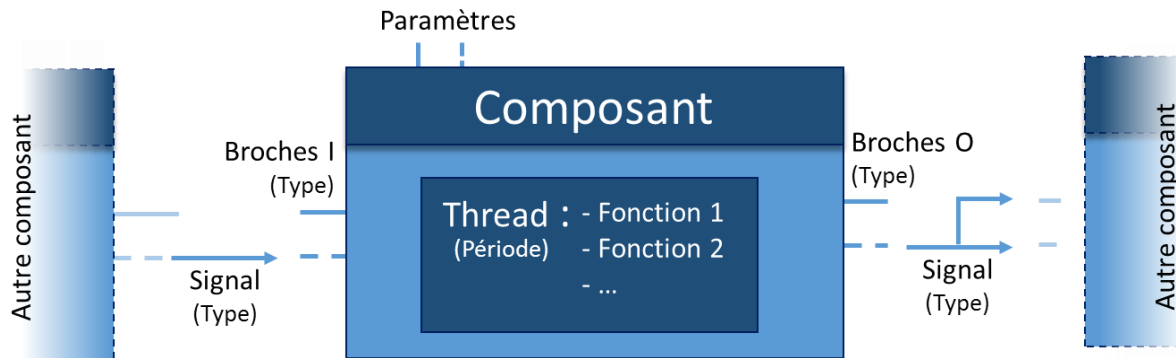


Figure 52 - Différents éléments de HAL

Dans le HAL, nous retrouvons des **composants** (*components*), équivalents de pièces logicielles (ex. PID). Ces composants sont constitués de différentes **broches** (*pin*), d'entrées et de sorties, permettant d'être reliés à d'autres composants logiciels, mais aussi composants réels (*physical_pin*), par l'intermédiaire de **signaux**, dont le **type** (bit, float, ...) doit correspondre à celui des broches. A l'intérieur des composants, nous retrouvons un ensemble de **fonctions** (*functions*), qui ont pour rôle de transformer les valeurs d'entrée en valeurs de sortie particulières (par exemple, une sortie peut être la différence de deux entrées). Ces fonctions sont exécutées dans un ordre particulier et un intervalle de temps (**période**) défini par un **thread**. Enfin, les composants sont réglables par l'intermédiaire de **paramètres** (*parameters*) qui généralement correspondent à des gains (type potentiomètre).

b. Configuration HAL au sein de XPAIM

Les différents composants et fonctions sont définis dans les codes sources de LinuxCNC (langage CC), et il faut donc les considérer comme des briques élémentaires. La configuration HAL consiste donc à relier ces différentes briques entre elles, afin de constituer un système complexe capable de piloter une MOCN. Par ailleurs, les principaux paramètres (cinématiques, asservissements) sont définis dans un fichier d'initialisation (.ini) précisant aussi les fichiers HAL à considérer (au démarrage de LinuxCNC).

Dans le cas de notre plateforme CNO, à l'aide d'une base existante de configuration de machine trois axes, l'ensemble de la configuration en fonctionnement normal a été réalisé.

Nous pouvons par exemple illustrer l'équivalence de quelques lignes de codes et le comportement d'un axe asservi sous forme graphique (Figure 53).

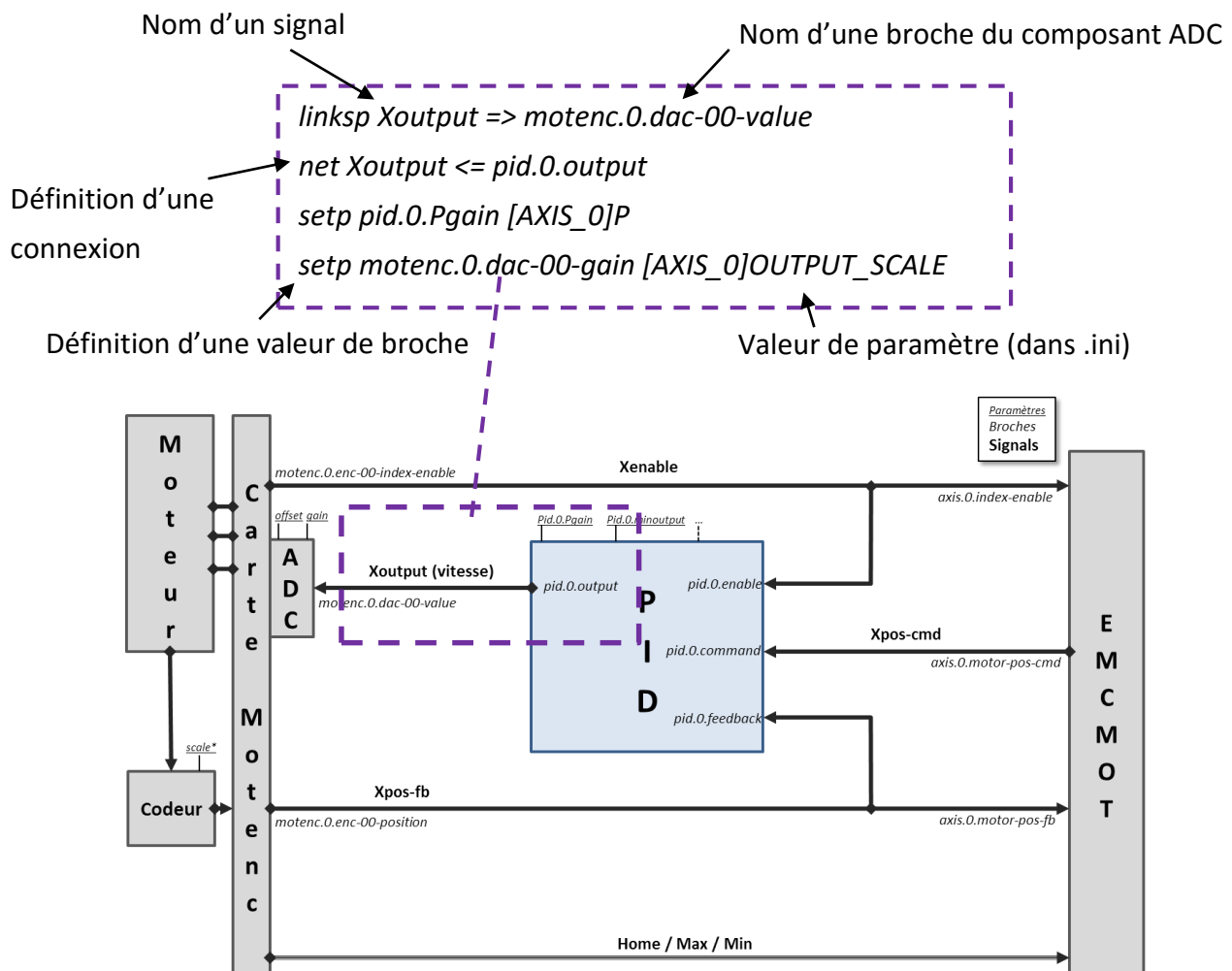


Figure 53 - Configuration de l'asservissement de l'axe X de la plateforme CNO

Nous pouvons alors remarquer qu'à partir d'éléments simples, nous obtenons une configuration complexe en observant seulement l'ensemble des trois axes asservis (Figure 54). A cela, il faut alors ajouter la gestion des entrées/sorties analogiques et la gestion de la broche pour avoir la configuration complète.

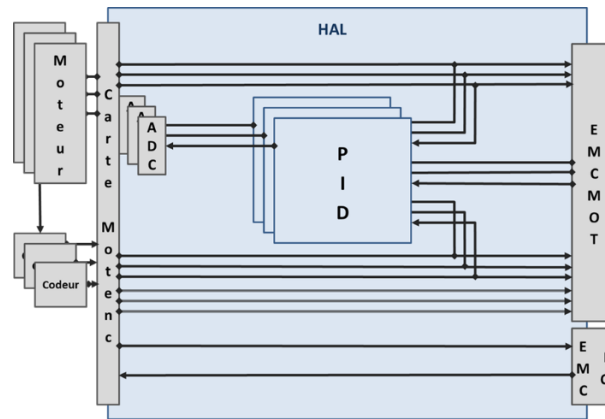


Figure 54 - Illustration du câblage logiciel de 3 trois axes asservis

La compréhension de la configuration de notre plateforme CNO, en fonctionnement normal, nous permet d'envisager différentes possibilités de modifications permettant des actions correctives. Nous pouvons choisir de modifier directement les composants (avec des fonctions existantes) ou de modifier des fonctions (dans des composants existants). Cependant, tout comme dans le cas du module EMCMOT, ces deux premières possibilités nécessitent une recompilation de LinuxCNC. Une dernière possibilité consiste à profiter du caractère modulaire de HAL en modifiant les connexions et en ajoutant des composants déjà existants. C'est cette dernière solution que nous choisissons car elle permet d'avoir les mêmes résultats que les deux premières possibilités (la connexion de plusieurs composants simples peut donner le même résultat qu'un seul nouveau composant avec une fonction plus complexe) sans nécessité de recompiler.

Une possibilité d'ébauche de modification serait d'agir sur une configuration de simulation (machine fictive et entrée fictive), ceci permettant alors de tester différentes solutions et ensuite de les appliquer à la configuration réelle. Cependant, tous les travaux sur les configurations fictives sont inexploitable de façon directe, car les configurations cinématiques fonctionnelles les plus proches ne font pas apparaître de différence entre positions commandées et retours de positions. En effet, comme nous pouvons le voir dans un exemple de configuration simulée, sur la Figure 55, la position de commande est envoyée par le module EMCMOT à des composants de dérivation (*ddt.x*) qui permettent d'obtenir les vitesses et accélérations des axes. Le retour (*feedback*) est relié directement au module EMCMOT (*axis.motor-pos-fb*) sans passer pour une boucle d'asservissement (visible sur la Figure 53). Le composant *comp* permet de déterminer si les axes sont bien situés dans les limites atteignables par la machine simulée.

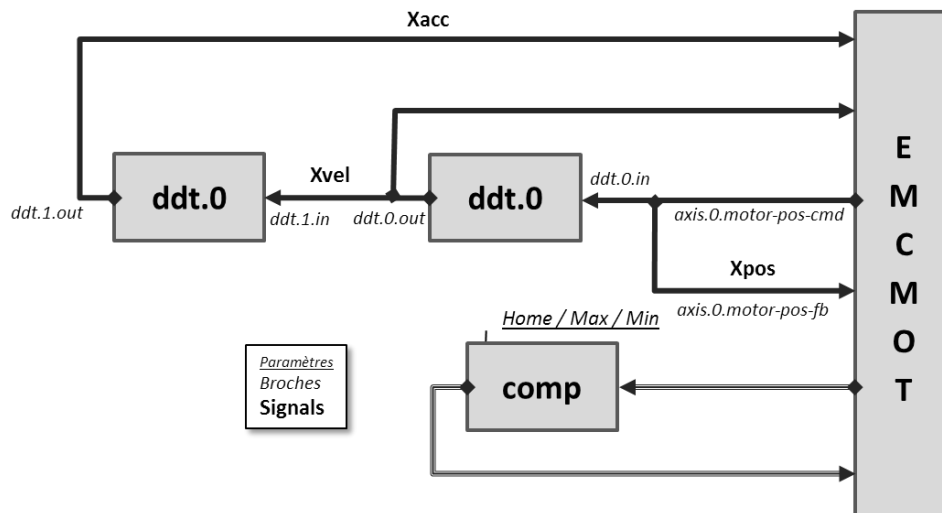


Figure 55 - Illustration de la configuration simulée du déplacement d'un axe X

Ces modèles permettent seulement une simulation de positions en temps réels et un respect des contraintes cinématiques, mais pas de possibilités de prise en compte de perturbations.

Il s'avère que les configurations les plus avancées de simulation, faisant intervenir des codeurs simulés et des boucles PID, sont plus complexes à faire fonctionner que les configurations avec des câblages réels.

Les tentatives de modifications des configurations de simulations ont tout de même été riches en enseignements (sur le fonctionnement global de HAL) et nous ont permis de passer à des modifications sur la configuration réelle par étapes (entrées fictives puis entrées codeurs).

Notre choix s'est porté sur l'ajout d'un composant permettant de sommer la valeur relevée par un codeur additionnel (intégré au montage expérimental) et le retour de position du codeur originel de l'axe concerné sur la MOCN (Figure 56).

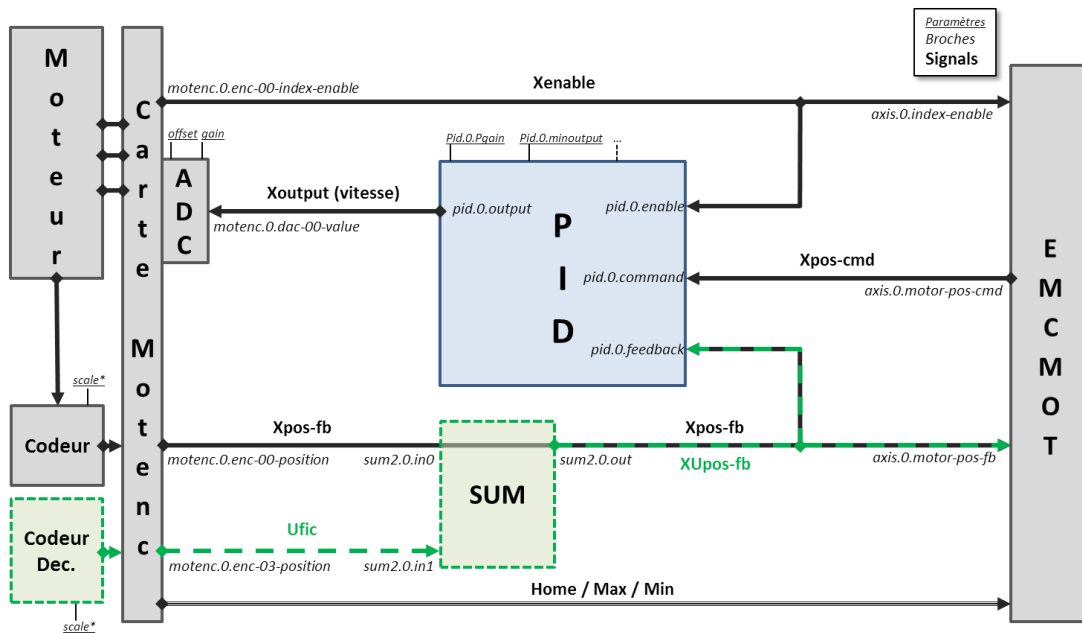
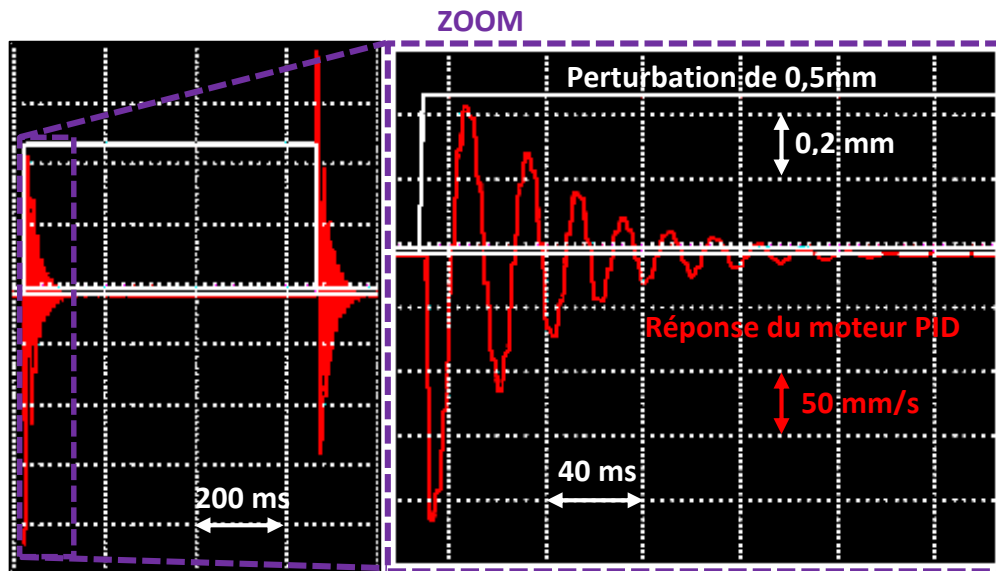


Figure 56 - Ajout d'éléments permettant une compensation temps réel au sein de la configuration d'un axe asservi

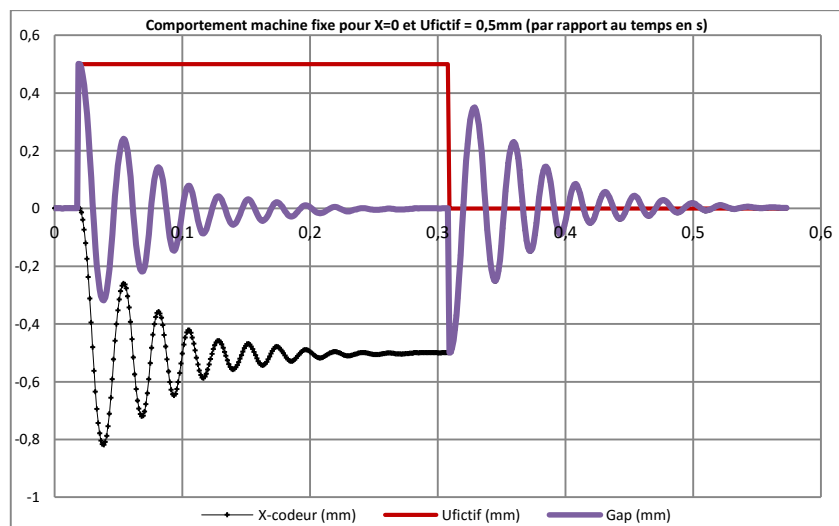
Les modifications apportées à la configuration normale sont de plusieurs ordres :

- Ajout de fonctions de lecture et d'écriture d'une deuxième carte d'axe
- Ajout de deux composants sommes (SUM)
- Définition de la période du thread des fonctions sommes (temps réel $\approx 11\text{ms}$, l'hypothèse de la section 3.2.1.b $T_m = T_{cy}$ est donc validée)
- Déclaration d'un nouveau signal U_{fictif} et Y_{fictif}
- Connexions des entrées des sommateurs aux codeurs originels et additionnels
- Connexion de la sortie des sommateurs à l'entrée feedback des PID

Afin de valider la configuration, une première série d'essais a porté sur l'emploi d'une entrée fictive manuelle ; la configuration étant validée si les moteurs se mettent bien en mouvement lors de l'entrée d'une valeur fictive. Les résultats sont observables au pied de la machine (mise en mouvement de l'axe visible et audible) mais aussi à l'aide de l'oscilloscope intégré à LinuxCNC (Figure 57).



