

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد – تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



THESE

Présentée pour l'obtention du **grade de DOCTORAT 3^{ème} Cycle**

En : Génie Mécanique

Spécialité : Fabrication Mécanique et Productique

Par : ZIANI Belkheir

Sujet

Modélisation et optimisation des trajectoires
de l'outil de coupe pour les MOCN 5 axes

Soutenue publiquement, le 18 / 04 /2026 , devant le jury composé de :

M.BENACHOUR Mustapha	Professeur	Univ. Tlemcen	Président
M.SEBA A Fethi	Professeur	Univ. Tlemcen	Directeur de thèse
M.RAHOU Mohammed	Professeur	ESAA Tlemcen	Co- Directeur de thèse
M.MERGHACHE Sidi Mohammed	Professeur	Univ. Tissemsilt	Examineur 1
M.HADJOU I Fethi	MCA	Univ. Tlemcen	Examineur 2
M.SERDOUN Mohammed Nadjib	MCA	ESAA Tlemcen	Examineur 3

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد – تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



THESE

Présentée pour l'obtention du **grade de DOCTORAT 3^{ème} Cycle**

En : Génie Mécanique

Spécialité : Fabrication Mécanique et Productique

Par : ZIANI Belkheir

Sujet

**Modélisation et optimisation des trajectoires
de l'outil de coupe pour les MOCN 5 axes**

Soutenue publiquement, le 18 / 04 /2026 , devant le jury composé de :

M.BENACHOUR Mustapha	Professeur	Univ. Tlemcen	Président
M.SEBA A Fethi	Professeur	Univ. Tlemcen	Directeur de thèse
M.RAHOU Mohammed	Professeur	ESAA Tlemcen	Co- Directeur de thèse
M.MERGHACHE Sidi Mohammed	Professeur	Univ. Tissemsilt	Examineur 1
M.HADJOU I Fethi	MCA	Univ. Tlemcen	Examineur 2
M.SERDOUN Mohammed Nadjib	MCA	ESAA Tlemcen	Examineur 3

Remerciements

Avant tout, je rends grâce à Allah, le Tout-Puissant, de m'avoir accordé la force, la foi et la persévérance nécessaires pour mener à bien ce modeste travail.

*Je tiens à exprimer ma profonde gratitude à **Monsieur SEBAA Fethi** et **Monsieur RAHOU Mohammed**, mes encadrants, pour leur accompagnement précieux tout au long de cette thèse. Leur disponibilité, leurs conseils avisés, ainsi que leurs qualités humaines remarquables ont été une source constante de motivation et d'enrichissement personnel et scientifique. Je leur adresse mes sincères remerciements et ma reconnaissance la plus profonde.*

*Mes remerciements vont également aux honorables membres du jury. Je remercie chaleureusement **Monsieur BENACHOUR Mustapha** pour l'honneur qu'il m'a fait en acceptant de présider cette soutenance. J'adresse également mes vifs remerciements à **Monsieur MERGHACHE Sidi Mohammed**, **Monsieur HADJOUI Fethi** et **Monsieur SERDOUN Mohammed Nadjib**, pour avoir accepté d'évaluer ce travail et pour l'intérêt bienveillant qu'ils lui ont accordé.*

Je n'oublie pas d'adresser ma gratitude à l'ensemble de mes enseignants, pour les connaissances et le savoir qu'ils m'ont transmis tout au long de mon parcours universitaire.

Enfin, mes pensées reconnaissantes vont à mes collègues de formation, sans exception, pour leur soutien, leurs encouragements, leur camaraderie et les bons moments partagés au cours de ces années. Merci également à toutes les personnes, de près ou de loin, qui ont contribué à la réalisation de ce travail.

DEDICACES

Louange à Allah, sans qui rien n'est possible. C'est grâce à Sa miséricorde et à Sa guidance que j'ai pu achever ce travail.

Je dédie humblement cette thèse :

À mes chers parents, pour leur amour inconditionnel, leur soutien indéfectible et leurs prières constantes. Leur confiance, leurs sacrifices et leurs encouragements ont été ma force à chaque étape. Je leur dois tout.

À toute la famille ZIANI, pour leur présence bienveillante et leur soutien moral.

À mes frères bien-aimés, Bilal et Karim, pour leur affection, leur patience et leur encouragement de chaque instant.

À mes amis fidèles, qui m'ont soutenu de près comme de loin : Djamel, Abdennour, Sid Ali, Mekki, Moussabe, Oussama, Aboubekour, Hossine, Mahjoub, Walid, Hanan, Hicheme, Salah Eddine, Abdeladim, Ichraf, Youcef, Moussa, Omar, Badr Adin, Hossine, Driss, Mangouchi, Khadidja, Mohammed... Je vous remercie pour votre amitié sincère, vos mots d'encouragement et les moments partagés.

À tous mes enseignants, qui ont su éveiller en moi la curiosité, le goût de l'effort et la rigueur scientifique.

À tous ceux qui, d'une manière ou d'une autre, ont contribué à la réalisation de ce travail, je vous adresse mes plus sincères remerciements.

Résumé

L'usinage CN 5 axes constitue une technologie de pointe permettant la réalisation de formes complexes avec une précision élevée grâce à la liberté de mouvement simultané sur cinq axes. Dans ce cadre, cette recherche propose une approche d'optimisation intelligente des trajectoires d'outil, visant à améliorer la qualité d'usinage, la continuité des trajectoires et la réduction des temps de cycle. Une méthode hybride d'interpolation a été développée, combinant les techniques spline cubique, Lagrange et PCHIP afin d'assurer une meilleure fluidité et précision des trajectoires. Parallèlement, un algorithme d'optimisation adaptatif a été conçu pour ajuster automatiquement les paramètres d'interpolation et minimiser l'erreur quadratique moyenne. Les simulations réalisées sous MATLAB, ainsi que les tests sur des trajectoires réelles générées en G-code, démontrent une amélioration significative des performances d'usinage, notamment lorsqu'il est couplé à un algorithme génétique. Enfin, le système intègre des agents d'intelligence artificielle (basés sur GPT-4, Mistral, et Gemini) pour l'analyse, la validation et la génération automatisée de trajectoires, inscrivant ce travail dans la dynamique de l'Industrie 4.0.

Mots-clés : Usinage CN 5 axes ; Optimisation des trajectoires ; Interpolation hybride ; Algorithme adaptatif ; Erreur quadratique moyenne ; Algorithme génétique ; Intelligence artificielle ; Industrie 4.0 ; G-code ; MATLAB

Abstract

Five-axis CNC machining is an advanced manufacturing technology that enables the production of complex geometries with high precision through simultaneous motion along five axes. This research proposes an intelligent optimization approach for toolpath generation, aiming to enhance machining quality, trajectory smoothness, and cycle time efficiency. A hybrid interpolation method combining spline cubique, Lagrange, and PCHIP techniques has been developed to improve continuity and accuracy of tool trajectories. In parallel, an adaptive optimization algorithm was designed to automatically adjust interpolation parameters and minimize the root mean square error. Simulations performed in MATLAB, along with experiments on real 2D/3D G-code trajectories, demonstrate significant machining performance improvements, especially when coupled with a genetic algorithm. Moreover, the system integrates AI-driven agents (based on GPT-4, Mistral, and Gemini) for trajectory analysis, validation, and automated G-code generation, aligning this work with the Industry 4.0 paradigm.

Keywords : 5-axis CNC machining ; Tool path optimization ; Hybrid interpolation ; Adaptive algorithm ; Mean squared error (MSE) ; Genetic algorithm ; Artificial intelligence ; Industry 4.0 ; G-code ; MATLAB

ملخص

يُعدّ التشغيل الآلي بخمسة محاور (CNC 5 Axes) من التقنيات المتقدمة التي تُمكن من تصنيع الأشكال المعقدة بدقة عالية بفضل حرية الحركة المتزامنة على خمسة محاور. تقترح هذه الأطروحة منهجية ذكية لتحسين مسارات الأداة بهدف رفع جودة التشغيل، وتحسين سلسلة المسارات، وتقليل زمن الدورة الإنتاجية. تم تطوير طريقة استيفاء هجينة تجمع بين الطرق Cubic spline ، و Lagrange ، و PCHIP لضمان استمرارية ودقة أعلى في مسارات الأداة. بالتوازي، تم ابتكار خوارزمية تحسين تكيفية تقوم بضبط معاملات الاستيفاء تلقائيًا لتقليل متوسط الخطأ التربيعي. أظهرت عمليات المحاكاة باستخدام MATLAB والتجارب على مسارات حقيقية بتنسيق G-code تحسنًا ملحوظًا في أداء التشغيل، خصوصًا عند دمج الخوارزمية مع الخوارزميات الجينية. كما يدمج النظام وكلاء ذكاء اصطناعي مبنية على GPT-4 و Mistral و Gemini لتحليل المسارات والتحقق من الأداء وتوليد شفرات G-code تلقائيًا، مما يضع هذا العمل في إطار الصناعة 4.0.