Nous pouvons aussi effectuer l'acquisition de la position codeur et de la sortie du PID afin de mieux exploiter les résultats, comme nous pouvons le voir sur la Figure 58.



Ces premiers résultats expérimentaux montrent qu'il y a une réponse immédiate à la perturbation en sortie de boucle PID. Cependant, due à la perturbation brutale définie fictivement (passage de 0 à 0,5 mm en 11ms), il y a présence de plusieurs oscillations qui sont amorties en environ 0,25 s. Nous constatons par ailleurs que la vitesse de rectification, qui correspond à la vitesse nécessaire à la compensation de la perturbation, a une valeur maximale de 120mm/min, qui est une valeur relativement faible, nous ne sommes donc pas limités par la configuration cinématique de la machine.

La configuration modifiée est donc validée pour une entrée fictive. Dans un second temps, les mêmes essais ont été réalisés avec les codeurs additionnels (sans table) pour valider cette fois-ci la prise en compte de l'interface matérielle-logicielle et observer un comportement avec des perturbations plus réelles (pas d'accélération et vitesse infinie et comportement aléatoire) comme nous pouvons le voir sur la Figure 59 .

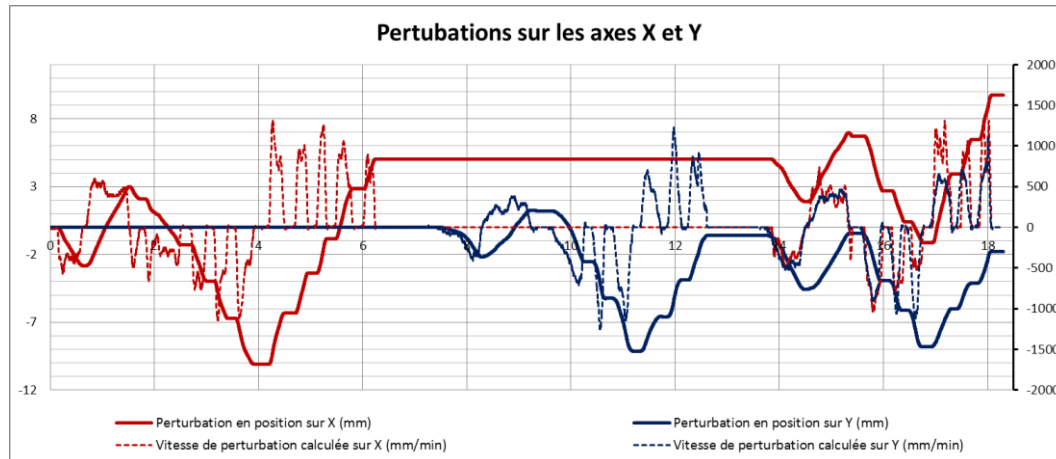


Figure 59 - Acquisitions de la position des codeurs et vitesse associée

Sur cette figure, nous pouvons observer que les amplitudes de perturbation peuvent être très grandes en position (des bonds d'environ 3 mm) et les vitesses associées peuvent dépasser les 1000 mm/min. Il est possible d'observer le comportement de la machine à vide et sans la table, en agissant seulement sur les codeurs. Cependant, ces observations ne sont pas retranscrites dans ce manuscrit, afin de se consacrer sur la démonstration durant un usinage. Nous pouvons préciser que le test de la configuration avec des entrées codeurs a été concluant (mouvement de l'axe en outil en adéquation avec ceux des codeurs) et que la configuration est donc validée.

3.2.3. Conclusion de la démarche de mise en place du montage expérimental d'actions correctives

Par les différentes étapes de mise en place, nous avons pu observer les possibilités et difficultés de modification de plateforme CNO. La démarche effectuée est illustrée sur la Figure 60.

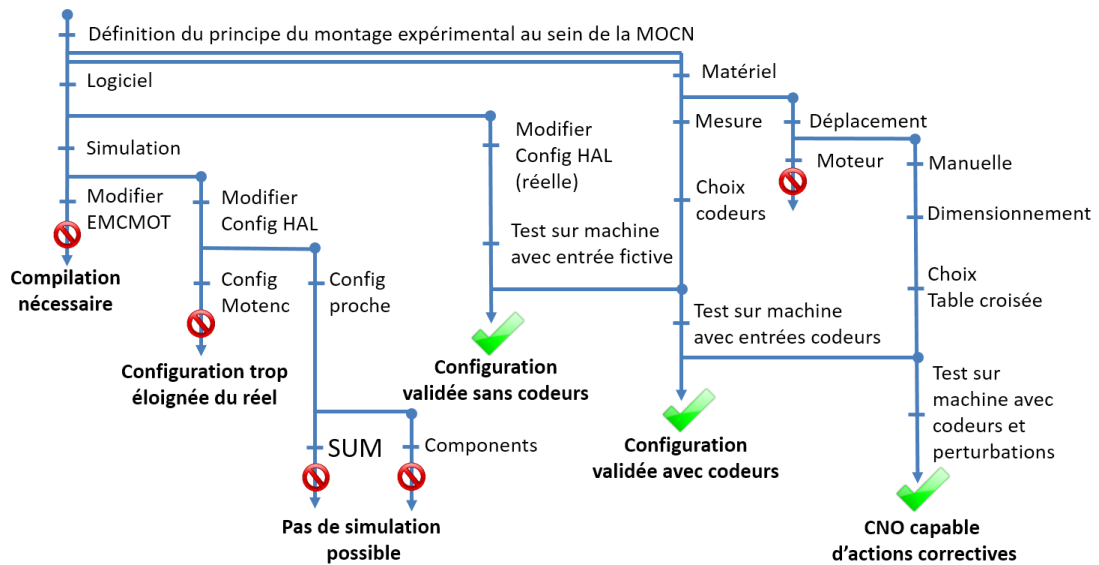


Figure 60 - Etapes de développements effectués du montage expérimental

Sur ce graphique, nous retrouvons les points bloquants du développement logiciel (liés à la difficulté de compilation de EMCOT et aux manques de descriptions réalistes des configurations simulées). Les développements logiciels ont bien aboutis à une configuration validée sans codeurs. Concernant le développement expérimental, une première validation a été effectuée sur l'interface logicielle/matérielle à l'aide des codeurs physiques. Ensuite le choix a été de développer un montage avec possibilité de perturbations manuelles plutôt que motorisées. Au final, nous obtenons bien une plateforme CNO fonctionnelle permettant des actions correctives en temps réel. Le montage expérimental développé permet d'avoir un simulateur de perturbation et de démontrer la fonctionnalité de compensation en ligne de la plateforme XPAIM. Les différentes observations ainsi que les résultats de ces démonstrations sont étudiés dans la section suivante.

3.3. Démonstration d'actions correctives : expérimentations et résultats

Nous allons présenter dans cette section les résultats d'une démonstration de compensation de perturbation en temps réel à l'aide du montage expérimental défini précédemment. Nous allons observer le comportement de la machine face aux perturbations dans le cas d'un outil en position fixe (l'outil doit rester en place malgré les perturbations) puis dans le cas de la réalisation d'un profil carré. Ces usinages sont réalisés tout d'abord dans de la mousse condensée puis dans de l'alliage d'aluminium. Les différents cas sont aussi réalisés sans perturbation pour avoir une référence. Les perturbations sont opérées manuellement et hors machine à l'aide de deux tubes PVC visibles sur la Figure 61.

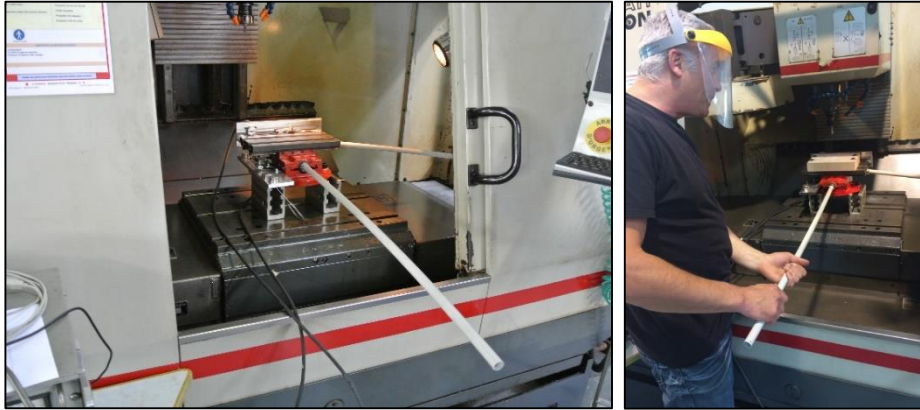


Figure 61 - Montage au sein de la plateforme XPAIM avec système de manipulation hors machine

Avant de réaliser les essais, il est nécessaire de définir le plus précisément possible l'échelle de conversion des données capteurs au sein de la configuration HAL. Pour cela, nous effectuons plusieurs tours de codeurs tout en relevant la position de la table à l'aide d'un palpeur micrométrique (visible sur la Figure 62).



Figure 62 - Réglage de l'échelle de conversion des codeurs

Pour chaque essai, nous faisons une acquisition des données nécessaires à notre étude (position commandée, valeurs des codeurs et sortie du PID entres autres). Ces acquisitions sont comparées à des mesures réalisées sur MMT dans le cas de l'usinage de la pièce en alliage d'aluminium. Nous présentons tout d'abord les résultats de l'usinage d'une rainure dans de la mousse condensée.

3.3.1. Démonstration de compensations en ligne : des résultats bien visibles

La première série d'essais est réalisée dans de la mousse condensée pour pouvoir observer le comportement du montage durant un usinage avec des forces spécifiques faibles. C'est seulement dans ce cas que nous réalisons des usinages avec la configuration normale de la

XPAIM (sans compensation) sous perturbations, afin d'avoir des résultats visuels sans crainte d'endommager la machine. Ces résultats sont visibles sur la Figure 63.

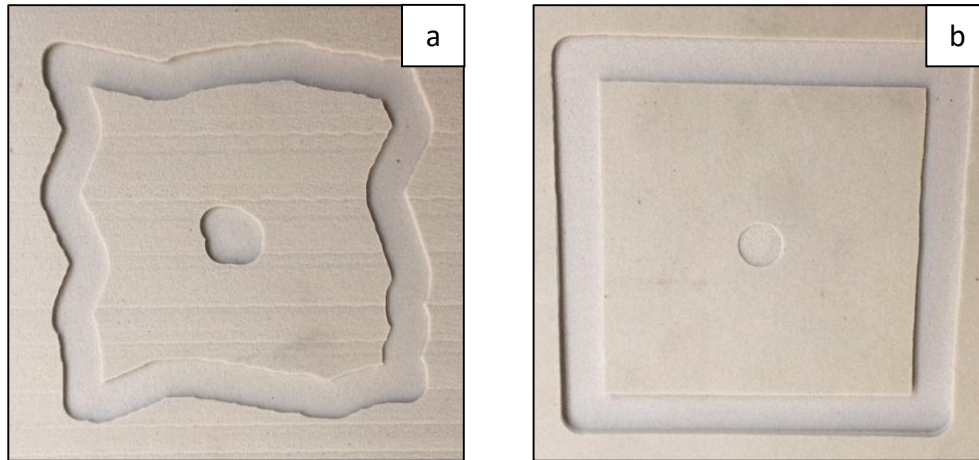


Figure 63 - Effet visible de la compensation : carré de 100mmx100mm perturbé-non compensé (a) et perturbé-compensé (b)

La Figure 63.a correspond à la photographie du profil réalisé dans le cas d'un usinage subissant des perturbations sans compensation. Nous observons nettement que le profil voulu n'est pas respecté et que les perturbations sont importantes. La Figure 63.b correspond à la photographie du profil réalisé avec exactement les mêmes conditions (même programme d'usinage et perturbations) mais cette fois-ci avec la configuration modifiée permettant la compensation en temps réel. Le résultat est probant, les perturbations sont visiblement bien prises en compte et le profil correspond à celui voulu.

Bien que les acquisitions liées à ces essais soient exploitables, nous faisons le choix de ne pas les présenter dans ce manuscrit pour se focaliser sur le cas de l'usinage sur une pièce en alliage d'aluminium.

3.3.2. Démonstration de compensations en ligne : cas d'un usinage dans des conditions industrielles

Nous avons choisi de réaliser des essais dans de l'alliage d'aluminium car c'est un matériau couramment utilisé dans l'industrie. Dans le cas de notre démonstration, nous souhaitons être dans les conditions de réalisation d'une finition de flanc de poche (engagement radial faible et vitesses élevées). Cependant, le manque de rigidité du montage nous oblige à limiter la vitesse de déplacement de l'outil à une valeur raisonnable (500mm/min) de même pour la profondeur de passe, qui est choisie à 2 mm, pour un engagement radial classique de 0,2 mm.

Dans un premier temps, nous effectuons des essais sans trajectoire spécifiée. L'outil est placé en mode manuel dans une position fixe à 2 mm de profondeur (en Z) dans l'alliage

d'aluminium. Nous réalisons donc un trou du diamètre de l'outil sans perturbation, puis un autre trou en appliquant des perturbations suivant les directions X et Y durant 20 s. Le profil des perturbations est visible sur la Figure 64.

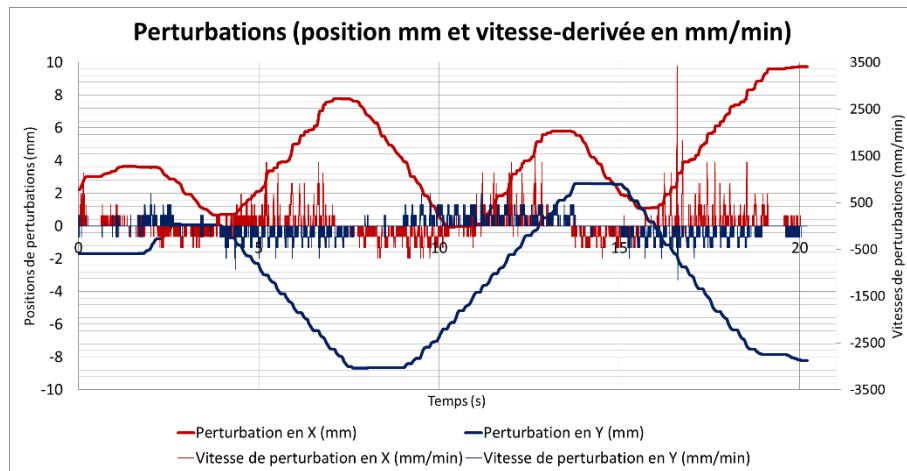


Figure 64 - Profil de perturbations appliqué durant l'usinage en position fixe de l'outil

Nous observons que l'amplitude des perturbations est importante. Des bonds de plusieurs dixièmes de mm à une vitesse supérieure à 1000 mm/min sont courants. Dans l'ensemble, les perturbations effectuées suivant l'axe X sont plus importantes. Nous allons voir si cela est observable sur le profil obtenu. Tout d'abord, au visuel (voir Figure 65), il est difficile de repérer des différences entre le trou obtenu avec perturbations et celui obtenu sans.



Figure 65 - Usinage perturbé-compensé à gauche et usinage non perturbé à droite

Nous exploitons en premier lieu les acquisitions pour analyser la position de l'outil soumis à des perturbations. Sur la Figure 66, nous affichons sur un même graphique le profil obtenu avec compensation (en vert) et celui qui aurait été obtenu sans compensation (en rouge).

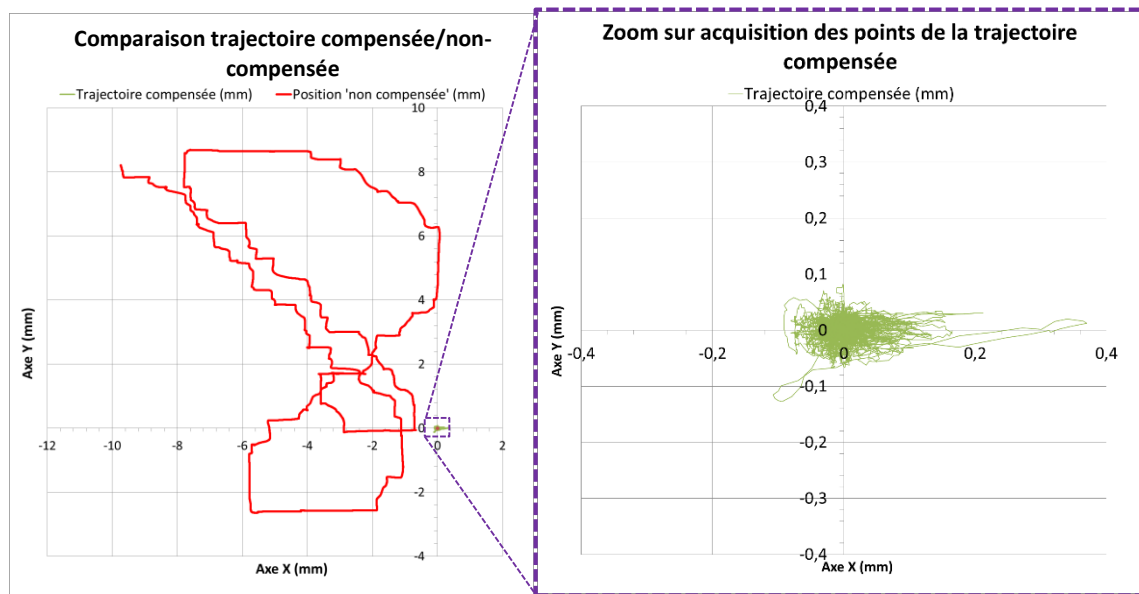


Figure 66 - Acquisition des positions outils durant la perturbation de la position outil fixe

Sur la base de ces acquisitions, nous remarquons que la compensation a été très efficace. La quasi-totalité des positions du centre outil est comprise dans un cercle de diamètre 0,1 mm. L'outil est constamment ramené à sa position d'origine. L'écart le plus important est tout de même de 0,37 mm (suivant X, comme pressenti), mais cela est explicable par la forte perturbation subite (un décalage de la table de montage de 0,5 mm à 3500 mm/min) qui est au-dessus des valeurs souhaitées (NB : En ébauche, vitesse maximale de 1700 mm/min et en finition, vitesse maximale de 250 mm/min). Le premier constat à partir de ces acquisitions est de dire que la plateforme XPAIM permet de compenser assez les perturbations dans le cas d'un usinage en ébauche.

Cependant, il est nécessaire d'observer le profil obtenu par un moyen de mesure physique. Nous utilisons donc une machine à mesure tridimensionnelle (MMT) pour relever un ensemble de points du contour des deux trous usinés (Figure 67).



Figure 67 - Mesure des profils de démonstration sur MMT

Précisons qu'afin de comparer les acquisitions aux mesures réelles, il a été nécessaire de développer un algorithme permettant de passer des positions du centre de l'outil au profil correspondant à l'enveloppe de l'outil. Nous regroupons alors l'ensemble des profils sur la Figure 68.

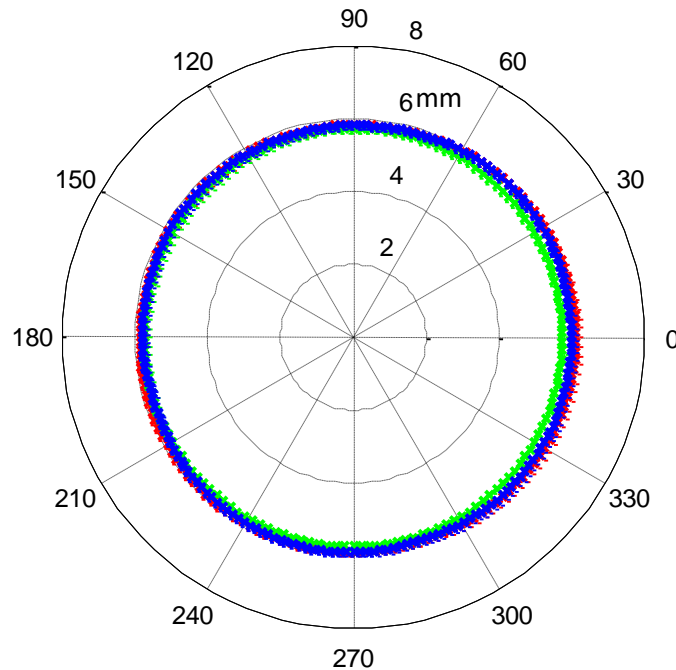


Figure 68 - Affichage des différents profils obtenus : sans perturbations (en vert), avec perturbations (mesure MMT en bleu et acquisitions en ligne en rouge)

Le premier constat est que les profils mesurés et acquis sont très proches (écarts inférieurs à 10%). Ensuite nous remarquons que le profil d'acquisition montre des écarts plus importants au profil sans perturbation. La Figure 69 affiche ces écarts le long du profil quasi-circulaire (écart de circularité de 0,30 mm dans le cas du profil acquis et de 0,25 mm dans le cas du profil mesuré).

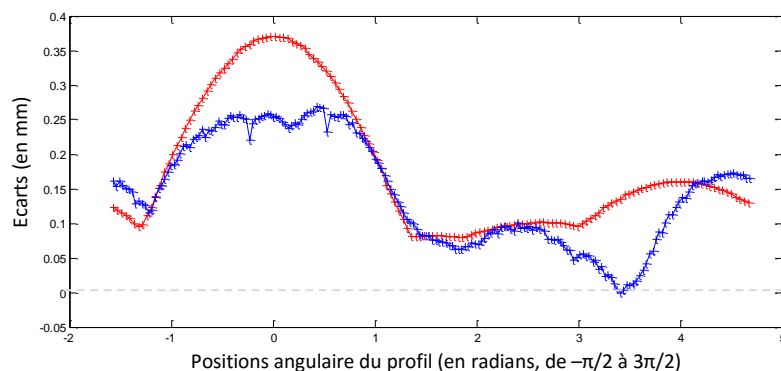


Figure 69 - Ecart du profil acquis (en rouge) et mesuré (en bleu) par rapport au profil idéal (ligne horizontale recalée à 0)

Cet écart peut être justifié par une possible flexion de l'outil durant l'usinage (de 0,1 mm en bout d'outil dans les cas les plus importants) qui amortirait les perturbations subites.

Nous retiendrons que les acquisitions sont à priori une bonne base pour statuer de la conformité de la pièce, les écarts acquis étant supérieurs (d'environ 10%) aux écarts réels. Si les écarts acquis sont supérieurs à la tolérance visée, il est nécessaire de mesurer la pièce ou d'appliquer un facteur de réduction. Dans le cas présent, la tolérance d'usinage en ébauche est respectée. Seule une petite partie du profil respecte la tolérance en finition, mais cela est justifiable par l'intensité des perturbations.

Nous allons maintenant observer le comportement de la plateforme XPAIM durant l'exécution d'une trajectoire. Le protocole est le même que pour le cas précédent sauf que les perturbations sont exécutées durant le déplacement de l'outil. Les profils obtenus sont montrés sur la Figure 70.

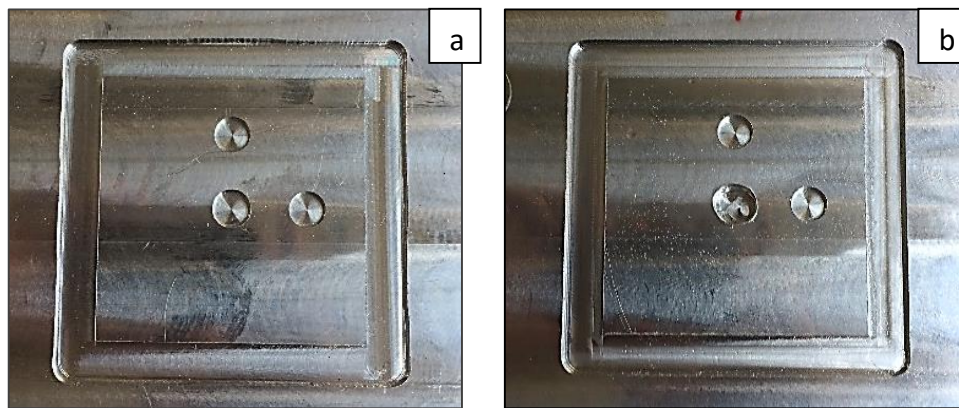


Figure 70 - Usinage perturbé-compensé (face a) et usinage non perturbé (face b)

Les deux cercles les plus proches du profil carré servent de références pour recalibrer les mesures. Le cercle du milieu (face a) a été utilisé précédemment. Le cercle (face b) a un défaut apparent dû à un bris d'outil mais n'est pas utilisé dans nos développements. Les profils à observer en détails sont ceux correspondants au profil extérieur du carré. Comme dans le cas précédent, les perturbations sont importantes (Figure 71) et la compensation est efficace.

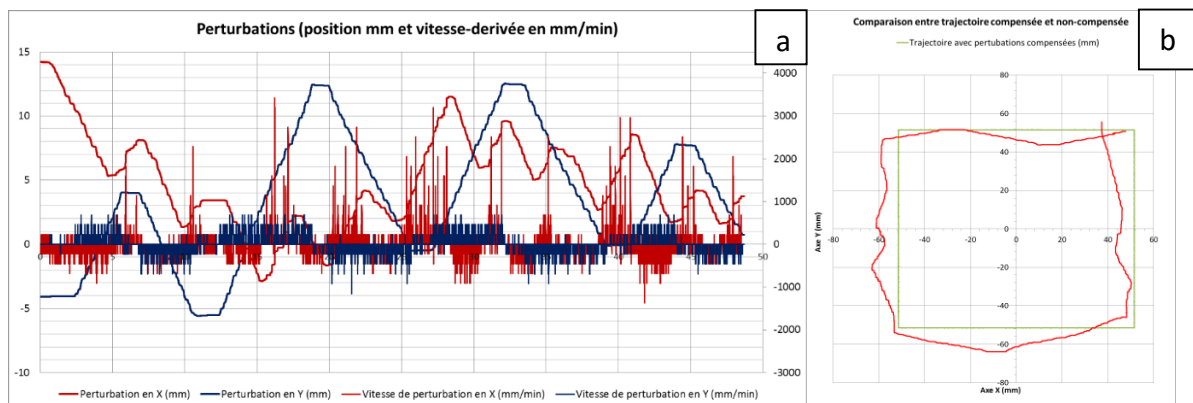


Figure 71 - Profil des perturbations (a) et aperçu du profil supposé sans compensation (b)

En superposant les acquisitions correspondantes à l'usinage sans perturbations et à l'usinage avec perturbations compensées, il est difficile d'observer des écarts (Figure 72).

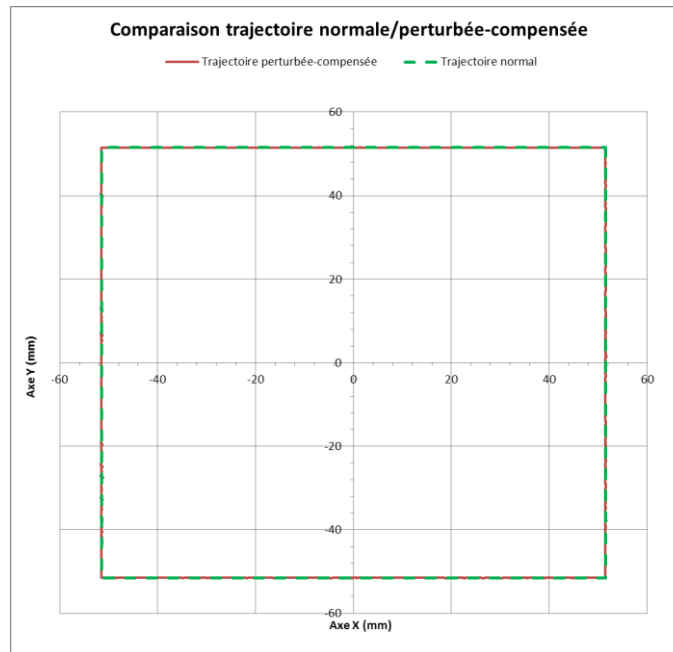


Figure 72 - Comparaison des profils dans le cas d'un usinage sans perturbations (en vert) et avec perturbations (en rouge)

Afin de mettre en évidence ces écarts, nous appliquons un facteur grossissant de 100 sur ces écarts et nous faisons apparaître l'intervalle de tolérance sur cette même échelle dans un même graphique (Figure 73).

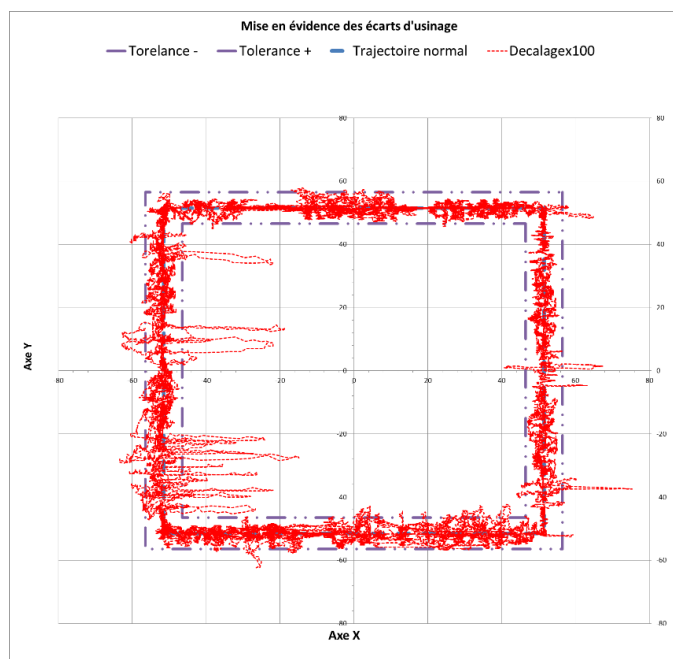


Figure 73 - Mise en évidence de traces liées aux perturbations (accentuation de facteur 100)

Il apparait alors que 96% des points acquisitions sont bien compris dans l'intervalle de tolérance correspondant à une finition. Les écarts maximaux peuvent atteindre jusqu'à 0,35 mm, soit 7 fois supérieurs à la limite tolérée en finition. Cependant, ces écarts sont toujours observés pour des perturbations ayant une vitesse dépassant les 500 mm/min pour des bonds de l'ordre du dixième de mm (Figure 74).