الكلمات المفتاحية : التصنيع باستخدام الحاسوب بخمسة محاور ؛ تحسين المسارات ؛ الاستيفاء الهجين ؛ الخوارزمية التكيفية ؛ الخطأ التربيعي المتوسط ؛ الخوارزمية الجينية ؛ الذكاء الاصطناعي ؛ الصناعة 4.0 ؛ G-code ؛ MATLAB

Sommaire

Remerciements	I
Sommaire	VI
Liste des figures	XI
Liste des tableaux.....	XIV
Glossaire	XV
Liste des notations.....	XVI
Introduction Générale	1
Chapitre I État de l'Art	3
I.1 Introduction.....	3
I.2 Historique.....	3
I.3 Définition d'une machine à commande numérique	5
I.4 Revue de la littérature sur les trajectoires d'outil 5 axes	5
I.5 Méthodes et algorithmes existants	19
I.6 Applications sectorielles des systèmes multi-axes	20
I.6.1 Secteur de l'énergie	20
I.6.2 Automobile	21
I.6.3 Fabrication d'outils et de matrices	21
I.6.4 Secteur Médical	22
I.6.5 Industrie aérospatiale	22
Chapitre II Trajectoire de l'outil pour MOCN	26
II.1 Usinage multiaxes.....	26
II.2 Système d'axes cartésiens	26
II.3 Système d'axes supplémentaires	26
II.4 Transformations géométriques et programmation des axes.....	28
II.4.1 Fondements mathématiques des transformations	28

II.4.2	Transformations de mouvement.....	29
II.4.3	Types de transformations de mouvement	29
II.5	Langage de programmation	30
II.6	Langage G-Code et contrôle des axes.....	32
II.7	Interpolation des trajectoires.....	33
II.8	Introduction aux méthodes d’optimisation numérique	35
Chapitre III Modélisation et optimisation des trajectoires d’outil		42
III.1	Introduction.....	42
III.2	Interpolation.....	42
III.3	Limites de l'interpolation	42
III.4	Algorithme d'optimisation du parcours d'outil	43
III.5	Combinaison d'interpolateurs	44
III.5.1	Interpolation de spline cubique.....	44
III.5.2	Interpolation de Lagrange.....	45
III.6	Simulation Matlab.....	46
III.7	Générateur G-Code	49
III.8	Modélisation des trajectoires réelles 2D	50
III.9	Modélisation des trajectoire réel 3D	57
III.10	Conclusion	60
Chapitre IV Analyse comparative et résultats.....		64
IV.1	Introduction.....	64
IV.2	Impact de l’ajout d’un interpolateur sur la précision de la trajectoire	64
IV.3	Optimisation de paramètres d’usinage à l’aide d’un algorithme génétique.....	66
IV.3.1	Optimisation intelligente de paramètres d’interpolation	66
IV.3.2	Algorithme d’optimisation hybride.....	68
IV.4	Résultats expérimentaux	71
IV.5	Conclusion	74

Chapitre V Implémentation du Système Intelligent.....	77
V.1 Introduction.....	77
V.2 Automatisation intelligente.....	78
V.3 Plateforme d’automatisation des workflows (N8N)	78
V.4 Large language models (LLM)	78
V.5 Model context protocol (MCP).....	78
V.6 Interface de programmation (API).....	79
V.7 Architecture générale et philosophie de conception	79
V.8 Description détaillée des nœuds de déclenchement.....	79
V.8.1 Nœud de planification.....	79
V.8.2 Réception de données	80
V.8.3 Obtention d'une page de base de données.....	80
V.8.4 Agents d'intelligence artificielle	81
V.8.5 Modèles de langage large.....	81
V.8.5.1 Google Gemini chat model	82
V.8.5.2 OpenAI GPT-4 chat model.....	82
V.8.5.3 Mistral Cloud chat model	83
V.8.6 Outils mémoire	83
V.8.7 Outils de code	84
V.8.8 Nœuds conditionnels et de contrôle de flux.....	86
V.8.9 Nœuds d'attente et de synchronisation.....	86
V.8.10 Modification et transformation des données.....	86
V.8.11 Sérialisation et gestion de formats	87
V.8.12 Système de fichiers et persistance	87
V.8.13 Fusion et agrégation de données.....	87
V.8.14 Communication et intégration	88

V.8.15	Nœud de code	88
V.9	Évolution des automatisations pour MOCN Intelligentes	88
V.9.1	Première automatisation : système d'interpolation CNC basique	89
V.9.2	Deuxième automatisation : système CNC multi-algorithmes.....	90
V.9.3	Troisième automatisation : système CN intelligent optimisé	92
V.9.4	Performances et bénéfices industriels.....	101
V.9.5	Intégration industrie 4.0	102
V.9.6	Système MOCN intelligent pour l'usinage 4.0.....	103
V.10	Implémentation d'un système de conversion XYZ vers G-code basé sur l'IA.....	103
V.10.1	Architecture du système.....	103
	Vue d'ensemble	104
V.10.2	Implémentation détaillée.....	104
	<i>V.10.2.1 Structure du workflow n8n</i>	<i>105</i>
	<i>V.10.2.2 Format d'entrée supporté.....</i>	<i>105</i>
	<i>V.10.2.3 Prompt système optimisé.....</i>	<i>105</i>
V.10.3	Analyse comparative des modèles d'IA pour la génération de G-code.....	106
	<i>V.10.3.1 Données d'entrée.....</i>	<i>106</i>
	<i>V.10.3.2 Résultats obtenus</i>	<i>106</i>
	<i>V.10.3.3 Analyse comparative détaillée</i>	<i>107</i>
	<i>V.10.3.4 Tableau d'évaluation des critères techniques.....</i>	<i>108</i>
	<i>V.10.3.5 Visualisation du parcours d'outil</i>	<i>109</i>
	<i>V.10.3.6 Analyse des risques en production.....</i>	<i>109</i>
V.10.4	Métriques de performance	110
V.10.5	Conclusion de l'analyse.....	110
V.11	Impact sur l'évolution des machines-outils intelligentes	111
V.12	Conclusion	113

Conclusion générale	115
Références bibliographiques et webographiques	117

Liste des figures

Chapitre 1 État de l'Art

Fig. I. 1 Machine CNC à 5 axes [2].	5
Fig. I. 2 Simulation et vérification des mouvements de l'outil.	7
Fig. I. 3 Mouvement de la fraise selon les exigences de lissage et de non-interférence [5].	8
Fig. I. 4 Erreur de forme mesurée avant et après l'application de la compensation [6].	9
Fig. I. 5 Erreurs des surfaces usinées pratiquement [7].	10
Fig. I. 6 Taux d'enlèvement de matière calculés à partir du temps d'usinage mesuré [9].	11
Fig. I. 7 Vérification CN réalisée dans CG-Tech Vericut® [10].	12
Fig. I. 8 Résultat de la contrainte d'erreur du trajet d'outil de test, Les erreurs de lissage de position [12].	14
Fig. I. 9 Machine virtuelle dans Vericut® [14].	15
Fig. I. 10 Came cylindrique balayée [15].	16
Fig. I. 11 Composant de rotor haute précision fabriqué par usinage CN multi-axes [27].	20
Fig. I. 12 Composant moteur (culasse) fabriqué par usinage CN 5 axes [27].	21
Fig. I. 13 Moule industriel en acier trempé fabriqué par usinage CN multi-axes [27].	22
Fig. I. 14 Application de l'usinage multi-axes dans la fabrication de turbines aéronautiques [27].	23

Chapitre 2 : Trajectoire de l'outil pour MOCN

Fig. II. 1 Axes primaires et axes additionnels.	27
Fig. II. 2 Exemple de profil d'interpolation circulaire (G02/G03) [41]	33
Fig. II. 3 Illustration du croisement en un point qui divise le génome de deux solutions en un point arbitraire et les réassemble pour obtenir deux nouvelles solutions [47].	38
Fig. II. 4 Mutation en codage binaire [47].	39

Chapitre 3 Modélisation et optimisation des trajectoires d'outil

Fig. III. 1 Algorithme d'optimisation du parcours d'outil (TPOA).	43
Fig. III. 2 Graphique d'erreur du TPOA par rapport à spline cubique et Lagrange.	44
Fig. III. 3 Comparaison entre les trois méthodes.	46
Fig. III. 4 Comparaison des erreurs de trois méthodes.	47
Fig. III. 5 Graphique de trajectoire et d'erreur pour l'exemple 7.	48
Fig. III. 6 G-Code Générateur version 1 testé sur NC Viewer V1.2.0.	49
Fig. III. 7 G-Code Générateur version 2 simulation sur Octave.	50

Fig. III. 8 Trajectoire trochoïdale.....	50
Fig. III. 9 Modélisation de la trajectoire trochoïdale.	51
Fig. III. 10 Modélisation de la trajectoire Spirale logarithmique.	53
Fig. III. 11 Modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et Lagrange.	54
Fig. III. 12 Modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et PCHIP.	55
Fig. III. 13 Modélisation de la trajectoire Sigmoidale.....	56
Fig. III. 14 Modélisation de la trajectoire de Hilbert.	57
Fig. III. 15 Modélisation de la trajectoire spirale.	58
Fig. III. 16 Modélisation de la trajectoire Hilbert 3D.....	59
Fig. III. 17 Comparaison des erreurs de trois méthodes.	60

Chapitre 4 Étude comparative et intégration de l’algorithme génétique

Fig. IV. 1 Résultats obtenus avec trois interpolateurs.	64
Fig. IV. 2 Résultats de l’erreur quadratique moyenne obtenus avec trois interpolateurs.	65
Fig. IV. 3 STPOA (Smart Tool Path Optimization Algorithm).	66
Fig. IV. 4 STPOA avec deux méthode d’interpolation (spline cubique, Lagrange) optimisé par le Génétique Algorithme.....	67
Fig. IV. 5 TPOA avec deux méthode d’interpolation (spline cubique, PCHIP).....	68
Fig. IV. 6 Génétique Algorithme avec deux méthode d’interpolation (spline cubique, PCHIP).	69
Fig. IV. 7 Méthode TPOA avec deux méthode d’interpolation (spline cubique, PCHIP) optimisé par le Génétique Algorithme.....	69
Fig. IV. 8 HOA, (Hybrid Optimization Algorithm).....	70
Fig. IV. 9 Machine utilisée BoxFord 160 LTCL.....	71
Fig. IV. 10 Logiciel Boxford V10 CAD/CAM.....	72
Fig. IV. 11 Pièce à usiner après découpe selon la méthode TPOA.....	73

Chapitre 5 Architecture et implémentation du système intelligent

Fig. V. 1 Nœud de planification dans n8n.	79
Fig. V. 2 Nœud de déclenchement pour la réception de messages dans n8n.....	80
Fig. V. 3 Agents d’intelligence artificielle.	81
Fig. V. 4 Modèles de langage large.....	82
Fig. V. 5 Outils mémoire.	84

Fig. V. 6 Outils de code.	85
Fig. V. 7 Nœuds conditionnels.....	86
Fig. V. 8 Édition et transformation des données.....	87
Fig. V. 9 Nœud de code.	88
Fig. V. 10 Système d'interpolation CNC basique.	89
Fig. V. 11 Système CNC multi-algorithmes.	91
Fig. V. 12 Système CN intelligent optimisé.	93
Fig. V. 13 Agent interpolation.	93
Fig. V. 14 Agent de validation de la qualité.	94
Fig. V. 15 Agent décision production.	95
Fig. V. 16 Système de contrôle qualité.	101
Fig. V. 17 Système de conversion XYZ vers G-code basé sur l'IA (ChatGpt).	104
Fig. V. 18 Système de conversion XYZ vers G-code basé sur l'IA (Mistral AI).	104