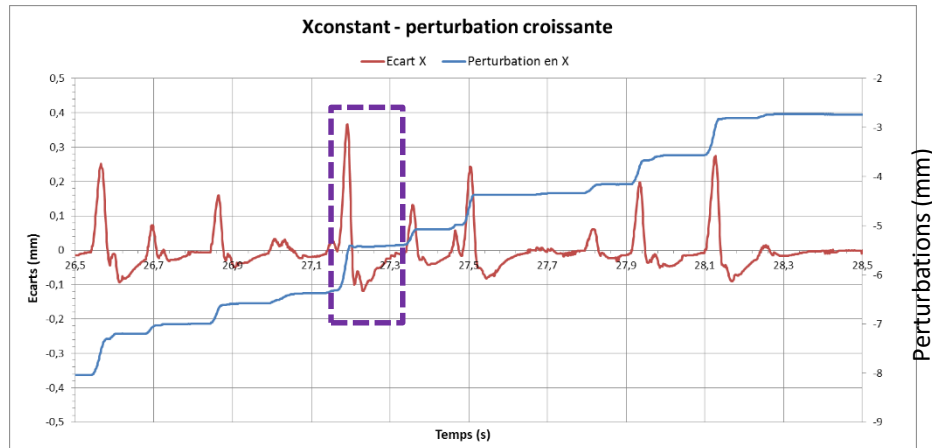


Figure 74 - Illustration des résultats d'écarts en fonction du profil de perturbation sur une portion du profil

Ces perturbations importantes sont dues à la mise en déplacement manuel du montage. Afin de se donner une idée du profil en ne considérant pas ces perturbations, nous obtenons le graphique suivant (Figure 75).

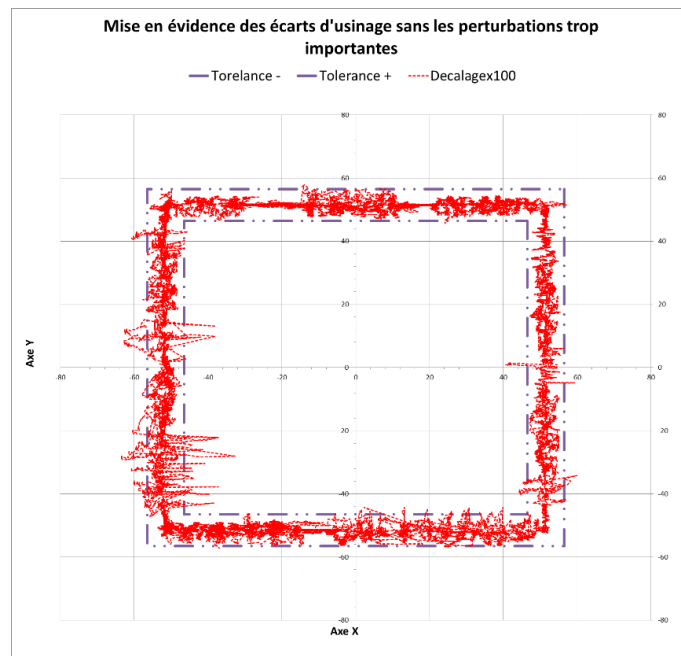


Figure 75 - Mise en évidence de traces liées aux perturbations ayant une vitesse inférieure à 500 mm/min

Dans ce graphique, 99,4% des points sont bien situés dans l'intervalle défini. Et les écarts les plus importants sont dus à l'influence des perturbations filtrées précédemment. L'action de compensation semble donc possible pour une application industrielle au vu des

acquisitions. Mais il est nécessaire de vérifier cela avec les mesures réelles sur MMT. Nous intégrons donc les données mesurées sur des portions de droites sur le graphique suivant (Figure 76).

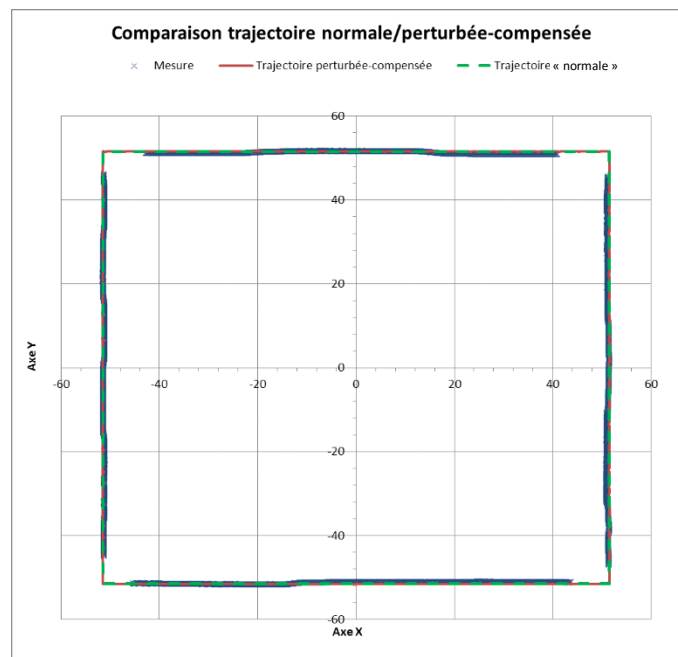


Figure 76 - Visualisation des différents profils d'usinage (mesure en bleu, acquisition CN en rouge, référence en vert)

Cette fois-ci, les écarts sont plus importants que précédemment (jusqu'à 0,5 mm) et ceci sur une grande partie du profil. Une corrélation des deux profils (mesuré et acquis) est possible sur certaines parties (Figure 77) concernant la compensation des perturbations en X.

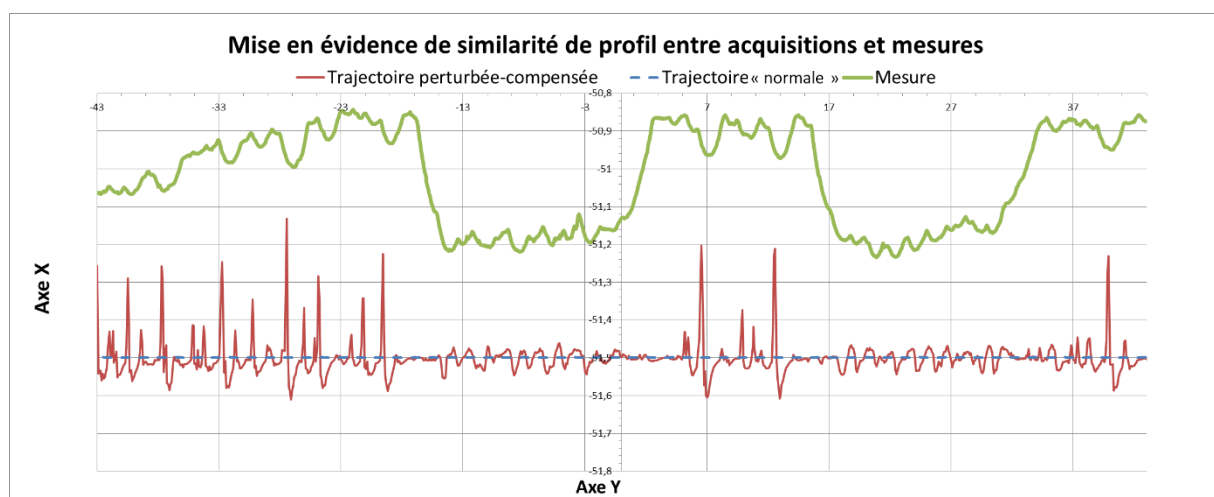


Figure 77 - Similarité entre profil mesuré et profil acquis (compensation suivant X)

Par contre, les différences sont plus importantes pour les compensations suivant l'axe Y (Figure 78).

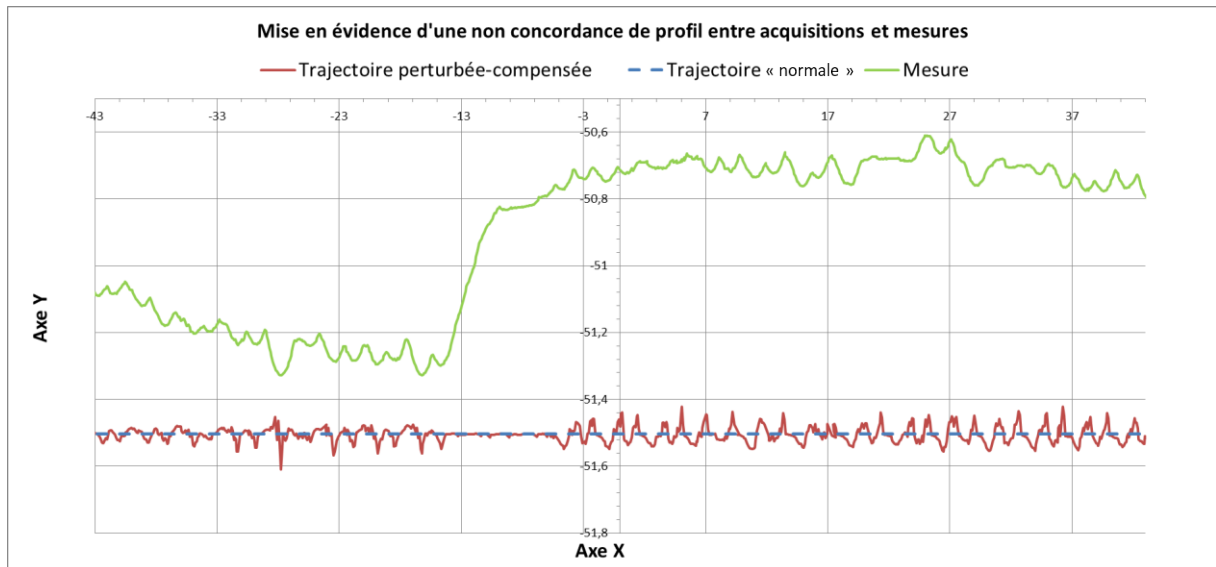


Figure 78 - Similarité entre profil mesuré et profil acquis (compensation suivant Y)

Il peut exister différentes causes à ces écarts. Tout d'abord, les écarts les plus importants sont toujours situés à l'intérieur du profil et correspondent à un sous-usinage. Nous comprenons donc que les déplacements réels sont plus importants hors-matière, ceci peut être imputable au fait qu'il y ait moins de résistance appliquée à l'outil. Il peut aussi y avoir des écarts dû aux réglages des codeurs, mais aussi liés au comportement du montage. L'avantage des codeurs qui sont directement reliés à la structure du montage est qu'il y a une prise en compte de la souplesse du montage. Cependant, la pièce étant maintenue par pressions (4 vis), des décalages (glissements) ont pu avoir lieu durant l'usinage et induire les erreurs observées suivant l'axe Y.

Soulignons que les écarts observés après compensation permettent toujours une récupération de la pièce, car ceux-ci sont observés hors matière.

3.3.3. Démonstration de compensations en ligne : des améliorations possibles

Dans l'ensemble, la démonstration de compensation de perturbations sur XPAIM montre des résultats très prometteurs. Les écarts observés sont toujours acceptables dans le cas d'une ébauche. Concernant une application en finition, la pièce ne présente pas d'écart en sur-usinage pour des perturbations avec une vitesse inférieure à 500 mm/min.

Les résultats sont améliorables s'il y a une prise en compte de la flexion d'outil, ou tout simplement une adaptation des compensations lorsque la perturbation est hors-matière ou non. De même, lorsque la perturbation est dans le sens du parcours d'outil, il n'est pas nécessaire de compenser. Précisons que le temps de parcours reste exactement le même qu'il y ait perturbation ou non. La CN suit toujours les positions commandées en sortie d'interpolateur. Il faudrait une prise en compte de l'entité à réaliser pour gagner en temps et en souplesse. Cela est réalisable dans la configuration HAL, en ajoutant un module type comparateur entre la position commandée et l'entrée du PID. Celui-ci peut permettre de rectifier la commande envoyée selon les cas (hors-matière, dans le parcours ou hors parcours). Une solution plus prometteuse serait envisageable à l'aide du format de description de trajectoires STEP-NC, qui a l'avantage d'être orienté entités et qui réunit donc des informations sur les entités déjà réalisées et à réaliser.

3.4. Conclusion

Dans ce chapitre, nous avons défini une nouvelle approche de génération de trajectoires pouvant compléter les méthodes de programmation avancée déjà existantes. Dans le cadre de cette nouvelle approche, une démarche expérimentale a permis de démontrer l'apport d'une compensation de perturbations en ligne et en temps réel au sein de la plateforme XPAIM. Ceci a permis par ailleurs de montrer l'intérêt de l'utilisation d'une plateforme ouverte. La commande numérique a un rôle plus décisionnel, car avec un programme générique défini en entrée, la trajectoire devient spécifique à la géométrie réelle de la pièce. A partir des développements effectués, différentes évolutions sont possibles, en particulier en intégrant le langage STEP-NC au sein de XPAIM comme nous pouvons le voir sur la Figure 79.

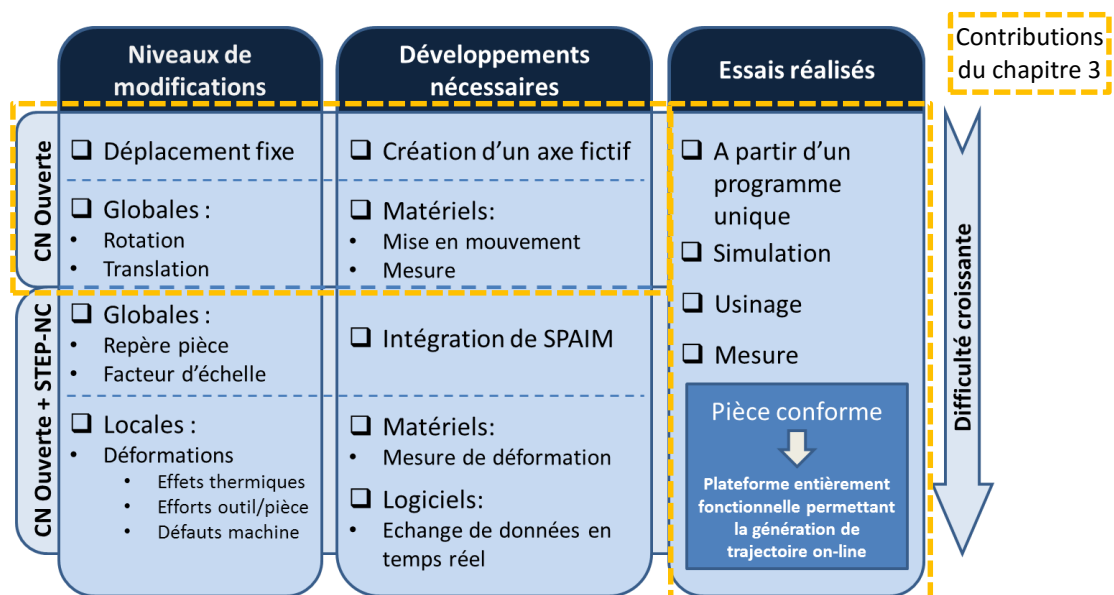


Figure 79 - Contribution de ce chapitre dans une vision plus globale faisant intervenir le langage STEP-NC

Cette nouvelle approche de génération de trajectoires a des répercussions positives sur la chaîne numérique ainsi que dans les perspectives d'une usine du futur dans sa globalité. Ces aspects sont développés dans le chapitre 4.

Chapitre 4

Adéquation de XPAIM avec l'usine du futur

Chapitre 4 – Adéquation de XPAIM avec l'usine du futur

La dynamique de recherche et d'innovation de ces dernières années est illustrée par de nombreux financements publics et privés autour de l'usine du futur. Dans le cadre du septième programme européen [FP7], qui a pris fin en 2013, une priorité a été donnée au développement de nouvelles technologies de production en fort lien avec l'évolution des technologies de l'information et de communication (TIC). Cette association donne naissance au concept d'usine digitale, intelligente et virtuelle qui améliore la compétitivité des industries européennes. Entre autres, des efforts sont employés à réduire le temps de mise sur le marché des produits, à automatiser la production ou encore à augmenter et à stabiliser l'obtention de la conformité des produits manufacturés. Cette dynamique poursuit son élan avec le programme Horizon 2020 [H2020] dont l'un des axes de recherche est consacré à la primauté industrielle. L'industrie doit être propre, ultra performante et durable à travers l'utilisation de technologies innovantes et d'une bonne gestion des informations au sein de l'usine.

L'objectif de ce chapitre est de démontrer l'adéquation de nos travaux avec ce cadre. Nous présentons dans un premier temps un champ d'application élargie de notre plateforme XPAIM (section 4.1). L'exploitation de l'ensemble des applications n'est possible que dans le cas d'une gestion intégrée des données entre tous les acteurs d'une usine. Ceci nous amène à synthétiser les travaux autour de l'usine du futur (section 4.2), tout d'abord dans sa globalité (section 4.2.1) puis nous nous focalisons sur la chaîne numérique de fabrication (section 4.2.2). Enfin, nous abordons la notion de CN étendue ; dans laquelle de nombreux modules sont en adéquation avec nos travaux (section 4.2.3).