Liste des tableaux

Chapitre 2 : Trajectoire de l'outil pour MOCN

Tableau. II. 1	Tableau récapitulatif des axes CN.	28
----------------	---	----

Tableau. II. 2	Explication des Codes G M S T.....	32
----------------	------------------------------------	----

Chapitre 3 Modélisation et optimisation des trajectoires d'outil

Tableau. III. 1	Erreur quadratique moyenne des différentes méthodes d'interpolation.	48
-----------------	---	----

Tableau. III. 2	Erreur quadratique moyenne de la trajectoire trochoïdale..	52
-----------------	---	----

Tableau. III. 3	Erreur quadratique moyenne de la trajectoire Spirale logarithmique..	52
-----------------	---	----

Tableau. III. 4	Erreur quadratique moyenne de la trajectoire Lemniscate de Bernoulli.....	54
-----------------	---	----

Tableau. III. 5	Erreur quadratique moyenne de la trajectoire sigmoïde.	55
-----------------	---	----

Tableau. III. 6	Erreur quadratique moyenne de la trajectoire de Hilbert.	57
-----------------	---	----

Tableau. III. 7	Erreur quadratique moyenne de la trajectoire spirale.	58
-----------------	--	----

Tableau. III. 8	Erreur quadratique moyenne la trajectoire Hilbert 3D.....	59
-----------------	---	----

Chapitre 4 Étude comparative et intégration de l'algorithme génétique

Tableau. IV. 1	Erreur quadratique moyenne obtenus avec trois interpolateurs.....	65
----------------	---	----

Tableau. IV. 2	Résumé des paramètres optimisés et vérification des bornes.....	67
----------------	---	----

Tableau. IV. 3	Erreur quadratique moyenne avec deux méthode d'interpolation.	68
----------------	--	----

Tableau. IV. 4	Erreur quadratique moyenne obtenue en intégrant l'algorithme génétique.....	69
----------------	---	----

Tableau. IV. 5	Erreur quadratique moyenne de l'algorithme hybride.....	70
----------------	---	----

Tableau. IV. 6	Paramètres de coupe utilisés dans The BoxFord 160 LTCL.....	72
----------------	---	----

Tableau. IV. 7	Rugosité de surface, Ra (mm) pour différentes avances, F (mm/tour).....	73
----------------	---	----

Chapitre 5 Architecture et Implémentation du Système Intelligent

Tableau. V. 1	Performances et bénéfices industriel.	102
---------------	--	-----

Tableau. V. 2	Tableau comparatif des coordonnées.....	107
---------------	---	-----

Tableau. V. 3	Analyse comparative des critères de génération de G-code	108
---------------	--	-----

Tableau. V. 4	Comparaison des performances entre Mistral et ChatGPT	110
---------------	---	-----

Tableau. V. 5	Impacts et bénéfices industriels des technologies d'optimisation intelligente.....	111
---------------	--	-----

Glossaire

AFNOR : Association Française de Normalisation (norme pour les axes et mouvements)

API : Application Programming Interface (interface de programmation pour intégration)

CAO / CAD : Conception Assistée par Ordinateur (Computer-Aided Design)

FAO / CAM : Fabrication Assistée par Ordinateur (Computer-Aided Manufacturing)

NC / CN : Commande Numérique (Numerical Control)

CNC : Commande Numérique par Calculateur

MOCN : Machine-Outil à Commande Numérique

DNC : Direct Numerical Control (pilotage direct d'une machine)

ERP : Enterprise Resource Planning (planification des ressources)

GPT : Generative Pre-trained Transformer

HTTP / HTTPS : HyperText Transfer Protocol / Secure

IA : Intelligence Artificielle

ISO : International Organization for Standardization (normes CNC)

IoT : Internet of Things (internet des objets)

JSON : JavaScript Object Notation

LLM : Large Language Model

MCP : Model Context Protocol

MES : Manufacturing Execution System

MIT : Massachusetts Institute of Technology (origine de la CN)

TPOA : Tool Path Optimization Algorithm

STPOA : Smart Tool Path Optimization Algorithm

HOA : Hybrid Optimization Algorithm

GA : Genetic Algorithm

TSP : Traveling Salesman Problem

RMSE : Root Mean Square Error

CW : ClockWise (sens horaire)

CCW : Counter ClockWise (sens antihoraire)

Liste des notations

T : Matrice de transformation homogène (rotation + translation)

R_(3×3) : Matrice de rotation 3×3

t_(3×1) : Vecteur de translation 3×1

t_x, t_y, t_z : Composantes de translation

R(X, θ) : Rotation autour de l'axe X d'un angle θ

R(Y, θ) : Rotation autour de l'axe Y d'un angle θ

R(Z, θ) : Rotation autour de l'axe Z d'un angle θ

θ : Angle de rotation

S(x) : Spline cubique globale

P(x) : Polynôme de Lagrange

RMSE_T : Erreur quadratique moyenne pour TPOA

RMSE_LA : Erreur quadratique moyenne pour Lagrange

RMSE_C : Erreur quadratique moyenne pour spline cubique

RMSE_P : Erreur quadratique moyenne pour PCHIP

F : Avance (mm/tour)

N : Vitesse de broche (tr/min)

Ra : Rugosité (μm)

p : Profondeur de passe (mm)

Vc : Vitesse de coupe (m/min)

Introduction Générale

L'industrie moderne, fortement influencée par les exigences croissantes en matière de précision, de complexité géométrique et de compétitivité, s'oriente résolument vers l'intégration de technologies avancées dans ses procédés de fabrication. Dans ce contexte, l'usinage à commande numérique en cinq axes s'impose comme une solution incontournable pour la réalisation de pièces complexes, aux tolérances serrées, tout en optimisant le rendement et la qualité des produits finis.

L'usinage 5 axes se distingue des configurations classiques 3 axes par sa capacité à orienter l'outil selon plusieurs directions simultanément, permettant ainsi de réduire le nombre de repositionnements, d'améliorer la finition des surfaces, et de limiter les efforts mécaniques.

Toutefois, cette flexibilité accrue s'accompagne d'un défi majeur. La génération et l'optimisation des trajectoires d'outil doivent être à la fois précises, continues et compatibles avec les contraintes mécaniques et cinématiques de la machine.

L'objectif recherché est le développement d'une approche hybride alliant des méthodes d'interpolation avancées et des algorithmes d'optimisation intelligente pour l'amélioration de la qualité des trajectoires, la minimisation des erreurs géométriques ainsi que la réduction des temps de cycle d'usinage.

La structure de ce travail de recherche s'articule autour de cinq chapitres.

Le premier chapitre est consacré à un état de l'art approfondi des technologies d'usinage multi-axes et des méthodes existantes de génération de trajectoires. Il y est question de l'évolution historique des machines à commande numérique, de leurs domaines d'application, ainsi que des principales stratégies d'optimisation développées dans la littérature. Ce chapitre met en lumière les limites des approches actuelles, ce qui justifie le recours à de nouvelles méthodes hybrides.

Le deuxième chapitre introduit les fondements numériques des trajectoires CN et des systèmes d'axes. On y présente la modélisation géométrique des axes de déplacement (linéaires et rotatifs), les transformations spatiales utilisées en cinématique multi-axes, ainsi que les différents langages de programmation CN, notamment le G-code. Une attention particulière est portée sur les méthodes d'interpolation numérique telles que la spline cubique, l'interpolation de Lagrange, la méthode PCHIP et les polynômes de Chebyshev. Le chapitre se conclut par une introduction aux algorithmes évolutionnaires et à l'optimisation numérique.

Le troisième chapitre présente les contributions méthodologiques de la thèse. Il introduit un modèle de génération et d'optimisation de trajectoires d'outil, basé sur une combinaison d'interpolateurs. Plusieurs trajectoires de test (trochoïdale, spirale logarithmique, lemniscate de Bernoulli, sigmoïde, courbe de Hilbert, etc.) sont modélisées et utilisées pour comparer les performances des différentes méthodes. Par ailleurs, un algorithme intelligent d'optimisation, dénommé TPOA (Tool Path Optimization Algorithm), est introduit. Ce dernier ajuste dynamiquement les paramètres d'interpolation pour minimiser l'erreur quadratique moyenne.

Le quatrième chapitre est dédié à l'étude comparative et à l'intégration de l'algorithme génétique dans le processus d'optimisation. Il analyse l'impact de l'ajout d'un interpolateur sur la précision, la continuité et la régularité des trajectoires. Des simulations et tests pratiques sont réalisés sur des trajectoires réelles à l'aide de MATLAB et de générateurs de G-code. Les résultats démontrent que l'algorithme TPOA, lorsqu'il est couplé à un algorithme génétique, permet une réduction significative des erreurs, une amélioration de la fluidité des trajectoires et une optimisation globale du processus d'usinage.

Le cinquième et dernier chapitre détaille l'architecture logicielle du système intelligent développé dans cette thèse. Il présente l'automatisation des workflows à l'aide de la plateforme N8N, la conception d'agents spécialisés pour l'analyse avancée des trajectoires, la validation automatique des performances (RMSE, contraintes cinématiques), et l'intégration de modèles de langage (LLM) pour la génération assistée de G-code et la gestion des prompts. Ces éléments montrent la faisabilité d'une intégration directe des méthodes d'optimisation dans une chaîne logicielle orientée Industrie 4.0.

À travers cette démarche, cette thèse vise à contribuer à l'amélioration des procédés d'usinage 5 axes en proposant des solutions innovantes alliant rigueur mathématique et intelligence algorithmique. Les travaux menés ouvrent également des perspectives prometteuses pour l'intégration de l'intelligence artificielle dans les systèmes de fabrication avancés, en vue de développer des trajectoires plus intelligentes, adaptatives et performantes.

Cette thèse se termine par une conclusion général suivie de perspectives.

Chapitre I

État de l'Art

Chapitre I

État de l'Art

I.1 Introduction

L'usinage en cinq axes est une technologie avancée qui permet de déplacer un outil de coupe sur cinq axes différents simultanément, ce qui offre une flexibilité et une précision accrues pour la fabrication de pièces complexes. Cette technique est largement utilisée dans des secteurs tels que l'aérospatiale, l'automobile et l'ingénierie biomédicale, où la précision et l'efficacité sont primordiales.

L'état de l'art de l'optimisation des trajectoires d'outil en 5 axes englobe un large éventail de recherches et de développements technologiques. Ces travaux visent à améliorer la précision des mouvements, à réduire les temps de cycle, à minimiser l'usure des outils et à maximiser la qualité de surface des pièces usinées. Les recherches récentes se concentrent sur plusieurs aspects clés, notamment la modélisation des trajectoires, les algorithmes de planification et les techniques d'optimisation.

Les trajectoires d'outil optimisées jouent un rôle crucial dans l'usinage en 5 axes, car elles déterminent directement la qualité de la pièce finie et l'efficacité du processus d'usinage. Les chercheurs ont développé diverses méthodes pour générer des trajectoires optimales, y compris des algorithmes heuristiques, des techniques basées sur les gradients et des méthodes d'optimisation stochastique. Chaque approche présente ses avantages et ses inconvénients, et le choix de la méthode dépend souvent des spécificités de l'application et des contraintes opérationnelles.

Le domaine des trajectoires d'outil en 5 axes est dynamique et en constante évolution. Il reflète les avancées technologiques et les exigences croissantes de l'industrie moderne. Les travaux actuels visent à surmonter les défis techniques et à explorer de nouvelles voies pour améliorer davantage l'efficacité et la précision de l'usinage en 5 axes.

I.2 Historique

La commande numérique (CN), ou *Numerical Control (NC)*, est une technologie qui a émergé au milieu du XXe siècle. Son origine remonte à 1952, aux États-Unis, où elle fut développée dans le cadre d'un projet mené par l'U.S. Air Force en collaboration avec le Massachusetts Institute of Technology (MIT). Ce développement est étroitement lié au nom de John Parsons

(1913–2007), considéré comme l'un des pionniers de cette technologie. Toutefois, son application industrielle réelle ne s'est généralisée qu'au début des années 1960 [1].

La toute première machine-outil à commande numérique date en réalité de 1942, utilisée pendant la Seconde Guerre mondiale pour l'usinage de pièces de réacteurs particulièrement complexes. C'est à l'exposition de Chicago en 1952 que la technologie a été véritablement lancée au grand public, notamment par des entreprises comme la Cincinnati Milling Corporation et le MIT [1].

Le véritable essor de la CN a eu lieu dans les années 1970 avec l'arrivée de la commande numérique par ordinateur, notamment autour de 1972, et s'est accéléré avec la démocratisation des micro-ordinateurs dans la décennie suivante. Cette évolution a profondément transformé le secteur manufacturier, en particulier l'industrie de la métallurgie [1].

La commande numérique a ainsi apporté des avantages considérables, non seulement dans l'usinage proprement dit, mais également dans des domaines comme la métrologie, le soudage, l'oxycoupage, le guidage des faisceaux laser, l'électroérosion, et le découpage par jet d'eau [1].

Dans les années 1980 et 1990, cette technologie a continué de se diffuser massivement. En France, par exemple, on comptait environ 60 000 MOCN installées entre 1990 et 1998, principalement dans les PME pour moderniser l'outil de production. Au Canada, notamment au Québec, la majorité des entreprises manufacturières ont également adopté cette technologie. Le Japon, quant à lui, détient plus de 50 % du marché mondial des MOCN, preuve de son leadership dans ce domaine [1].

Il est important de noter que l'apparition et le développement de la commande numérique répondaient avant tout à un problème technique, celui de la complexité des pièces à usiner bien plus qu'à un simple impératif économique. Aujourd'hui, grâce aux progrès constants de la microélectronique et de l'informatique, la commande numérique reste en pleine évolution et continue de jouer un rôle central dans les ateliers de fabrication à travers le monde [1].

La terminologie employée dans le domaine de la commande numérique peut être présentée de la manière suivante [1]:

- La technologie CN (Numerical Control) : terme utilisé pour le contrôle numérique avec des verniers, des appareils de mesure, etc;
- La technologie CNC (Computerized Numerical Control) : l'ordinateur est utilisé pour contrôler les déplacements dans les MOCN;

- Le terme DCN (Direct Numerical Control) signifie souvent qu'une machine est pilotée par un ordinateur ;

I.3 Définition d'une machine à commande numérique

Une machine à commande numérique est une machine, totalement ou partiellement automatique. Dans une MOCN les ordres sont, contrairement aux machines conventionnelles, transmis sous forme d'un programme à l'aide d'un support externe. Lorsqu'une machine à outil est équipée d'une commande numérique ayant la capacité d'effectuer le calcul de coordonnées des points définissant la trajectoire de l'outil, on parle d'une CNC [2].

La figure I.1 illustre une machine CNC à trois axes (CNC Router).



Fig. I. 1 Machine CNC à 3 axes [2].

I.4 Revue de la littérature sur les trajectoires d'outil 5 axes

L'usinage en 5 axes est une technique avancée permettant de déplacer un outil de coupe simultanément sur cinq axes différents. Contrairement aux machines-outils traditionnelles à 3 axes, qui ne peuvent se déplacer que sur les axes X, Y, et Z, les machines à 5 axes ajoutent deux axes rotatifs, offrant ainsi une flexibilité et une complexité accrues. Cette capacité permet de fabriquer des pièces complexes avec une précision et une efficacité inégalées, ce qui est particulièrement crucial dans des industries telles que l'aérospatiale, l'automobile, et la biomédicale, où la précision des composants est essentielle.

L'optimisation des trajectoires d'outil en 5 axes est essentielle pour maximiser la précision, l'efficacité et la qualité des pièces usinées. En optimisant les trajectoires, il est possible de minimiser le temps de cycle de fabrication et de réduire l'usure des outils, tout en assurant une finition de surface de haute qualité. Cela permet non seulement de gagner du temps et des

ressources, mais aussi de garantir que les pièces produites répondent aux spécifications rigoureuses requises dans des applications critiques.

De nombreuses recherches ont été menées dans ce domaine, démontrant que l'optimisation des trajectoires permet également d'éviter les collisions et de prolonger la durée de vie des machines-outils, rendant ainsi le processus de fabrication plus fiable et durable. Parmi ces recherches, on peut citer :

I.4.1 Génération de trajectoires d'outil lisses pour l'usinage 5 axes d'une surface maillée triangulaire à genre non nul

Yuwen Sun et al. ont introduit des méthodes innovantes pour générer des trajectoires d'outil sur des surfaces maillées. Le processus implique la transformation de la surface en utilisant le mappage conforme pour obtenir une surface de genre 0 et générer des trajectoires d'outil dans une région rectangulaire.

L'article se concentre sur le calcul des incréments de paramètre pour la ligne de guidage et propose une méthode d'amélioration de la précision pour déterminer les points de contact de l'outil. De plus, la recherche explore l'optimisation de l'orientation de l'outil pour améliorer les performances des machines-outils à 5 axes. Implémentées en langage C, les méthodes sont rigoureusement testées avec divers exemples, fournissant des résultats expérimentaux détaillés qui démontrent leur efficacité dans la génération de trajectoires d'outil et assurant des transitions fluides dans l'orientation de l'outil [3].

I.4.2 Amélioration de l'optimisation de la planification de trajectoire par PSO pour le fraisage sur flanc en usinage 5 axes

Hsin-Ta Hsieh vise à améliorer la planification des trajectoires d'outil pour le fraisage à flanc en usinage 5 axes des surfaces réglées, en se concentrant sur la minimisation des erreurs d'usinage. L'étude utilise deux algorithmes avancés d'optimisation par essaim particulaire (PSO) - l'APSO (Advanced PSO) et le FIPS (Fully Informed PSO) - pour améliorer la qualité des solutions optimales et l'efficacité de la recherche. En élargissant l'espace des solutions grâce à des contraintes assouplies, la recherche atteint des erreurs d'usinage plus faibles par rapport aux travaux précédents. Notamment, le FIPS s'avère très efficace pour réduire les erreurs dans toutes les expériences, tandis que le PSO standard excelle avec un faible nombre de positions de l'outil.

Les résultats apportent des informations précieuses sur l'optimisation basée sur le PSO de la planification des trajectoires d'outil, faisant progresser la compréhension des algorithmes avancés dans ce contexte [4].

La figure I.2 présente la simulation et la vérification des mouvements de l'outil [4].

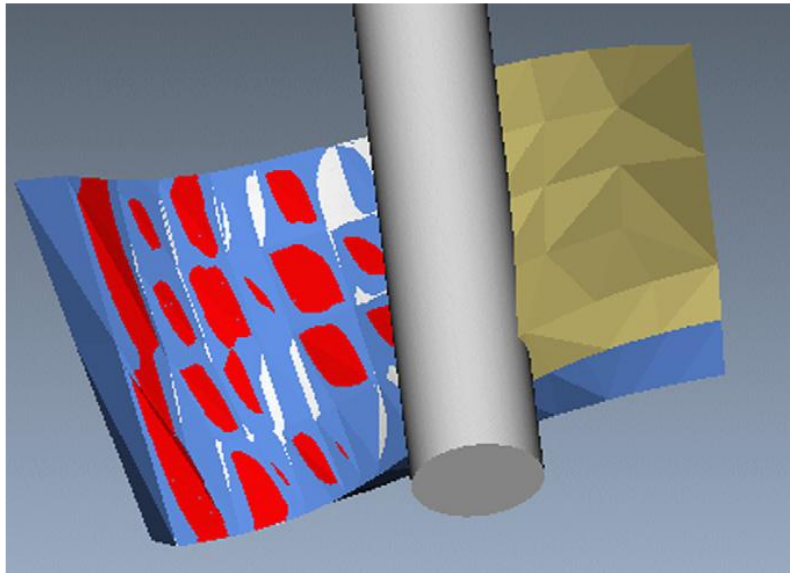


Fig. I. 2 Simulation et vérification des mouvements de l'outil [4].

I.4.3 Planification de trajectoire pour le fraisage 5 axes de surfaces réglées en prenant en compte les positions optimales de la fraise

Shuoxue Sun et al. ont proposé une méthode innovante de planification des trajectoires d'outil pour le fraisage à flanc 5 axes de surfaces réglées. L'accent est mis sur les positions optimales de la fraise, appelées Cutter Location (CL) sous de multiples contraintes géométriques. L'approche implique un modèle de positionnement de l'outil à trois points de contact, utilisant un algorithme méta-heuristique pour construire un pool d'orientations de l'outil. Des courbes angulaires de lissage sont dérivées pour éviter les interférences et assurer la fluidité géométrique. Un modèle de détermination adaptative des intervalles permet la planification simultanée de multiples CL avec le calcul parallèle. Les tests sur des surfaces représentatives démontrent une amélioration de la précision de l'usinage et du temps de calcul par rapport aux méthodes précédentes. L'étude est soutenue par la Fondation nationale des sciences naturelles de Chine et la Fondation chinoise pour la science postdoctorale [5].