4.1. Concept XPAIM : Un champ d'application étendu

4.1.1. Les différents niveaux d'information observables en temps réel sur XPAIM et possibilités associées

A l'aide de l'approche développée relative à la prise en compte de perturbations et de compensation en temps réel au sein de la MOCN, nous répondons à un certain besoin d'adaptation de la trajectoire durant l'usinage d'une pièce, généralisable à de nombreux process de fabrication. Grâce à l'ouverture de la CN, nous avons un lien direct possible entre l'observation du process et le comportement des axes de la MOCN. Nous permettant ainsi de compenser en temps réel des perturbations aux origines diverses durant le processus de fabrication (Figure 80).

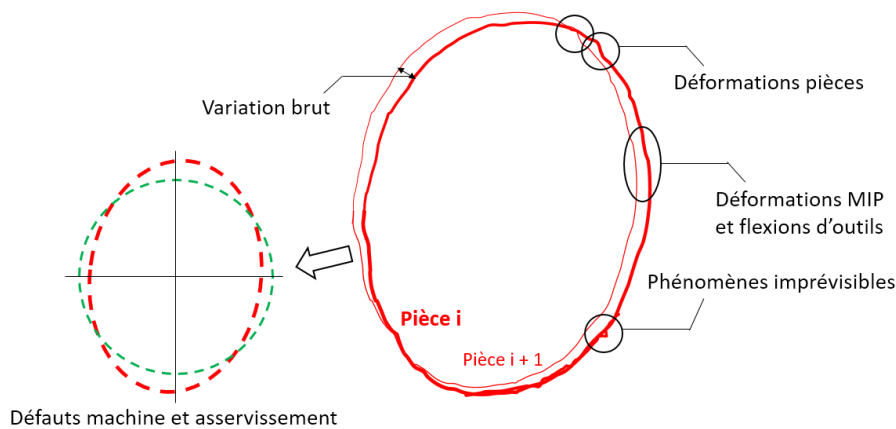


Figure 80 - Illustrations des défauts observables et compensables en temps réels

Les informations observées sont exploitables directement par la CN mais servent aussi à compléter les modèles de connaissance et de prédiction du process comme nous pouvons l'observer sur la Figure 81.

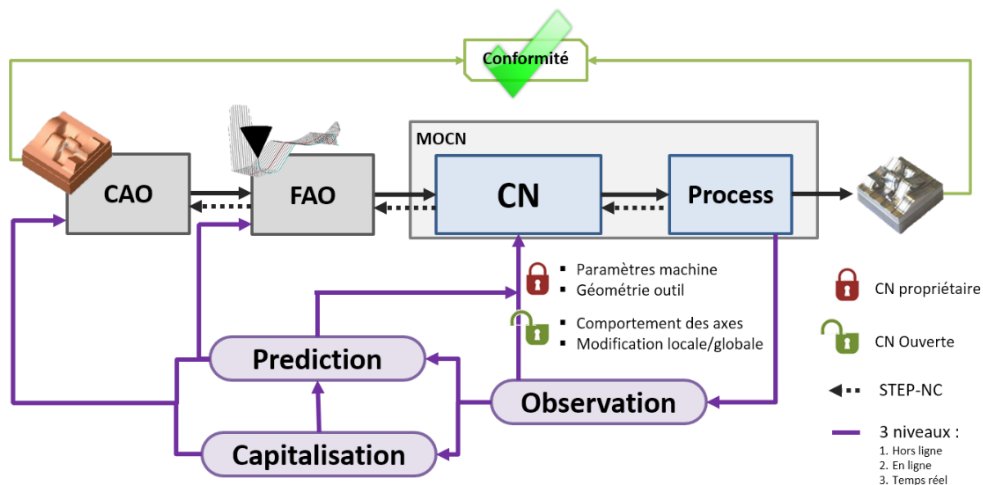


Figure 81- Capitalisation, prédiction et observation du process au sein de la chaîne numérique

Il y a une distinction à faire entre les données exploitables concernant la pièce en cours de réalisation et celles capitalisables pour l'ensemble de la production. Il est alors nécessaire de séparer ces informations par niveau (Figure 82). Nous pouvons identifier un niveau d'information (1) propre à la pièce i , un niveau (2) propre à la réalisation des pièces suivantes $i+N$ et enfin un niveau (3) concernant l'état de la machine durant la production.

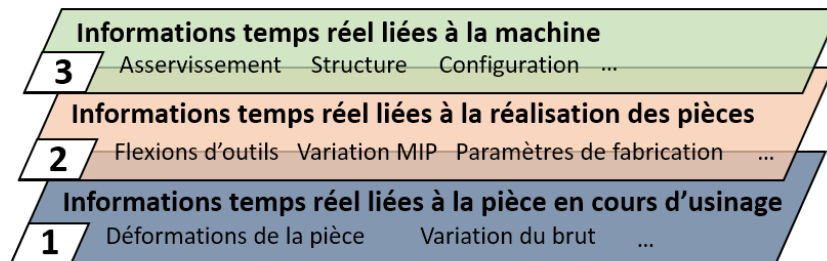


Figure 82 – Différents niveaux d'information exploitables

Au cours d'un premier usinage, toutes les informations sont utilisées pour que la conformité de la pièce soit respectée. Dans le cas de notre démonstration, les perturbations sur les axes X et Y sont compensées en temps réel. Les informations observées sont directement exploitées. Ceci correspond au premier niveau d'informations (liée aux phénomènes difficilement prévisibles de déformations de la pièce et de variations du brut).

Durant toute la production, certaines informations observées seront similaires d'une pièce à une autre. L'inspection de la pièce peut montrer un écart géométrique entre le profil voulu et le profil obtenu qui peut être dû aux déformations de l'outil ou du montage de mise en position (MIP), nécessitant alors une adaptation de la trajectoire. Les paramètres de fabrication peuvent aussi être adaptés afin de diminuer les efforts de coupe, ou encore la trajectoire peut être modifiée pour faciliter le passage de discontinuités. Ces adaptations sont liées au deuxième niveau d'information qui concerne tous les éléments invariants (présents durant toute la production) permettant la réalisation d'une pièce (outil, montage, trajectoire).

Le dernier niveau concerne les informations liées au comportement de la machine. Du fait de l'ouverture de la commande numérique, il est possible de modifier le comportement de la machine, de corriger des erreurs d'asservissement, de compenser des défauts de structure. Nous avons vu dans le cas de la démonstration sur XPAIM qu'il est possible de sommer des informations sur les « feedbacks » de position à l'aide de codeurs additionnels. Dans le même esprit, il est possible de définir des fonctions plus complexes (dépendants des positions courantes de l'outil) qui modifient les positions commandées de la machine et compensent un défaut d'orthogonalité par exemple. Ce réglage est difficilement

imaginable en cours de production, mais tout à fait envisageable dans le cas d'un diagnostic Ballbar durant lequel les composantes des fonctions de compensations seraient évaluées et intégrées en temps réel jusqu'à une dissolution des défauts de la machine.

Le caractère ouvert de la CN permet de considérer l'utilisation d'équipements additionnels pouvant modifier le comportement de la machine. Il est alors possible d'imaginer l'utilisation d'un bras robotisé par exemple pour la mise en position de la pièce (Figure 83). Ceci permet alors d'augmenter les capacités de la machine, la pièce étant orientable, il est possible d'obtenir des surfaces complexes sur une machine 3 axes nécessitant normalement une remise en position ou l'utilisation d'une machine 5 axes. Il est possible d'identifier la souplesse du robot [Qin11] [Arteaga00] et de mesurer sa position (à l'aide de lignes de référence et d'un système de *laser tracking* par exemple [Aguado12]) afin de compenser en temps réel les mouvements de l'ensemble pièce/montage durant l'usinage.

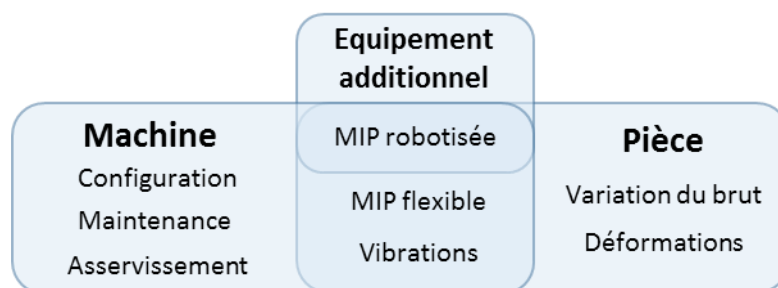


Figure 83 - Vue d'ensemble des possibilités de XPAIM

Nous allons présenter une illustration concrète permettant de prendre en compte et de capitaliser ces informations au sein de LinuxCNC.

4.1.2. Illustration de configuration avancée au sein de LinuxCNC

Dans le chapitre 3 de ce manuscrit, nous avons présenté les modifications nécessaires à la configuration HAL la plateforme XPAIM, permettant de compenser en temps réel des perturbations simulées à l'aide d'un montage expérimental. Les développements effectués sont adaptables afin d'obtenir une prise en compte avancée des informations observées durant le processus de fabrication. La Figure 84 illustre la modification du comportement de l'axe X de la plateforme XPAIM.

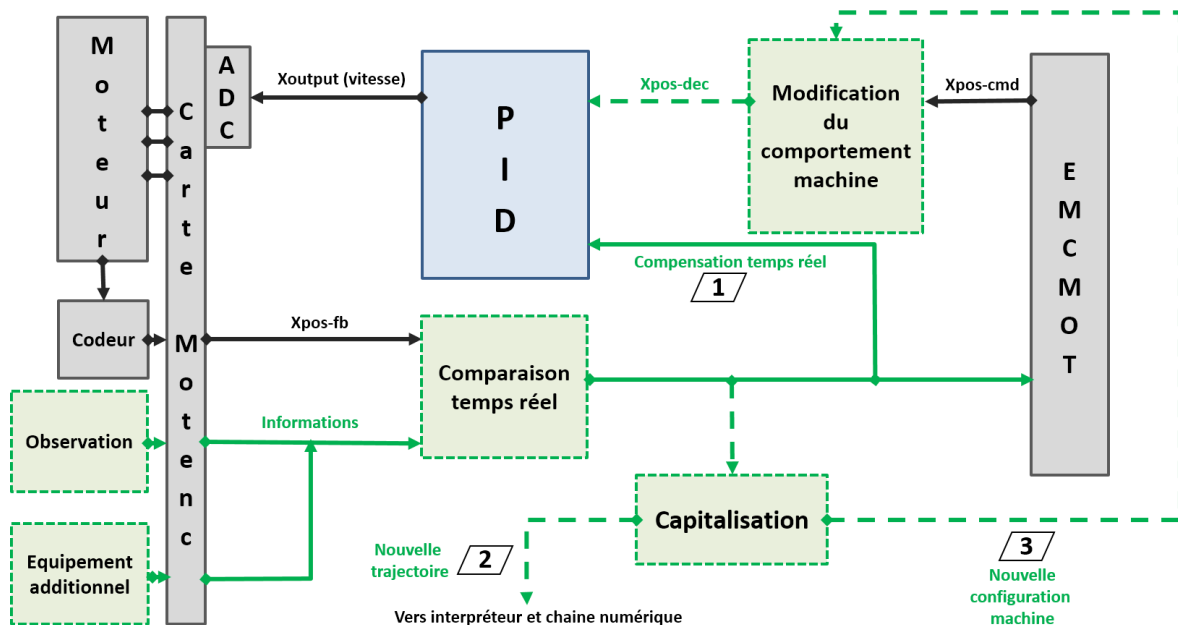


Figure 84 – Configuration HAL avancée de l'axe X de la plateforme XPAIM

Tout d'abord, le bloc « comparaison temps réel » est en lieu et place du bloc « somme » utilisé dans le chapitre 3. Ce bloc fait appel à des fonctions avancées de prise en compte des informations observées. Dans la réalité, le lien perturbation/action de compensation ne sera pas aussi simple que dans le cas de la démonstration qui nécessitait seulement une somme. Ce bloc peut exploiter les informations de l'ensemble des axes de la machine et modifier le retour de position du codeur (Xpos-fb) de manière optimisée. Ceci permet de compenser de manière directe (lien 1 vers PID) les perturbations sur la pièce comme déjà démontré. Les informations issues du bloc sont aussi exploitées par un bloc de capitalisation. Les données correspondant au deuxième niveau d'information (présenté dans la section précédente) permettent de définir une nouvelle trajectoire qui est injecté dans la chaîne numérique de la pièce en cours de production. D'autres données liées au troisième niveau d'information sont exploitées pour modifier le comportement de la machine à plus long terme (réglage) et ainsi rectifier la position commandée par l'unité de commande de mouvement de LinuxCNC (EMCMOT).

Les développements effectués rendent alors les méthodes de génération de trajectoires plus flexibles. Nous avons vu qu'il est difficile d'obtenir des pièces toujours conformes avec une trajectoire ne prenant pas en compte les conditions réelles de réalisation d'une pièce. A l'aide des méthodes prédictives, les trajectoires sont devenues spécifiques en prenant en compte le triptyque machine/process/pièce. La vision développée dans nos travaux permet de privilégier une génération de trajectoires plus générique qui s'adapte ensuite au

système de production. Il est alors plus facilement envisageable d'exploiter les avantages d'une usine globalisée.

Les possibilités sont donc nombreuses mais dans l'ensemble, cela demande une maîtrise de la gestion des données. Afin de capitaliser les données et d'avoir des informations exploitables pour l'ensemble de la production, il est nécessaire de respecter une certaine interopérabilité et d'utiliser des langages standardisés, d'homogénéiser le format des échanges des données, ce qui est possible avec des systèmes ouverts utilisant le langage STEP-NC.

Finalement, les développements effectués respectent bien notre volonté d'être en adéquation avec le concept d'usine dite du futur, car ils permettent une meilleure flexibilité de la programmation et mettent en avant l'atout d'une bonne gestion des données. Nous allons maintenant présenter un cadre concret d'intégration de nos travaux en partant d'une description de l'usine du futur dans sa globalité.

4.2. Usine du futur : Concepts généraux et applications

4.2.1. Usine du futur dans sa globalité

L'industrie manufacturière fait face à des exigences de production de plus en plus intenses. Le marché est devenu très concurrentiel et exige toujours plus de réduction des délais, des coûts mais aussi plus de flexibilité, car les produits sont plus variés et ont un temps de cycle de vie plus court. En plus d'être performante, cette industrie doit aussi être respectueuse de l'environnement (en intégrant des contraintes de durabilité). Dans ce contexte, cette dernière décennie a vu de nombreux concepts et paradigmes émerger concernant l'usine de demain, et ceci sous des appellations multiples : Usine numérique [Bracht05] [Kuhn06], Cloud Manufacturing [Xu12] [Wu13], Ubiquitous Manufacturing (en lien avec l'informatique ubiquitaire) [Suh08] [Yoon12], Usine intelligente [Radziwon14] ou encore l'Industrie 4.0 [Kagermann13].

Bien que nombreux, ces concepts se rejoignent sur une idée principale : la gestion collaborative des informations.

Pour cela, il est nécessaire d'estomper les frontières entre chaque élément et acteur de l'industrie, d'homogénéiser le format d'échange de données et d'exploiter toutes les informations pour optimiser la production. La Figure 85 permet de visualiser l'ensemble des informations liées à la production d'une pièce.

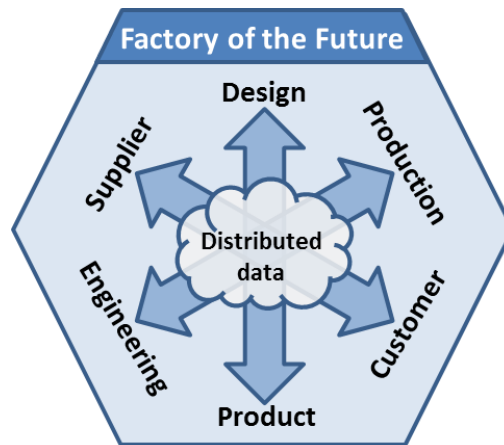


Figure 85 - Ensemble des informations liés à la fabrication d'un produit [Simoes15]

Sur cette figure, nous retrouvons trois axes principaux [Fortin08] qui reprennent les fondements de la gestion du cycle de vie des produits (Product Life Cycle Management – PLM) [Stark04]. Chaque axe est focalisé sur un aspect de la production et fait appel à différents outils et méthodes que nous présentons succinctement ci-dessous :

- Axe Fournisseurs-Client : cet axe traite de la logistique et de la gestion globale et externe. Il fait intervenir des outils de gestion de l'approvisionnement (Supply Chain Management – SCM) [Chauhan03] et de gestion de la relation client (Customer Relationship Management – CRM) [Brown06].
- Axe Ingénierie-Production : cet axe traite les procédures de décision et la gestion globale interne de l'entreprise. Il est basé sur des outils tels que ceux permettant de définir le produit lors de son cycle de vie (Product Data Management – PDM) et ceux permettant la gestion des ressources de production (Enterprise Resource Planning – ERP et Manufacturing Ressources Planning – MRP) [Umble03].
- Axe Concept-Produit : cet axe est lié à toutes les caractéristiques intrinsèques du produits. Tous les outils qui permettent de passer de l'idée à la pièce réalisée peuvent être réunis dans le concept général CIM (Computer Integrated Manufacturing) [Mikell07] qui intègre entre autre des outils tels que ceux liés à l'innovation (Computer Aided Innocation – CAI) [Leon09] et à la planification de process (Computer Aided Process Planning – CAPP).