La figure I.3 montre le mouvement de la fraise selon les exigences de lissage et de non-interférence.

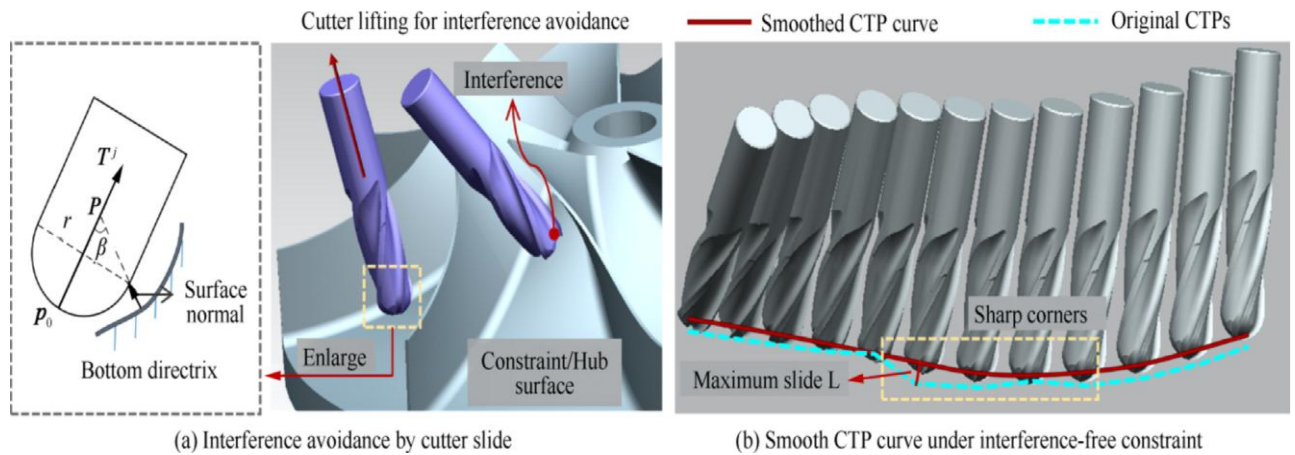


Fig. I. 3 Mouvement de la fraise selon les exigences de lissage et de non-interférence [5].

I.4.4 Génération auto-optimisante de trajectoires d'outil pour l'usinage 5 axes en tenant compte de la déflexion de l'outil

Marc-André Dittrich et al. ont introduit une méthode de planification de processus auto-optimisante pour le fraisage 5 axes qui traite la déflexion de l'outil. L'approche combine des simulations de retrait de matériau, des mesures d'erreur de forme et l'apprentissage automatique pour prédire et adapter automatiquement la trajectoire de l'outil, entraînant une réduction de 50 % de l'erreur de forme. Le modèle de régression utilisé montre une forte corrélation entre les déviations de forme mesurées et prédites, prouvant son applicabilité à des poches de formes différentes. Le processus implique une simulation de retrait de matériau parallèle au processus, stockant les conditions de coupe dans le modèle de la pièce. Diverses méthodes, y compris les modèles de poutres en porte-à-faux élastiques et la méthode des éléments finis (FEM), prédisent et compensent la déflexion de l'outil.

L'article décrit également un programme de mesure dérivé du modèle de pièce usinée, assignant la déviation de forme à des dexels individuels. L'étude explore le comportement du processus d'apprentissage et la transférabilité de l'approche à différentes géométries de pièces [6].

La figure I.4 met en évidence l'erreur de forme mesurée avant et après l'application de la compensation [6].

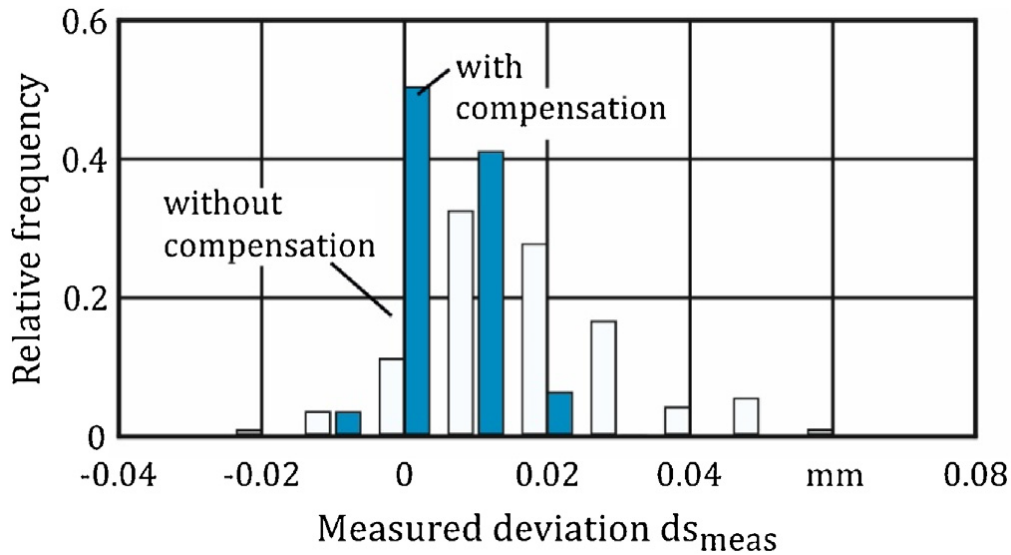


Fig. I. 4 Erreur de forme mesurée avant et après l'application de la compensation [6].

I.4.5 Optimisation iso-paramétrique des trajectoires CN basée sur la génération adaptative de grilles

Wei He a introduit un schéma d'optimisation innovant pour les trajectoires d'outils CN iso-paramétriques, exploitant la génération de grilles adaptatives pour résoudre des problèmes tels que le chevauchement et les inefficacités découlant des plus petits intervalles dans les systèmes CAM. Le cœur de l'optimisation implique l'emploi d'une grille adaptative pour générer une trajectoire d'outil optimale, pouvant potentiellement produire le même nombre de trajectoires d'outils discrètes que l'approche iso-paramétrique mais avec des erreurs d'usinage réduites. Alternativement, elle peut produire des erreurs équivalentes à la trajectoire d'outil iso-paramétrique tout en employant moins de trajectoires d'outils discrètes, améliorant ainsi l'efficacité.

Le schéma est salué pour sa robustesse, sa concision et son intégration transparente dans les systèmes CAM, démontrant sa capacité à générer automatiquement des trajectoires d'outils optimales pour l'usinage de surfaces sculptées [7].

La figure I.5 représente les erreurs observées sur les surfaces usinées pratiquement [7].

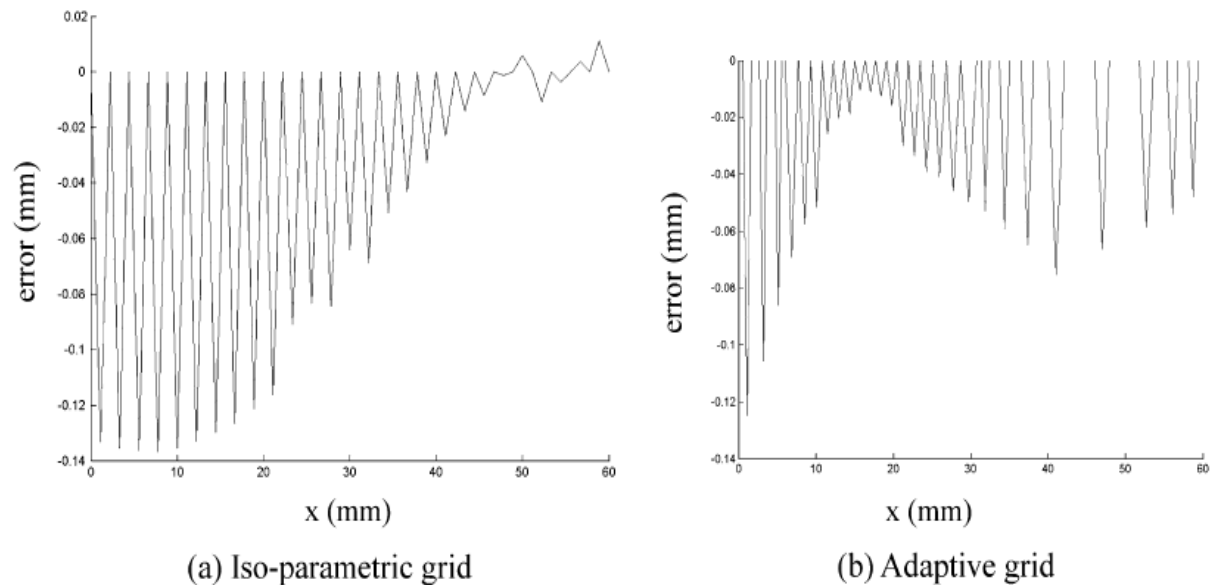


Fig. I. 5 Erreurs des surfaces usinées pratiquement [7].

I.4.6 Optimisation des trajectoires dans le post-processeur des machines-outils à commande numérique 5 axes

Jiangang Li propose un nouvel algorithme d'optimisation des trajectoires d'outil pour les machines-outils à cinq axes au sein du postprocesseur. L'algorithme utilise l'interpolation par spline cubique pour lisser la trajectoire de l'outil, réduisant ainsi les erreurs non linéaires. La densification des données est utilisée pour ajouter davantage de points le long de la trajectoire de l'outil, qui sont ensuite convertis en codes NC. L'algorithme est testé sur une machine-outil à cinq axes non orthogonale, montrant une précision et une qualité de surface améliorées. La vérification des erreurs non linéaires garantit la précision de la trajectoire de l'outil. La trajectoire d'outil optimisée, obtenue grâce à l'interpolation par spline cubique et au traitement des erreurs, est plus lisse.

Elle offre une plus grande précision pour l'usinage des pièces. Le processus implique la génération d'un fichier de position de la fraise à l'aide d'un logiciel CAM, comprenant les points centraux de la pointe de la fraise et les vecteurs de posture de la fraise [8].