Les frontières au sein de ces axes tendent à disparaître et les liens entre chaque acteur deviennent plus étroits. Par exemple dans le cas de l'axe Concept-Produit, nous pouvons citer les méthodes de Design For Manufacturing (DFM) qui visent à éviter les problèmes de fabrication dès l'étape de conception [Gupta97] en rapprochant le bureau d'étude et le bureau des méthodes. Cependant, l'exploitation concrète des données de plusieurs axes

reste encore rare. Nous pouvons citer une application des travaux de [Wang13] qui permet à partir d'une requête client de calculer le cout et les délais d'obtention d'un produit en exploitant toutes les informations, stockées dans un cloud, liées au produit, aux fournisseurs et aux moyens de productions.

Dans le cadre de nos travaux, nous nous intéressons plus particulièrement à l'axe Concept-Produit, pour lequel une interaction doit exister avec les deux autres axes. Ceci est justement le point de vue du système XMIS (eXtended Manufacturing Integrated System) [Laguionie10] comme illustré sur la Figure 86.

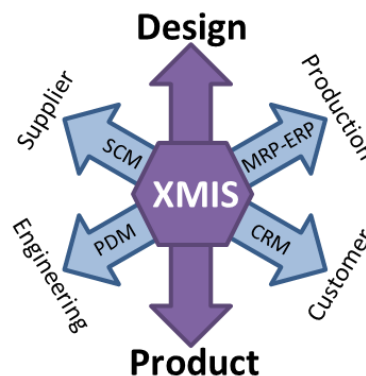


Figure 86 - XMIS au centre des acteurs de l'usine du futur [Simoes15]

Nous décrirons donc plus particulièrement ce système, en se focalisant tout d'abord sur son concept puis sur ses parties opératives.

4.2.2. XMIS : Pour une meilleure gestion des données au sein de l'usine du futur

Le système XMIS permet de lier un grand nombre d'informations d'une industrie à travers une unité de production de projet (Production Project Unit) comme illustré sur la Figure 87.

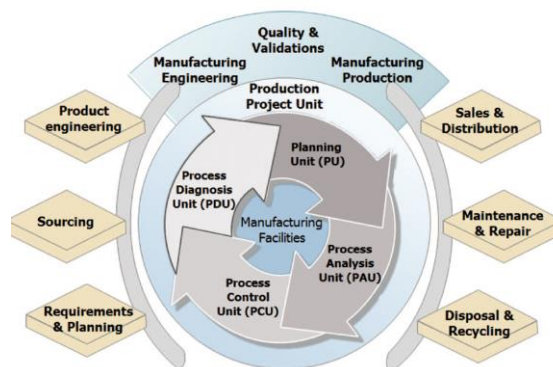


Figure 87 - Echange de données au sein de XMIS [Laguionie11]

Au sein de XMIS, nous retrouvons quatre principales unités définies ci-après :

- L'unité de planification (*Planning Unit* – PU) dans laquelle le choix du process est effectué en adéquation avec les entités à réaliser.
- L'unité d'analyse de processus (*Process Analysis Unit* – PAU) dans laquelle les paramètres de fabrications sont optimisés en lien avec des outils de simulations (lien avec les méthodes prédictives).
- L'unité de contrôle de processus (*Process Control Unit* – PCU) qui permet d'adapter en temps réel la trajectoire et les paramètres d'usinages (lien avec les méthodes correctives et usinage adaptatif).
- L'unité de diagnostic du processus (*Process Diagnosis Unit* - PDU) qui exécute toutes les tâches liées à l'inspection et à la validation de conformité.

Ces différentes unités échangent des informations à l'aide de standards de communication (Open Communication – OC, Informations and Communication Technologie – ICT) et plus particulièrement le langage STEP-NC.

Le langage STEP-NC permet d'avoir un échange multidirectionnel de données entre chaque unité du PPU mais aussi entre diverses PPU d'une même usine, afin d'avoir un grand réseau de données de fabrication (Manufacturing Information Pipeline) et permet d'exploiter le concept de « Build Anywhere » (Usine globalisée qui permet de « fabriquer partout »). Une description complète des unités XMIS peut être faite au sein de l'usine principale du fabricant, et cette dernière peut communiquer avec des combinaisons de modules XMIS d'autres contributeurs selon leur compétence comme illustré sur la Figure 88 (adapté de [Lagionie10]).

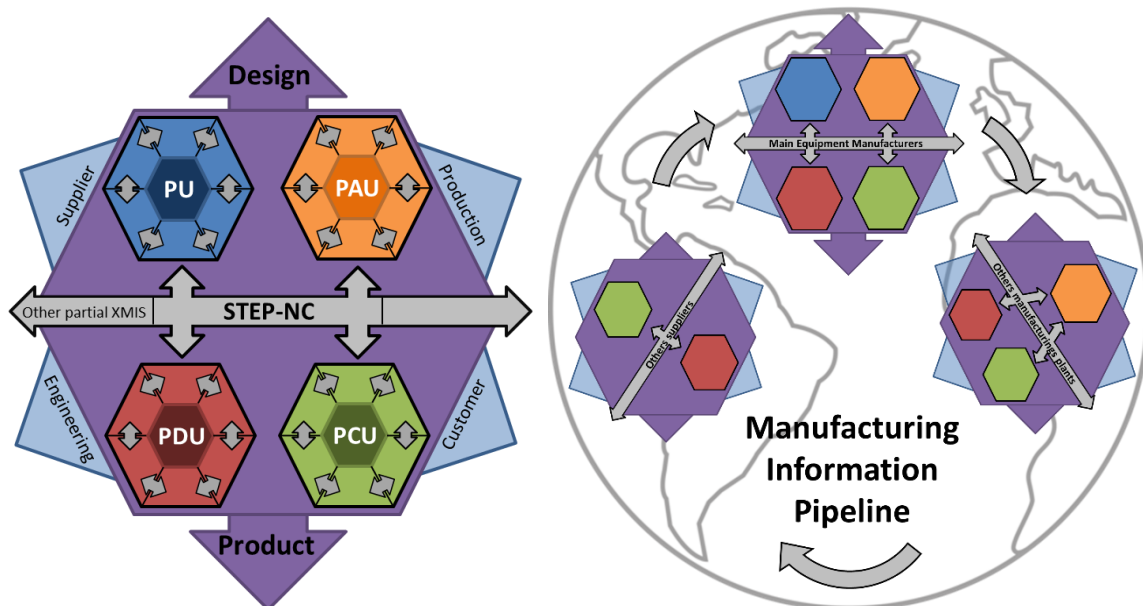


Figure 88 – Illustration d'un concept d'usine globalisée basé sur XMIS : « Build Anywhere » [Simoes15]

Ce concept d'échange d'informations contribue à une application innovante dite d'e-manufacturing dans un contexte d'usine étendue [Wang04] tels que :

- Supervision d'usinage et gestion de machine de production [Hao05]
- Optimisation par compétences déportées [Hascoet05]
- Maintenance et contrôle à distance [Alvares06]
- E-marketing, e-expertise, e-learning [Rauch07]

4.2.3. Vers une CN étendue aux nombreuses applications

Après avoir décrit le concept de XMIS, nous nous intéressons à la description opérative de la chaîne numérique représentée sur la Figure 89 et faisant apparaître deux outils principaux : la CAO étendue (eXtended Computer Aided Design – XCAD) et la CN étendue (eXtended Computer Numerical Control – XCNC).

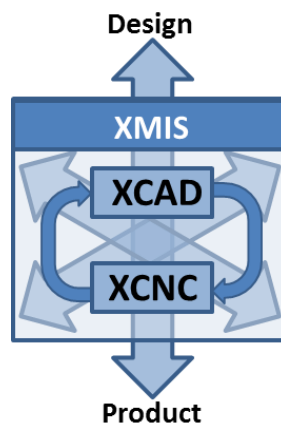


Figure 89 - Chaîne numérique de fabrication au sein de XMIS [Simoes15]

Brièvement, XCAD incorpore toute l'unité de planification de XMIS, à l'aide de langages standards STEP et STEP-NC. Cet outil est en lien étroit avec la CN étendue dont nous allons montrer plus en détails les différentes possibilités.

Le cœur de XCNC est complètement intégré au PPU et gère toutes les couches d'applications d'analyse, de contrôle et de diagnostic (PAU, PCU et PDU). Comparée à une simple CN, qui peut principalement interpréter et exécuter une trajectoire, la CN étendue possède un grand nombre de fonctionnalité comme illustré sur la Figure 90.

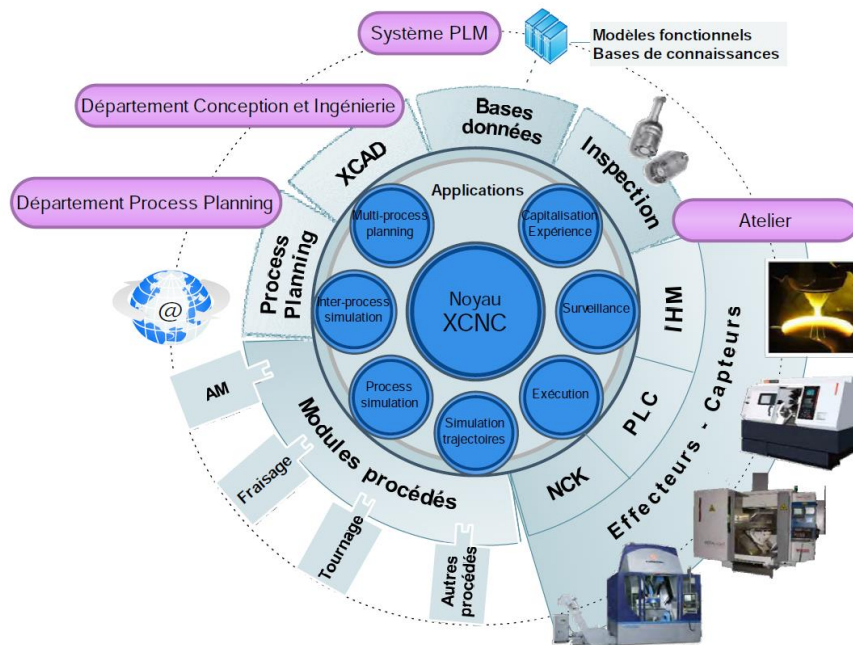


Figure 90 - Vue d'ensemble des applications de la CN étendue [Laguionie10]

En plus des composants habituels (NCK, PLC et IHM), la CN étendue est en lien avec les outils d'inspection, de planification de process et a accès à l'ensemble des données de l'usine, lui permettant une optimisation de la trajectoire. Nous retrouvons donc le concept de « cloud manufacturing » où toutes les données doivent être accessibles pour optimiser la production. La partie de prise en compte des modules de procédés est aussi en adéquation avec deux concepts innovants :

- Système de production modulaire : Aussi appelé Usine Reconfigurable ou concept MPS (Modular Production System), c'est un concept qui met en avant la possibilité de reconfigurer un système de fabrication en fonction des exigences du produit à réaliser [Koren99] [Koren10][Friedrich14] (Figure 91).

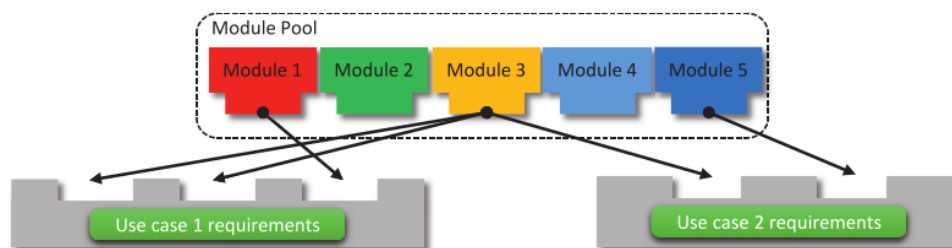


Figure 91 - Concept MPS qui combine des modules de production pour s'adapter à la pièce à produire [Friedrich14]

- Optimisation multi-process : Concept qui permet d'augmenter la productivité à l'aide du langage STEP-NC. Celui-ci donnant la possibilité d'intégrer, de superviser, de lier et rendre interopérable l'ensemble des process [Laguionie10] (Figure 92).

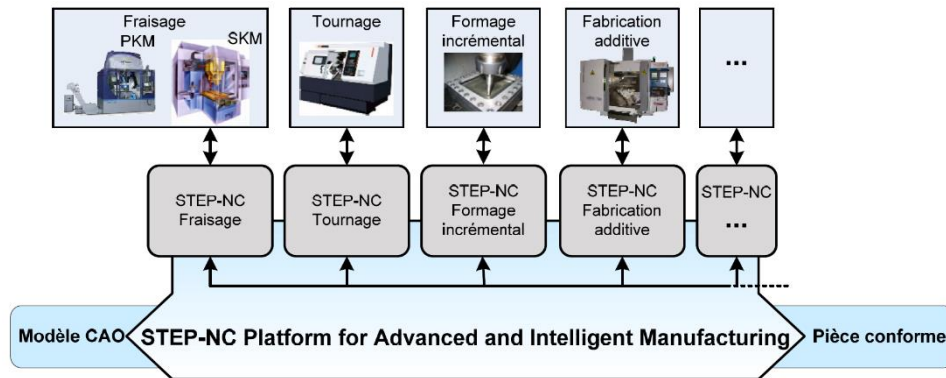


Figure 92 - Concept de fabrication multi-process à l'aide de STEP-NC [Laguionie10]

Afin de démontrer l'ensemble des possibilités des outils de la CN étendue, la plateforme SPAIM (STEP-NC Platform for Advanced and Intelligent Manufacturing) a été développée par l'IRCCyN. Une vue globale de cette plateforme est visible sur la Figure 93.

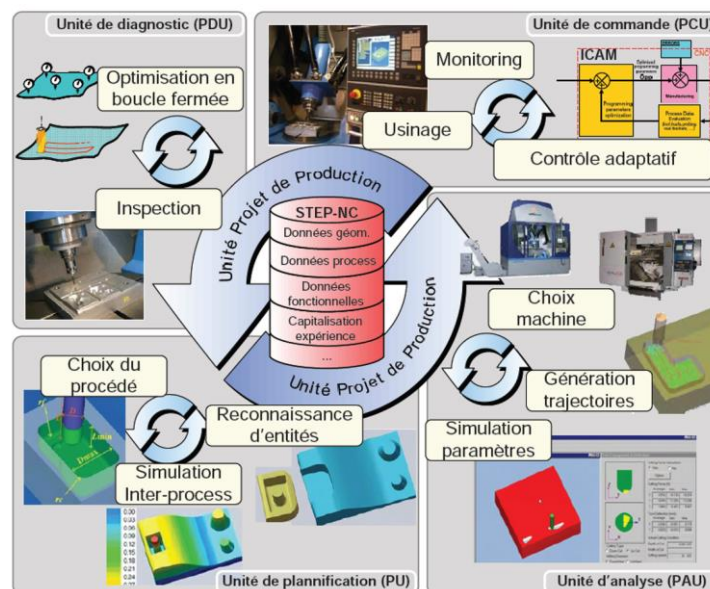


Figure 93 - Vue globale des outils de la plateforme SPAIM

Finalement, les travaux présentés dans ce manuscrit ne sont pas cantonnés seulement à l'outil de contrôle adaptatif de l'unité de pilotage, mais ont un lien avec l'ensemble des outils. Il est possible d'utiliser les informations observées durant la fabrication d'une pièce pour optimiser les paramètres au sein de l'unité d'analyse du process par exemple. Il y a aussi naturellement un lien très proche avec les outils d'inspection et surveillance. Le fait de s'adapter aux variations du brut peut permettre une meilleure exploitation de la fabrication inter-process (en modifiant une trajectoire d'usinage pour s'adapter à un dépôt fait par ajout de matière par exemple).