I.4.7 Optimisation de la forme des trajectoires en fraisage trochoïdal par courbes B-spline

Adam Jacso a introduit une nouvelle méthode d'optimisation des trajectoires d'outil pour le fraisage trochoïdal en utilisant des courbes B-spline. Bien que les avantages de la productivité et de la durée de vie des outils du fraisage trochoïdal soient bien connus, les études précédentes se sont principalement concentrées sur des trajectoires d'outil en forme de cercle et de cycloïde. L'algorithme de Jacso vise à maximiser le taux moyen d'enlèvement de matière (MRR) et à contrôler la charge de l'outil en optimisant les points de contrôle des courbes B-spline grâce à un algorithme d'évolution différentielle. La méthode développée surpasse les trajectoires d'outil traditionnelles et les systèmes CAM modernes en termes de productivité. L'article souligne l'importance de calculer la courbure de la trajectoire de l'outil et d'optimiser l'engagement de la fraise le long de la courbe B-spline pour des résultats efficaces. L'objectif principal est de maximiser le MRR moyen. L'étude analyse la productivité et l'évolution de la charge de l'outil pour différents paramètres, offrant une évaluation comparative avec d'autres stratégies [9].

La figure I.6 illustre les taux d'enlèvement de matière calculés à partir du temps d'usinage mesuré.

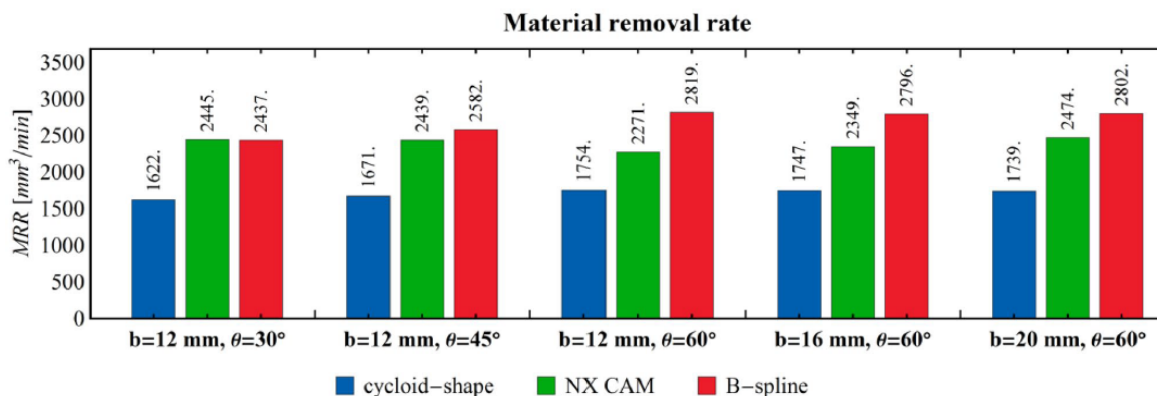


Fig. I. 6 Taux d'enlèvement de matière calculés à partir du temps d'usinage mesuré [9].

I.4.8 Planification de trajectoires pour l'usinage 5 axes à l'aide d'un algorithme évolutif viral multi-objectif

N. A. Fountas présente une méthodologie innovante pour optimiser les trajectoires d'outils dans l'usinage CN à 5 axes de surfaces sculptées. L'approche exploite un algorithme évolutionnaire viral avec des opérateurs viraux et un contrôle de système intelligent pour optimiser à la fois l'erreur de surface d'usinage et le temps d'usinage. L'algorithme surpasse les méthodes conventionnelles dans l'optimisation des objectifs de qualité et considère les paramètres

d'usinage influents, offrant une plateforme d'optimisation basée sur la simulation. La méthodologie de Fountas encourage l'exploration globale et contribue à des perspectives précieuses dans le domaine. Notamment, l'algorithme atteint des solutions d'optimum global tout en évitant les résultats quasi-optimaux. La validation par des tests de simulation et des processus d'usinage réels confirme l'efficacité de l'algorithme évolutionnaire viral dans l'optimisation des trajectoires d'outils [10].

La figure I.7 montre la vérification NC réalisée dans CG-Tech Vericut® et les différentes configurations d'usinage simulées.

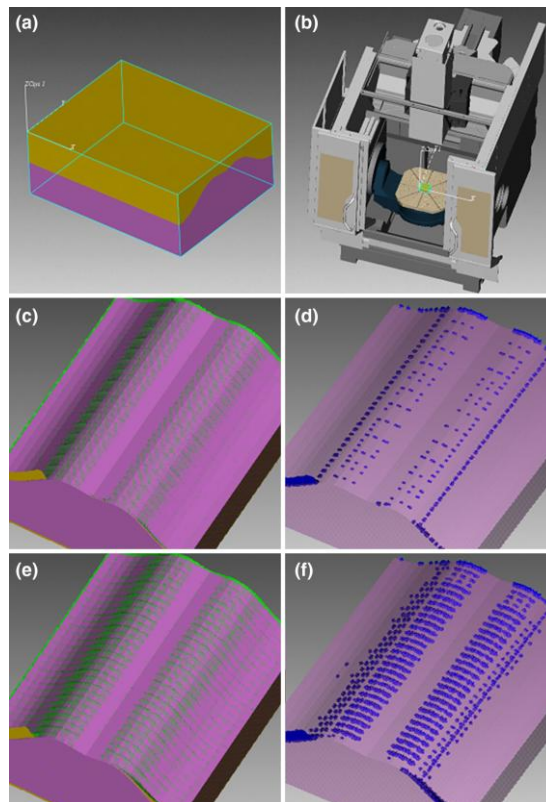


Fig. I. 7 Vérification CN réalisée dans CG-Tech Vericut® [10] :

- (a) : pièce avec brut défini ;
- (b) : configuration d'usinage virtuelle ;
- (c) solide coupé simulé avec GA évolutif viral ;
- (d) région d'erreur excessive pour le modèle GA évolutif viral ;
- (e) solide coupé simulé du GA conventionnel ;
- (f) région d'erreur excessive du modèle GA conventionnel.

I.4.9 Optimisation de la trajectoire pour le frittage laser sélectif (SLS) basée sur l'apprentissage profond

D. H. Kim a exploré l'optimisation des trajectoires d'outils dans les processus de Frittage Laser Sélectif (SLS), une technique de fabrication additive populaire. Le SLS utilise la puissance laser pour lier le matériau en poudre en structures complexes, reconnu pour sa capacité à créer facilement des géométries complexes avec un effort de fabrication supplémentaire minimal.

Dans le contexte du SLS, la sélection optimale de la trajectoire d'outil est cruciale pour l'intégrité des structures imprimées en 3D, visant à minimiser les gradients de température et à réduire les contraintes résiduelles non désirées. L'objectif principal de l'article est de tirer parti des techniques d'apprentissage profond pour accélérer la simulation des processus SLS et obtenir des résultats numériques précis.

L'apprentissage profond permet aux ordinateurs d'apprendre et de détecter des modèles à partir d'ensembles de données complexes, permettant la prédiction de trajectoires d'outils optimales avec une grande précision et des coûts de calcul réduits. La recherche vise à développer un modèle d'apprentissage profond qui prédit efficacement la trajectoire d'outil optimale pour le processus SLS, minimisant spécifiquement le gradient thermique moyen [11].

I.4.10 Méthode locale de lissage de trajectoires pour robots hybrides 5 axes

SHI ZiKang présente une méthode pionnière de lissage de trajectoire d'outil locale conçue spécifiquement pour un robot d'usinage hybride à cinq axes.

L'objectif principal de cette recherche est la proposition d'une technique de lissage de trajectoire d'outil continue G3 adaptée pour améliorer les capacités d'un robot hybride engagé dans des opérations d'usinage à cinq axes. La clé de la méthodologie réside dans l'incorporation stratégique de B-splines dans le système de coordonnées machine (SCM), placées stratégiquement aux coins pour faciliter les transitions synchronisées à cinq axes.

Pour traiter les erreurs de transition dans les chemins de position et d'orientation de l'outil, l'étude emploie la méthode de section dorée pour une estimation précise. De plus, l'article décrit une méthode de modification adaptative des B-splines, impliquant l'ajout de points d'ancrage et l'optimisation des points de contrôle, contraignant efficacement les erreurs d'approximation.

Une contribution significative de cette recherche est l'introduction d'une méthode de recherche par bisection pour les modifications géométriques, garantissant une adhérence stricte aux limites de tolérance d'erreur définies par l'utilisateur. Le cadre proposé présente des

performances supérieures, générant une trajectoire plus lisse comparée aux méthodes conventionnelles basées sur le système de coordonnées pièce (SCP) sous les mêmes contraintes de tolérance d'erreur.

Pour valider l'efficacité de la méthode proposée, l'article inclut des simulations complètes et des résultats expérimentaux. Ces résultats soulignent non seulement l'applicabilité pratique de l'approche de lissage de trajectoire d'outil développée, mais fournissent également des perspectives précieuses sur son potentiel à élever la précision globale et l'efficacité des robots d'usinage hybride à cinq axes [12].

La figure I.8 présente le résultat de la contrainte d'erreur du trajet d'outil de test et les erreurs de lissage de position [12].

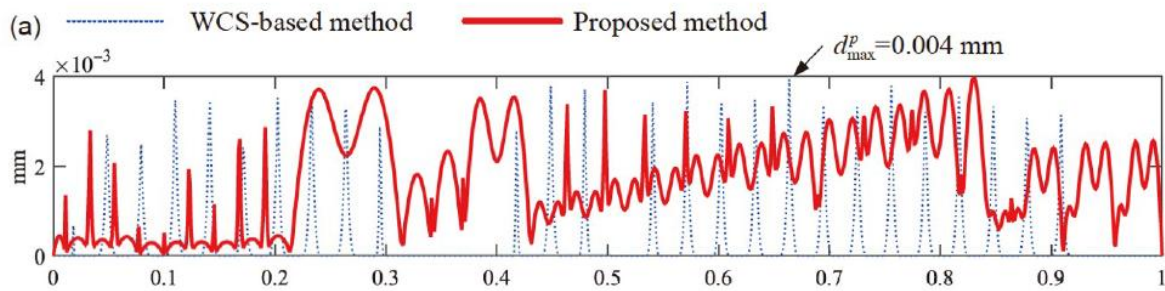


Fig. I. 8 Résultat de la contrainte d'erreur du trajet d'outil de test, Les erreurs de lissage de position [12].