4.3. Conclusion

Dans ce chapitre, nous avons montré que nos travaux offrent de nombreuses perspectives permettant d'optimiser la production d'une pièce. Nous avons identifié différents niveaux d'exploitation d'informations allant de l'adaptation du parcours d'usinage d'une pièce aux réglages de la machine au sein de la production. Nous avons démontré dans nos travaux que le caractère ouvert de la CN permet, en ayant accès aux informations en temps réel, de compenser directement les perturbations durant la réalisation d'une pièce. Ce chapitre illustre aussi que des informations sont capitalisables et permettent une génération de trajectoires plus flexible. Ces perspectives nécessitent une gestion collaborative des données au sein de l'entreprise manufacturière et sont en adéquation avec les courants de pensées de l'usine du futur. Nous avons donc synthétisé les différents concepts de cette usine afin d'identifier un cadre d'intégration concret de nos travaux. Nous avons tout d'abord présenté qu'il est nécessaire d'estomper les frontières entre chaque facette de l'industrie (fournisseurs, concept, production, client, produit et ingénierie). Ensuite, nous nous sommes focalisé sur l'axe entre le concept et le produit fini, autrement dit la chaîne de fabrication, afin d'en dégager le concept d'une CN étendue. C'est au sein de celle-ci que nos travaux ont idéalement leur place et ont un impact sur l'ensemble des outils de l'unité projet de production.

Conclusion et perspectives

Conclusion et perspectives

Synthèse

Les travaux de thèse présentés dans ce manuscrit s'intéressent à la programmation des MOCN au sein d'une industrie manufacturière compétitive. L'objectif identifié est d'explorer et mettre en œuvre une nouvelle méthode de génération de trajectoires en ligne afin d'améliorer la productivité au sein de l'usine du futur. Dans ce but, les différentes contributions de ce manuscrit portent sur :

- le développement d'une méthode de génération de trajectoires permettant de compléter les méthodes de programmation avancée actuelles,
- la mise en œuvre d'une plateforme intelligente de fabrication, basée sur une Commande Numérique Ouverte en alternative aux CNs propriétaires,
- les perspectives d'évolution des méthodes de programmation au sein de l'usine du futur.

Ces différentes contributions donnent un rôle décisionnel central à la commande numérique et sont développées dans l'optique d'être applicable à tout type de procédé.

Dans un premier temps, nous avons établi un état de l'art des évolutions de la chaîne numérique de fabrication. Le premier chapitre de ce manuscrit a présenté les différentes dynamiques de recherche permettant de répondre aux problématiques des modèles courants de la chaîne numérique. En effet, le contexte industriel actuel est caractérisé par le monopole d'utilisation d'une chaîne numérique cloisonnée et hétérogène CAO/FAO/CN qui laisse peu de possibilités d'optimisation de fabrication. En lien avec cette problématique, nous avons présenté les attraits de l'utilisation d'un langage de programmation basé entités, plus particulièrement le STEP-NC, ainsi que l'apport des méthodes de programmation avancées, plus particulièrement les méthodes prédictives et correctives. Il apparaît que l'exploitation de ces développements est encore limitée. Il existe un manque de prise en compte de certaines informations du process au cours de la réalisation d'une pièce. Et quand ces informations sont accessibles en temps réel, il est encore difficile de les exploiter au sein de la commande numérique. Un besoin de compenser les perturbations en temps réel est alors identifié et nous amène à définir le principe de fabrication adaptative. Dans le premier chapitre, nous avons montré que cette approche est encore limitée par des obstacles au sein des CN propriétaires. A la suite d'une

synthèse de ces limitations, nous avons établi une feuille de route permettant de répondre à la problématique de génération de trajectoires en ligne permettant de s'adapter aux conditions réelles de réalisation d'une pièce au sein de la commande numérique.

La première étape de cette feuille de route consiste à définir une alternative à l'utilisation d'une CN propriétaire. La plateforme avancée XPAIM est présentée au sein du second chapitre de ce manuscrit. Nous avons fait le choix d'y intégrer une commande numérique ouverte. Après avoir détaillé les différents critères à respecter d'une CNO, nous avons effectué une synthèse et une comparaison des principaux projets académiques et industriels existants. Ceci nous a permis d'identifier le projet LinuxCNC comme étant capable de servir de base CNO à nos développements, car celle-ci bénéficie d'une description complète et d'un intérêt actuel de la communauté de contributeurs. Nous avons détaillé son intégration au sein de XPAIM et effectué un ensemble d'expérimentations démontrant son potentiel. L'ensemble des contributions de ce chapitre répond aux trois objectifs technologiques :

- Intégrer une CN supplémentaire de nouvelle génération à une MOCN, tout en conservant la CN d'origine.
- Intégrer une CNO, dont les critères de portabilité, d'interopérabilité et d'extensibilité sont nécessaires à nos développements.
- Montrer l'entière fonctionnalité de cette CNO et illustrer un exemple de fonction avancée.

La résolution de ces objectifs technologiques permet de répondre à l'objectif scientifique de développement d'une nouvelle méthode de programmation avancée qui est abordée dans le troisième chapitre. Dans celui-ci, nous avons défini une approche de génération de trajectoires en ligne générique exploitable sur la plateforme XPAIM. Ce concept générique est appliqué dans le cas d'un procédé d'usinage avec perturbations. Dans ce cadre, une démarche expérimentale a été entreprise afin de montrer l'apport d'une compensation en ligne de perturbations. Ceci a permis par ailleurs de démontrer l'intérêt de l'utilisation d'une architecture ouverte permettant de modifier le comportement de la machine. La commande numérique a alors un rôle plus décisionnel, car avec un programme générique défini en entrée, la trajectoire devient spécifique en s'adaptant à la géométrie réelle de la pièce.

Enfin, dans le dernier chapitre de ce manuscrit, nous avons montré que nos travaux offrent de nombreuses possibilités permettant d'optimiser la production d'une pièce. Nous avons identifié différents niveaux d'exploitation d'informations qui sont capitalisables et permettent une génération de trajectoires plus flexible. Cette aspect de capitalisation

nécessite une gestion collaborative des données au sein de l'entreprise manufacturière et sont en adéquation avec les courants de pensées de l'usine du futur. Les différents concepts de cette usine sont alors synthétisés. Nous nous focalisons particulièrement sur ceux liés à l'axe allant du concept au produit fini, autrement dit la chaîne de fabrication. Ceci afin d'en dégager le concept d'une CN étendue, au sein de laquelle nos travaux ont idéalement leur place et ont un impact sur l'ensemble des outils de production.

Perspectives

De manière générale, les apports de ces travaux s'attachent à montrer l'intérêt d'une approche de programmation avancée rendant la production plus flexible. En rendant possible l'adaptation de trajectoires en temps réel durant un process de fabrication, nous rendons la commande numérique, et donc la machine, plus intelligente. Dans l'ensemble des développements, l'apport du langage STEP-NC est mis en avant, celui-ci permettant entre autre d'estomper les frontières au sein de l'usine manufacturière. Ce couple langage/méthode de programmation rentre parfaitement dans la dynamique de recherche et d'innovation consacrée à rendre l'usine intelligente et globalisée.

En lien direct avec nos travaux, différentes perspectives sont possibles. Nous pouvons tout d'abord identifier celles à court/moyen terme :

- Concernant la démonstration de compensation de perturbation effectuée : les perspectives à moyen terme à envisager sont de corréliser les informations observées de perturbation avec la géométrie de la pièce et les efforts durant le process de fabrication. Ceci permettrait d'adapter la compensation en fonction du déplacement de l'outil, en distinguant les perturbations effectuées hors matière et celles effectuées dans une zone transformée par la suite, de celles affectant la conformité de la pièce. Pour cette optimisation en fonction des efforts, nous avons un lien direct avec la méthode ICAM. Pour ce qui est de la prise en compte de la géométrie de la pièce, le lien est fait avec la programmation basé entité.
- Concernant les modifications de configuration au sein de XPAIM : Le caractère ouvert de LinuxCNC permet d'intégrer des composants plus complexes de prises en compte des informations observées. Il est possible de mettre en place une mise en œuvre de réglage de la machine ou encore d'extraire les données observées en temps réel pour réintégrer la trajectoire au sein de l'interpréteur et au sein de la chaîne numérique.

Concernant les perspectives à plus long terme, en adéquation avec le contexte industriel, un effort a été donné à ce que les travaux effectués ne soient pas seulement conceptuels

et à ce que leur intégration soit possible au sein des industries manufacturières. Il est montré que la CNO est parfaitement intégrable et peut fonctionner avec d'autres CN au sein d'une même machine de type industriel. Nous avons vu que la plateforme résultante est entièrement fonctionnelle et performante. Il est important de poursuivre cet effort pour convaincre les industriels de faire évoluer la programmation de leur machine. En particulier sur l'utilisation d'une programmation basée entités. Il est difficile d'imaginer un avenir à long terme pour la programmation actuelle code G. L'utilisation d'une CN ouverte est une nouvelle étape pour faciliter l'intégration du langage STEP-NC. En suivant le concept de CN étendue présenté dans ces travaux, il est possible de mettre en œuvre toutes les possibilités de contrôle intelligent et d'adaptation de production en temps réel en effectuant une intégration complète du langage STEP-NC au sein de XPAIM.

Finalement, le recours à une commande numérique ayant une architecture ouverte fait sauter des verrous technologiques importants et offre un terrain propice à une réelle évolution de la chaîne numérique de fabrication, et donc de l'industrie manufacturière dans sa globalité. Les travaux de thèse développés dans ce manuscrit offrent à la fois une vision d'une programmation avancée générique de fabrication adaptative et la démonstration d'un cas concret de compensation de perturbations en ligne sur une plateforme accessible à toute la communauté scientifique. Il convient maintenant que l'ensemble des acteurs rentre dans une dynamique de mise en œuvre concrète des concepts d'usine du futur, pour un système de production performant, souple et durable.

Références bibliographiques

- [Aguado12] S. Aguado, D. Samper, J. Santolaria, J. J. Aguilar, Identification strategy of error parameter in volumetric error compensation of machine tool based on laser tracker measurements, *International Journal of Machine Tools and Manufacture*, 53 (1) (2012) 160-169.
- [Achelker14] Achelker, C., Rao, N. S., Rajendra, R. et Krishaniah, A. (2014). Performance evaluation of machine tool probe for in-process inspection of 2D and 3D geometries. *Procedia Technology*, 14:244–251.
- [Affouard04] Affouard A., Duc E., Lartigue C., Langeron J. M., Bourdet P., Avoiding 5-axis singularities using tool path deformation, *International Journal of Machine Tools and Manufacture*, 2004, Vol. 44 n°4, pp. 415-425
- [Albus91] Albus, James S; et al; NIST Support to the Next Generation Controller Program: 1991 Final Technical Report; NISTIR 4888; National Institute of Standards and Technology, Gaithersburg, MD; July 1992
- [Altintas93] Altintas, Y., Newell, N. and Ito, M., A hierarchical open architecture multi-processor CNC system for motion and process control. *ASME PED-64 Manufacturing Science and Engineering*. (1993) 195-205.
- [Alvares06] Alvares A. J., Ferreira J. C. E., WebTurning: Teleoperation of a CNC turning center through the Internet, *Journal of Materials Processing Technology*, 2006, Vol. 179 n°1-3, pp. 251-259.
- [Ara15] Project Ara, Google, <http://www.projectara.com/ara-developers-conference/> Date de consultation : 06/05/15.
- [Arteaga00] Arteaga, M. a., & Siciliano, B. (2000). On tracking control of flexible robot arms. *IEEE Transactions on Automatic Control*, 45(3), 520–527.
- [Asato02] Asato, O. & Kato, E., 2002. Analysis of open CNC architecture for machine tools. *Journal of the Brazilian Society of Mechanical Sciences*, vol.24 no.3.
- [Beudaert13] Beudaert. (2013). *Commande numérique ouverte: interpolation optimisée pour l'usinage 5 axes grande vitesse des surfaces complexes*, Thèse de doctorat, École Normale Supérieure de Cachan.
- [Bracht05] Bracht, U. and Masurat, T. (2005). "The Digital Factory between vision and reality." *Computers in Industry*, Vol 56 (4), pp 325-333.
- [Brown06] Brown, S. (2006). "CRM Customer Relationship Management : la gestion de la relation client ", Village Mondial.

- [Chauhan03] Chauhan, S. S., (2003). " Chaînes d'approvisionnement: approches stratégique et tactique." Université de Metz.
- [Chen, 2012] Chen, J. (2012). Study on machining distortion of residual stress release. *Advanced Materials Research*, 426:143–146.
- [Cincinnati] <http://www.sterlingmachinery.com/db-files/cincinnati%20sabre%20-%20750%20cnc%20vertical%20machining%20center%20brochure.pdf> Dernière consultation : 20/07/2015
- [Dessein97] Dessein, G. (1997). Qualification et optimisation de la précision d'une machine-outil à commande numérique. Thèse de doctorat, Université Paul Sabatier Toulouse 3, Toulouse.
- [Duc98] Duc, E. (1998). Usinage de formes gauche : contribution à l'amélioration de la qualité des trajectoires d'usinage. Thèse de doctorat, École Normale Supérieure de Cachan.
- [Duc02] Duc E., Lartigue C., Bourdet P., Utilisation de l'interpolation polynomiale pour l'UGV - Cas de l'usinage à 5-axes, Assises Françaises Machine Outils & Usinage Grande Vitesse, Lille, 13-14 Mars 2002, pp. 287-296.
- [Dugas02] Dugas, A. (2002). CFAO et UGV - Simulation d'usinage de formes complexes. Thèse de doctorat, École Centrale de Nantes - Université de Nantes.
- [Dugas03] Dugas A., Lee J. J., Terrier M., Hascoet J. Y., Development of a machining simulator considering machine behaviour, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2003, Vol. 217 n°9, pp. 1333-1339
- [FoFdration] Foundation for the sustainable Factory of the Future <http://www.fofdation-project.eu/> Date de consultation : 13/11/13.
- [FoFdration11] Hamilton, K., Hascoët, J.-Y., Rauch, M., D3.1.1 Description of the SMC open controller, Deliverable FoFdration, Novembre 2011.
- [Fortin08] Fortin C., (2008) "Le PLM et le MPM", Séminaire interne, Lurpa-ENS de Cachan, mai 2008.
- [FP7] http://ec.europa.eu/research/fp7/index_en.cfm Date de dernière consultation : septembre 2015
- [Friedrich14] Scheifele, S., Friedrich, J., Lechler, A., & Verl, A. (2014). Flexible, Self-configuring Control System for a Modular Production System. *Procedia Technology*, 15, 398–405.
- [Fujita96] Fujita, S., Yoshida, T., OSE: Open System Environment for Controller, 7th International Machine Tool Engineers Conference. (1996) 234-243.

- [Gupta97] Gupta, S.K., DAS, D., REGLI, W.C. et al. Automated manufacturability analysis: A survey. Research in Engineering Design, 1997, vol. 9/3, p. 168-190.
- [H2020] <http://www.horizon2020.gouv.fr/> Date de dernière consultation : septembre 2015
- [Hal Manual] HAL User Manual, Available from: http://www.linuxcnc.org/docs/HAL_User_Manual.pdf Date de consultation : 05/02/14
- [Hao05] Hao Q., Shen W., Wang L., Towards a cooperative distributed manufacturing management framework, Computers in Industry, 2005, Vol. 56 n°1, pp. 71-84
- [Hardwick06] Hardwick, M. and Loffredo, D. (2006). "Lessons learned implementing STEP-NC AP-238." International Journal of Computer Integrated Manufacturing, Vol 19 pp 523-532.
- [Hascoet00] Hascoet J. Y., Dugas A., Lee J. J., High speed milling : Simulation considering machine and controller's limits, International CIRP Seminar on Improving Machine Tool Performance, La Baule (France), Juillet, 2000, CD-Rom
- [Hascoet05] Hascoet J. Y., Rauch M., Remote Control of a parallel kinematics machine tool in an e-manufacturing context, Journal of Machine Engineering, 2005, Vol. 5 n°1-2, pp. 95-104.
- [Hascoet06] Hascoet J. Y., Rauch M., A new approach of the tool path generation in manufacturing operations using CNC data, International Conference on High Speed milling, Suzhou (China), 11-13 mai 2006c, pp. 101-113
- [Hudson82] Hudson C. A., 1982, Computers in Manufacturing, Sciences, 215, p. 818-825.
- [IEEE95] IEEE 1003.0-1995 (IEEE Guide to the POSIX® Open System Environment (OSE))
- [Kagermann13] Kagermann, H., Wahlster, W., Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final Report of the Industrie 4.0 Working Group.
- [Koren83] Koren Y., Rasmussen F., 1983, Principal Developments in the Adaptive Control of Machine Tools, ASME, Volume 105, p. 107-112.
- [Koren88] Koren, Y., 1988. Adaptive control systems for machining. American Control Conference, 1988, IT1A Series.
- [Koren99] Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy AG et al. Reconfigurable manufacturing systems. CIRP Annals 1999; 48(2):6–12.
- [Koren10] Koren Y, Shpitalni M. Design of reconfigurable manufacturing systems. Journal of Manufacturing Systems 2010;29(4):130-141.