I.4.11 Optimisation du programme de commande numérique et simulation d'usinage basée sur VERICUT

Cet article est lié à l'optimisation du programme de commande numérique et à la simulation d'usinage basée sur VERICUT. Il se concentre sur la génération de trajectoires d'outils optimisées, la compilation et la vérification du programme de commande numérique, et la simulation du processus d'usinage pour améliorer l'efficacité d'usinage et garantir l'exactitude du programme CN. L'article prend spécifiquement l'exemple de l'usinage du châssis de bogie d'une locomotive et utilise l'algorithme de programmation dynamique, le logiciel Creo, et le logiciel VERICUT pour l'optimisation et la simulation. La méthode vise à économiser le temps d'usinage, optimiser les paramètres de coupe, inspecter la qualité d'usinage, et poser les bases pour l'intégration de plateformes similaires.

La plateforme de simulation virtuelle créée par les logiciels Creo et VERICUT est utilisée pour l'établissement de machines, la simulation de structure, l'optimisation des paramètres de coupe,

et l'inspection qualité. L'article souligne l'importance de l'économie de ressources, de la protection environnementale, et de l'utilisation de la technologie de simulation virtuelle pour améliorer l'efficacité de traitement et réduire les coûts [13].

I.4.12 Nouvelle méthode de correspondance de courbure et sélection optimale de l'outil pour l'usinage 5 axes

Than Lin présente une méthode précise et efficace pour l'usinage CN cinq axes de surfaces de forme libre, utilisant un outil de fraisage à bout plat. L'algorithme proposé sélectionne de manière autonome l'outil optimal et planifie la trajectoire d'outil grâce à une combinaison d'appariement de courbure et de cinématique inverse intégrée de la machine-outil. Notamment, l'algorithme utilise la trajectoire réelle de contact d'outil générée par cinématique inverse, s'écartant de la trajectoire conventionnelle linéarisée par segments de localisation réelle de l'outil.

La détermination de l'angle d'inclinaison minimal de l'outil est un aspect clé, empêchant stratégiquement le gougeage sur le plan YL-ZL. L'article comprend une revue complète de la littérature pertinente, expose le nouvel algorithme, et élucide la méthodologie à travers un exemple pratique. De plus, l'article fournit un aperçu des paramètres de configuration d'entrée et du logiciel de simulation utilisé pour la simulation d'usinage, contribuant à une compréhension holistique de la méthode proposée [14].

La figure I.9 illustre la machine virtuelle utilisée dans Vericut® .

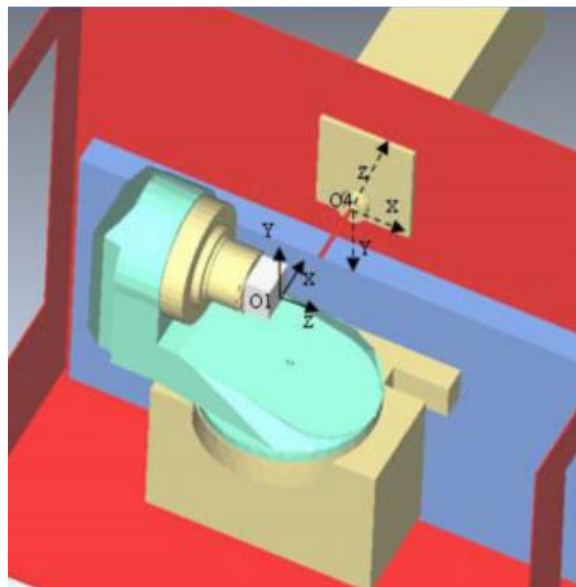


Fig. I. 9 Machine virtuelle dans Vericut® [14].

I.4.13 Algorithme d'optimisation pour l'usinage de cames spatiales basé sur la surface NURBS

L'étude aborde deux problèmes majeurs dans le fraisage de cames spatiales : les erreurs d'approximation lors de la discrétisation des points de données et les fluctuations significatives de la trajectoire de l'outil lors du traitement des surfaces réglées. Pour résoudre ces problèmes, les auteurs proposent un algorithme utilisant des surfaces NURBS, capables de modéliser avec précision des formes complexes. Basé sur la surface théorique de l'axe de la fraise dérivée du profil de la came spatiale, cet algorithme simplifie le processus de conception. La méthode décrit la génération d'une grille d'échantillonnage de données adaptative à la courbure de la surface, sélectionnant des points selon le degré de courbure maximal. Les auteurs construisent ensuite une surface réglée NURBS à partir des points de données valides, affinée par une méthode des moindres carrés pour améliorer la précision de l'usinage. L'efficacité de l'algorithme est validée par des calculs numériques et des simulations, démontrant son applicabilité pratique à l'amélioration de la précision de l'usinage CN. En somme, l'étude présente une solution innovante pour améliorer la précision et l'efficacité des processus de fraisage de cames spatiales [15].

La figure I.10 montre la came cylindrique balayée.

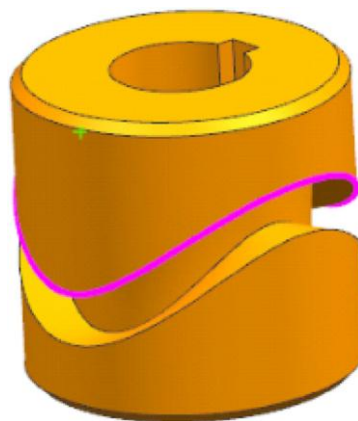


Fig. I. 10 Came cylindrique balayée [15].

I.4.14 Application d'algorithmes génétiques à l'usinage brut d'une hémisphère

L'étude se concentre sur l'optimisation de l'usinage brut CN d'une hémisphère, avec pour objectif principal de développer une méthode qui minimise le temps d'usinage tout en

maximisant l'enlèvement de matière du bloc de matériau d'origine. Cette optimisation est essentielle dans les processus de fabrication où l'efficacité et la conservation des matériaux sont importantes. La méthodologie employée utilise un algorithme génétique (GA) comme outil d'optimisation, illustré par une étude de cas sur l'application des algorithmes génétiques pour résoudre des problèmes complexes d'usinage. L'algorithme génétique permet d'explorer diverses solutions afin de trouver la stratégie d'usinage la plus efficace. Trois paramètres critiques sont identifiés dans le processus d'optimisation : le nombre de festons, qui se réfère aux segments de la surface usinée, la hauteur de chaque feston, qui affecte la qualité et l'efficacité du processus d'usinage, et les outils utilisés, incluant leurs diamètres, les profondeurs de coupe maximales et les vitesses d'avance maximales, tous définis dans une base de données d'outils.

L'étude introduit des fonctions de fitness qui peuvent être utilisées indépendamment ou en combinaison pour évaluer le temps de coupe et la matière restante après l'usinage, éléments essentiels pour déterminer l'efficacité du processus d'usinage.

Les résultats de la méthode proposée se concrétisent dans un tableau récapitulatif détaillant la séquence des outils employés et les hauteurs de feston correspondantes, indispensable pour les applications pratiques en usinage CN. En résumé, l'étude vise à améliorer l'efficacité de l'usinage CN grâce à un algorithme génétique, en se concentrant sur des paramètres clés et des résultats pertinents pour le domaine de la fabrication [16].

I.4.15 Génération intelligente de trajectoires pour le fraisage à bout sphérique 5 axes en utilisant des modèles de processus

Cette étude aborde l'usinage des surfaces sculptées (SSM) en utilisant le fraisage 5 axes, une méthode largement employée dans diverses industries. L'objectif du dégrossissage est de retirer rapidement le matériau, tandis que la finition vise à obtenir une qualité supérieure. Les variations des forces de coupe pendant le dégrossissage peuvent affecter la qualité de la finition, nécessitant parfois des passes de semi-finition pour préparer la surface. La génération de trajectoires d'outil et le choix des paramètres de processus sont essentiels pour l'efficacité de l'usinage. Cependant, les systèmes CAM limitent souvent l'application des paramètres de fraisage optimaux. Bien que des algorithmes efficaces pour la génération de trajectoires aient été développés pour diverses géométries, ces algorithmes négligent souvent la physique de la coupe des métaux, en se concentrant uniquement sur la géométrie.

L'intégration de modèles de processus avec les systèmes CAM peut améliorer les résultats manufacturiers.

Les avancées récentes dans la modélisation des processus de fraisage ont attiré une attention significative, avec diverses études proposant des modèles pour les forces de coupe et la stabilité. Les schémas de trajectoires d'outil influencent fortement l'efficacité et la qualité du processus de fraisage, et diverses approches d'optimisation ont été explorées pour améliorer les opérations de fraisage.

La planification et la génération des trajectoires d'outil pour le SSM 5 axes prennent en compte la profondeur de coupe totale, qui définit la relation entre le dégrossissage, la semi-finition et la finition. La profondeur de coupe et le pas de coupe sont cruciaux pour répartir la charge de copeau à travers les étapes de fraisage. Souvent, des paramètres conservateurs sont sélectionnés basés sur des limites minimales de stabilité. Le dégrossissage peut utiliser le fraisage 2½ axes ou 5 axes selon la géométrie de la pièce, tandis que la semi-finition prépare la surface pour la finition, impactant les forces de coupe. Il est essentiel de maintenir l'uniformité lors de la finition pour la qualité de la pièce.

La génération de trajectoires d'outil varie entre les approches de fraisage 2½ axes et 5 axes, avec des fluctuations significatives de la charge de copeau pouvant survenir entre les étapes de fraisage successives. Une trajectoire d'outil intelligente est nécessaire pour améliorer la performance du processus, et les modifications basées sur des simulations de processus peuvent identifier des améliorations potentielles des paramètres de processus. Les simulations permettent de mettre à jour la position et l'orientation de l'outil tout au long de la trajectoire, influençant les forces de coupe et les limites de stabilité.

Dans l'usinage des cames spatiales, les forces de coupe différentielles sont calculées en fonction de l'épaisseur locale du copeau, les forces étant additionnées le long de l'axe de l'outil pour chaque position de fraise. La prédiction des limites de stabilité est cruciale pour la qualité et la productivité, avec des diagrammes de stabilité simulés pour relier les vitesses de rotation aux fréquences de chatter. L'extraction précise des informations de surface après usinage permet des modifications locales de la trajectoire d'outil.