- [Kramer06] Kramer T. R., Proctor F., Xu X. W., Michaloski J. L., Run-Time interpretation of STEP-NC: implementation and performance, *International Journal of Computer Integrated Manufacturing*, 2006, Vol. 19 n°6, pp. 495-507
- [Kuhn06] Kuhn, W. (2006). Digital factory: simulation enhancing the product and production engineering process, *Proceedings of the 38th conference on Winter simulation*, Monterey, California, Winter Simulation Conference, 2006.
- [Kumar10] Kumar, V. A. and Sunil, R. (2010). "A review of machining feature recognition methodologies." *International Journal of Computer Integrated Manufacturing*, Vol 23 (4), pp 353-368.
- [Laguionie10] Laguionie, R., 2010. Intégration de la programmation basée entités pour la fabrication sur machines à commandes numériques dans un contexte multi-process, Thèse de doctorat, École Centrale de Nantes.
- [Laguionie10] Laguionie, R., 2010. Intégration de la programmation basée entités pour la fabrication sur machines à commandes numériques dans un contexte multi-process, Thèse de doctorat, École Centrale de Nantes.
- [Laguionie11] Laguionie, R., Rauch, M., Hascoët, J.-Y., & Suh, S.-H. (2011). An eXtended Manufacturing Integrated System for feature-based manufacturing with STEP-NC. *International Journal of Computer Integrated Manufacturing*, 24(9), 785–799.
- [Langeron04] Langeron J. M., Duc E., Lartigue C., Bourdet P., A new format for 5-axis tool path computation, using Bspline curves, *CAD Computer Aided Design*, 2004, Vol. 36 n°12, pp. 1219-1229
- [Lartigue01] Lartigue, C., Thiebaut, F. et Maekawa, T. (2001). CNC tool path in terms of B-spline curves. *Computer-Aided Design*, 33(4):307 – 319.
- [Lartigue04] Lartigue, C., Tournier, C., Ritou, M. et Dumur, D. (2004). High-performance NC for HSM by means of polynomial trajectories. *CIRP Annals - Manufacturing Technology*, 53(1):317 – 320
- [Leon09] Leon, N. (2009). "The future of computer-aided innovation." *Computers in Industry*, Vol 60 (8), pp 539-550.
- [Li04] Li Z. Z., Zhang Z. H., Zheng L., Feedrate optimization for variant milling process based on cutting force prediction, *The International Journal of Advanced Manufacturing Technology*, 2004c, Vol. 24 n°7-8, pp. 541-552
- [LinuxCNC_Hal14] http://www.linuxcnc.org/docs/2.5/pdf/LinuxCNC_HAL_Manual.pdf, consulté le 05/02/2014

- [LinuxCNC_Kinematics_2014] <http://linuxcnc.org/docs/html/motion/kinematics.html>, consulté le 21/05/2014
- [Liu89] Liu L., Sinha N., Elbestawi M., 1989, Adaptive control for geometric tracking in turning, *Computers in Industry*, 11, p. 147-159.
- [Marietta94] Marietta, M., 1994, Next generation controller (NGC) specifications for an open system architecture (SOSAS) revision 2.0. Technical report, National Center for Manufacturing Sciences.
- [Mikell07] Mikell, P. G. (2007). "Automation, Production Systems, and Computer-Integrated Manufacturing", Prentice Hall Press.
- [Miles98] Miles, P. (1998). Open architecture: forecasting the adoption wave, *Robotics World*, Vol 16 (2), pp 23-29
- [Nacs08] Nacs, J. Comparison of three different open architecture controllers. *Proceedings of IFAC MIM (2001)* 134-138.
- [Newman07] Newman, S. T., Ali, L., Brail, A., Brecher, C., Klemm, P., Liu, R., Nassehi, A., Nguyen, V. K., Proctor, F., Rosso, R. S. U., Stroud, I., Suh, S. H., Vit, M., Wang, L. and Xu, X. W. (2007). The Evolution of CNC Technology from Automated Manufacture to Global Interoperable Manufacturing, *The 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2007)*, Toronto, Canada, 22-24 July 2007.
- [NML] The NML Programmer's Guide, Available from: <http://www.isd.mel.nist.gov/projects/rcslib/NMLcpp.html> Date de dernière modification : 17/06/09 Date de consultation : 21/05/14.
- [Pateloup, 2005] Pateloup, V. (2005). Amélioration du comportement cinématique des machines outils UGV - Application au calcul de trajets d'évidement de poches. Thèse de doctorat, Université Blaise Pascal - Institut Français de Mécanique Avancée.
- [Piegl95] Piegl L., Tiller W., 1995. *The NURBS Book*. Springer-Verlag, London, UK, UK.
- [Poulhaon13] F. Poulhaon, M. Rauch, A. Leygue, J.-Y. Hascoet, F. Chinesta, 2013, Toward a Real Time Control of Toolpath in Milling Processes, *Key Engineering Materials*, Volumes 554-557, p. 706–713.
- [Poulhaon15] Poulhaon, F. (2015). Génération en ligne de trajectoires d'usinage par une approche de réduction de modèle. Thèse de Doctorat, École Centrale de Nantes
- [Prevost11] Prevost, D. (2011). Modélisation avancée du couple Machine Outil / Commande Numérique dans un contexte UGV pour l'optimisation du calcul de trajectoires multi axes. Thèse de doctorat, École Normale Supérieure de Cachan.

- [Pritschow01] Pritschow, G., Altintas, Y. & Jovane, F., 2001. Open controller architecture—past, present and future. *CIRP Annals-Manufacturing Technology*, 50(2001) 463–470
- [Proctor93] Proctor, F. M., and Michaloski, J., "Enhanced Machine Controller Architecture Overview," NIST Internal Report 5331, December 1993. Available online at ftp://129.6.13.104/pub/NISTIR_5331.pdf
- [Qin11] Qin, J., Leonard, F., & Abba, G. (2011). Commande robuste d'un robot d'usinage flexible: analyse de la précision. 20ème Congrès Français de Mécanique, Besançon, 29 août-2 septembre 2011.
- [Radziwon14] Radziwon, A., Bilberg, A., Bogers, M., & Madsen, E. S. (2014). The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions. *Procedia Engineering*, 69, 1184–1190.
- [Rai08] Rai J. K., Xirouchakis P., Finite element method based machining simulation environment for analyzing part errors induced during milling of thin-walled components *International Journal of Machine Tools & Manufacture* 48 (2008) 629–643
- [Rauch07] Rauch M., 2007, Optimisation de la Programmation des MOCN – Application aux Machines à Structure Parallèle, Thèse de doctorat, Ecole Doctorale MTGC, Ecole Centrale de Nantes.
- [Rauch09] Rauch, M., Hascoet, J.-Y., Hamann, J.-C., & Plenel, Y. (2009). Tool path programming optimization for incremental sheet forming applications. *Computer-Aided Design*, 41(12), 877–885.
- [Rauch09] Rauch, M., Laguionie, R., & Hascoet, J. (2009). STEP-NC, Advanced CNC programming CAD/CAM/CNC data interoperability, *Journal of Machine Engineering*, 2009, 9 (4), pp.26-37.
- [Rauch14] Rauch, M., Hascoët, J., Simoes, V., & Hamilton, K. (2014). Advanced programming of machine tools: interests of an open CNC controller within a STEP–NC environment. *International Journal of Machining and Machinability of Materials*, 15, 2–17.
- [Seo98] Seo, T. I. (1998). Intégration des effets de déformation d'outil en Génération de trajectoires d'usinage. Thèse de doctorat, Université de Nantes/Ecole Centrale de Nantes.
- [Shackleford00] Shackleford, W.P., Proctor, F.M., Michaloski J.L., 2000, The Neutral Message Language: A Model and Method for Message Passing in Heterogeneous Environments. *Proceedings of the World Automation Conference*, Maui, Hawaii, June 11-16, 2000

- [Siemens04] Siemens AG (2004). Fabrication de pièces complexes en fraisage. Édition 04/2004.
- [Siemens09] Siemens (2009). Sinumerik - 5 axis machining, DocOrderNo. 6FC5095-0AB10-0BP1.
- [Simoies14] Simoes V., Rauch M., Hascoët J.-Y. Commande Numerique Ouverte : Génération de Trajectoires "On-Line" dans un environnement de Programmation Avancée Step-NC. MUGV 2014, Oct 2014, Clermont Ferrand, France.
- [Simoies15] Simoes, V., Rauch M., Hascoet, J.-Y. (2015 - En soumission) Extended CNC in factory of the future: Interests and applications of online/real-time trajectory modification.
- [Simpson11] Simpson, B. et Dicken, P. J. (2011). Integration of machining and inspection in aerospace manufacturing. In IOP Conference Series : Materials Science and Engineering, volume 26. IOP Publishing.
- [Stark04] Stark, J. (2004). "Product Lifecycle Management: Paradigm for 21st century Product Realisation", Springer.
- [Staroveški09] Staroveški, T., Brezak, D., Udiljak, T., Majetic, D. Implementation of a Linux-based CNC open control system, 12th International scientific conference on production engineering (2009), Croatian Association of Production Engineering.
- [Suh02] Suh, S.-H., Cho, J.-H. and Hong, H.-D. (2002a). On the architecture of intelligent STEP-compliant CNC. International Journal of Computer Integrated Manufacturing, Vol 15 (2), pp 168-177.
- [Suh03] Suh, S. H., Lee, B. E., Chung, D. H., & Cheon, S. U. (2003). Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. Computer-Aided Design, 35(12), 1069–1083. [http://doi.org/10.1016/S0010-4485\(02\)00179-3](http://doi.org/10.1016/S0010-4485(02)00179-3)
- [Suh08] Suh, S. H., Shin, S. J., Yoon, J. S. and Um, J. M. (2008b). "UbiDM: A new paradigm for product design and manufacturing via ubiquitous computing technology." International Journal of Computer Integrated Manufacturing, Vol 21 (5), pp 540 - 549.
- [Tapie07] Tapie, L. (2007). Décomposition topologiques des outillages de forge pour la génération de trajectoires UGV. Thèse de doctorat, École Normale Supérieure de Cachan.
- [Terrier05] Terrier, M. (2005). Optimisation du processus de fabrication en usinage à grande vitesse sur machines-outils à structure parallèle. Thèse de doctorat, Institut de Recherche en Communications et Cybernétique de Nantes.

- [Tichadou04] Tichadou, S. (2004). Modélisation et quantification tridimensionnelles des écarts de fabrication pour la simulation d'usinage. Thèse de doctorat, École Centrale de Nantes.
- [Umble03] Umble, E. J., Haft, R. R. and Umble, M. M. (2003). "Enterprise resource planning: Implementation procedures and critical success factors." *European Journal of Operational Research*, Vol 146 (2), pp 241-257.
- [Wang04] Wang, H. and Xu, X. (2004). A STEP-Compliant "Adaptor" for Linking CAPP with CNC, proceedings of the 34th International MATADOR Conference, Manchester, UK, 7-9 July 2004, pp 45-50.
- [Wang05] Wang, Z.J., Chen, W.Y., Zhang, Y.D., Chen, Z.T., Liu, Q., 2005. Study on the machining distortion of thin-walled part caused by redistribution of residual stress. *Chin. J. Aeronaut.* 18 (2), 175–179.
- [Wang06] Wang, H., Xu, X. W. and Tedford, J. D. (2006). "Making a process plan adaptable to CNCs." *Int. J. Comput. Appl. Technol.*, Vol 26 (1/2), pp 49-58.
- [Wang07] Ma, X., Han, Z., Wang, Y., & Fu, H. (2007). Development of a PC-based Open Architecture Software-CNC System. *Chinese Journal of Aeronautics*, 20(3), 272–281.
- [Wang13] Wang, L., Song, Y., & Gao, Q. (2009). Designing function blocks for distributed process planning and adaptive control. *Engineering Applications of Artificial Intelligence*, 22(7), 1127–1138.
- [Watanabe83] Watanabe T., Iwai S., 1983, A control system to improve the accuracy of finished surfaces in milling, *Trans. ASME Dynamic Systems, Measure and Control*, 105, p. 192-199.
- [Wu13] Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, 32(4), 564–579.
- [Xu05] Xu, X. W., Wang, H., Mao, J., Newman, S. T., Kramer, T. R., Proctor, F. M. and Michaloski, J. L. (2005). STEP-compliant NC research: The search for intelligent CAD/CAPP/CAM/CNC integration. *International Journal of Production Research*, Vol 43 (17), pp 3703-3743.
- [Xu06] Xu X.W., Newman S.T., 2006, Making CNC machine tools more open, interoperable and intelligent—a review of the technologies, *Computers in Industry*, 57(2), p. 141-152.
- [Xu12] Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 28(1), 75–86.

- [Yamazaki97] Yamazaki, K. et al., 1997. Autonomously proficient CNC controller for high-performance machine tools based on an open architecture concept. *CIRP Annals-Manufacturing*, 46(2), pp.275–278.
- [Yang02] Yang M.-Y., Lee T.-M., 2002, Hybrid adaptive control based on the characteristics of CNC end milling, *International Journal of Machine Tools and Manufacture*, 42(4), p.489-499.
- [Yoon12] J.-S. Yoon, S.-J. Shin, and S.-H. Suh, A conceptual framework for the ubiquitous factory, *International Journal of Production Research*, vol.50, no. 8, Taylor & Francis, pp.2174–2189, 2012.
- [Zhao08] Zhao F., Xu X., Xie S., 2008, STEP-NC enabled on-line inspection in support of closed-loop machining, *Robotics and Computer-Integrated Manufacturing*, 24, p. 200-216.
- [Zhou04] Zhou, Z. D., Xie, J. M., Chen, Y. P., Chen, B., Qiu, Z. M., Wong, Y. S., Zhang, Y. F. The development of a fieldbus-based open-CNC system. *International Journal of Advanced Manufacturing Technology* 23(2004) 507–513.

Thèse de Doctorat

Vincent SIMOES

GENERATION DE TRAJECTOIRES «ON-LINE » POUR MACHINE A COMMANDE NUMERIQUE OUVERTE DANS UN ENVIRONNEMENT DE PROGRAMMATION AVANCEE STEP-NC

On-line path trajectory generation for Open Computerized Numerical Control
Machine in a STEP-NC advanced programming environment

Résumé

Les travaux présentés portent sur la programmation des machines-outils à commande numérique (MOCN) et la génération de trajectoires on-line.

La démarche mise en œuvre débute par une analyse de l'évolution de la programmation actuelle et des limitations des commandes numériques propriétaires. Cette analyse fait valoir un manque et un besoin d'optimisation de la commande par la prise en compte de données géométriques durant la mise en forme d'une pièce.

Le choix se porte alors sur l'utilisation d'une commande numérique ouverte (CNO) dont la définition et l'intégration sur une machine de type industriel sont détaillées.

Les développements effectués sur les possibilités et points d'entrées de la CNO permettent la validation et la mise en œuvre d'actions correctives en temps réel, à l'aide d'une étude expérimentale portant sur la mesure et la perturbation de la position d'une pièce en cours d'usinage. Sur cette base, nous structurons le concept d'une CN étendue, par l'utilisation du standard de programmation STEP-NC, qui permet notamment d'élargir la portée des actions correctives. Une étude est alors effectuée sur l'évolution de l'usine manufacturière et les mutations de l'environnement des MOCN, en s'intéressant à l'impact de la CN étendue tout au long de la chaîne numérique.

Mots clés

Machine à commande numérique ouverte, CN étendue, Action corrective, Génération de trajectoires en ligne, Temps réel, STEP-NC, CAO/FAO/CN, Programmation avancée, Usine du futur

Abstract

These research works aim to extend the possibilities for CNC machine tool programming allowing generation of on-line trajectories.

The developed approach begins with an analysis of current programming evolution and limitations of legacy CNC. This analysis argues a lack and need for control optimization by taking into account some geometric data during the machining of a part.

The choice is then focused on the use of open architecture CNC (OCNC), for which the criteria and integration on an industrial-size machine are elaborate.

The developments made on the opportunities and entry points of the OCNC allow the validation and the implementation of corrective actions in real time, using an experimental study for measuring and disturbance of position of a workpiece being machined. On this basis, we structure the concept of an extended CNC (XCNC) by the use of the STEP-NC standard, which especially broadens the scope of corrective actions. Therefore, a study is made on the evolution of the manufacturing plant and CNC environment mutations, focusing on the impact of the XCNC throughout the manufacturing numerical chain.

Key Words

Open architecture CNC, Extended CNC, Corrective Action, Online trajectory generation, Real Time, STEP-NC, CAD/ CAM/CNC, Advanced Programming, Factory of the Future