Les études de cas démontrent l'application des méthodes d'optimisation proposées : l'optimisation de la trajectoire d'outil en semi-finition pour améliorer l'uniformité des forces de coupe, l'optimisation basée sur la hauteur de feston pour réduire le temps de cycle en semi-finition, et le dégrossissage multi-niveaux utilisant des profondeurs de coupe variables pour

réduire le temps de cycle. En conclusion, les simulations de processus améliorent la planification et la modification des trajectoires d'outil, contribuant ainsi à une génération intelligente des trajectoires d'outil en SSM 5 axes [17].

I.5 Méthodes et algorithmes existants

Pour tirer pleinement parti des capacités offertes par l'usinage en 5 axes, l'optimisation des trajectoires d'outil est cruciale. De nombreuses méthodes et algorithmes ont été élaborés pour créer et perfectionner ces trajectoires, chacune apportant des solutions innovantes et spécifiques aux divers défis techniques inhérents à l'usinage en 5 axes.

Les principales méthodes d'optimisation utilisées :

- **Méthodes basées sur les gradients** utilisant des algorithmes d'optimisation pour minimiser une fonction de coût liée à la trajectoire [18] ;
- **Algorithmes heuristiques** utilisant des règles empiriques pour trouver des solutions satisfaisantes, souvent plus rapides mais moins optimales [19] ;
- **Techniques stochastiques** utilisant des éléments de hasard pour explorer l'espace des solutions possibles, comme les algorithmes génétiques [20] ;
- **Méthodes de machine learning** utilisant des modèles d'apprentissage automatique pour prédire et optimiser les trajectoires [21] ;
- **Splines cubiques** utilisant des courbes de Bézier pour générer des trajectoires lisses et précises [22] ;
- **Méthodes de voxelisation** utilisant des modèles volumétriques pour éviter les collisions et optimiser les trajectoires [23] ;
- **Méthodes de surface d'usinage** utilisant des représentations surfaciques pour générer des trajectoires en fonction de la géométrie de la pièce [24] ;
- **Algorithmes de planification de mouvement** planifiant les mouvements de l'outil pour éviter les collisions et optimiser le temps de cycle [25] ;
- **Méthodes d'optimisation multi-objective** optimisant plusieurs critères simultanément, comme la précision, le temps de cycle et l'usure des outils [26] ;

I.6 Applications sectorielles des systèmes multi-axes

L'utilisation des machines multi-axes a révolutionné l'industrie manufacturière, offrant une flexibilité et une précision inégalées dans la fabrication de pièces complexes.

Ces machines sont devenues indispensables dans de nombreux secteurs industriels. Elles réalisent des opérations d'usinage avancées et de répondre aux exigences croissantes de qualité et d'efficacité [27].

I.6.1 Secteur de l'énergie

Dans un contexte énergétique en constante évolution, la fabrication de composants complexes exige une précision et une efficacité accrues. L'usinage CN, notamment multiaxes, s'impose comme une solution incontournable pour répondre aux exigences de qualité, de rentabilité et de fiabilité, tant dans les secteurs des énergies renouvelables que non renouvelables. Il permet la production de pièces hautement résistantes, telles que celles utilisées dans les générateurs, échangeurs de chaleur ou réacteurs, capables de supporter des températures extrêmes et la corrosion, tout en assurant une géométrie précise et une excellente finition de surface, indispensables pour optimiser les performances thermiques [27].

La figure I.11 illustre un composant de rotor haute précision fabriqué par usinage CN multi-axes.



Fig. I. 11 Composant de rotor haute précision fabriqué par usinage CN multi-axes [27].

I.6.2 Automobile

L'usinage à axes multiples est devenu indispensable dans l'industrie automobile pour la fabrication de pièces critiques comme les culasses, blocs moteurs, arbres à cames ou carters de boîtes de vitesses. Ces composants exigent des tolérances très serrées et des géométries complexes, difficiles à obtenir avec des procédés traditionnels. Grâce à l'ajout d'axes supplémentaires, les machines multiaxes permettent d'usiner plusieurs faces d'une pièce en un seul serrage, ce qui améliore la précision, augmente la productivité et réduit les erreurs d'alignement. Cette technologie assure une précision allant jusqu'à 0,001 pouce, essentielle pour garantir les performances et la longévité des moteurs dans la production automobile en série [27].

La figure I.12 présente un composant moteur (culasse) fabriqué par usinage CN à cinq axes.



Fig. I. 12 Composant moteur (culasse) fabriqué par usinage CN 5 axes [27].

I.6.3 Fabrication d'outils et de matrices

La filière de l'outillage et de la fabrication de moules exploite pleinement les potentialités offertes par l'usinage multi-axes pour réaliser des moules et des modèles compliqués à dimensions complexes. En effet, les moules réalisés intègrent souvent des cavités profondes, des formes contournées et des surfaces de forme se rendant en plus compatibles avec des tolérances de très haute précision [27].

La figure I.13 montre un moule industriel en acier trempé réalisé par usinage CN multi-axes.

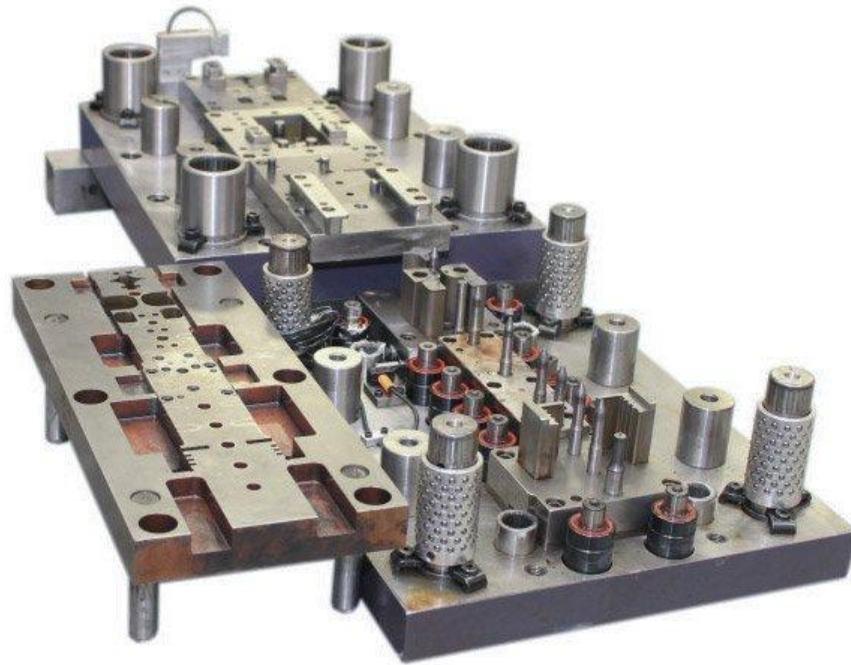


Fig. I. 13 Moule industriel en acier trempé fabriqué par usinage CN multi-axes [27].

I.6.4 Secteur Médical

L'usinage multi-axes joue un rôle essentiel dans les secteurs médical et pharmaceutique. Il s'agit notamment d'instruments orthopédiques aux contours spécifiques au patient, de vis osseuses en titane avec des modèles de filetage spéciaux et de composants dentaires de précision. La précision de ces composants réduit les complications post-chirurgicales et améliore également l'intégration des tissus [27].

I.6.5 Industrie aérospatiale

L'usinage multi-axes est essentiel pour la fabrication de diverses pièces dans l'industrie aérospatiale. Cette technologie produit des pièces de train d'atterrissage, des pales de turbine et des longerons d'aile avec une précision exceptionnelle. Elle est idéale pour le traitement des superalliages résistants à la chaleur et offre également la précision aérodynamique nécessaire aux composants qui fonctionnent dans des conditions difficiles [27].

La figure I.14 illustre l'application de l'usinage multi-axes dans la fabrication aéronautique.



Fig. I. 14 Application de l'usinage multi-axes dans la fabrication de turbines aéronautiques [27].

I.7 Justification de l'étude

L'étude des machines à 5 axes et de leurs applications industrielles est d'une importance capitale en raison de leur impact significatif sur la production manufacturière moderne. Voici plusieurs raisons clés pour lesquelles cette recherche est justifiée :

- **Révolution Technologique :**

Les machines à 5 axes représentent une avancée technologique majeure dans le domaine de l'usinage et de la fabrication. Comprendre leur fonctionnement, leurs avantages et leurs limitations est essentiel pour rester à la pointe de l'innovation technologique.

- **Amélioration de la Précision et de l'Efficacité :**

Ces machines permettent d'atteindre des niveaux de précision et d'efficacité inégalés, réduisant les marges d'erreur et les temps de production. Une étude approfondie aide à optimiser leur utilisation et à maximiser leur potentiel.

- **Réduction des Coûts de Production :**

En augmentant l'efficacité et en réduisant les temps de cycle, les machines à 5 axes peuvent considérablement diminuer les coûts de production. Justifier cette étude repose sur la compréhension des mécanismes par lesquels ces économies sont réalisées.

- **Applications Diversifiées :**

Les machines à 5 axes sont utilisées dans de nombreux secteurs, notamment l'aéronautique, l'automobile, et la fabrication de dispositifs médicaux. Chaque secteur présente des exigences spécifiques et des défis uniques que l'étude peut aider à surmonter.

- **Défis Techniques et Opportunités de Recherche :**

Malgré leurs avantages, les machines à 5 axes posent des défis techniques, tels que la complexité de programmation et la maintenance. Cette étude est justifiée par le besoin d'explorer et de proposer des solutions innovantes pour ces défis.

- **Conformité aux Normes et Règlementations :**

L'industrie manufacturière est régie par des normes strictes et des règlements de qualité. L'étude des machines à 5 axes contribue à garantir que les processus de production respectent ces exigences et à identifier les domaines nécessitant des améliorations.

- **Impact Environnemental :**

L'optimisation des processus de production grâce aux machines à 5 axes peut également réduire l'empreinte écologique de la fabrication. Justifier cette étude inclut l'évaluation des bénéfices environnementaux potentiels.

- **Formation et Compétences :**

La complexité des machines à 5 axes nécessite une formation spécialisée pour les opérateurs. Une étude détaillée peut identifier les besoins en formation et proposer des programmes éducatifs adaptés.

En conclusion, l'étude des machines à 5 axes est essentielle pour comprendre et exploiter pleinement leur potentiel dans l'industrie manufacturière. Elle offre des perspectives précieuses pour améliorer la production, réduire les coûts, et relever les défis techniques, tout en assurant une conformité aux normes et une réduction de l'impact environnemental.

Chapitre II

Trajectoire de l'outil

pour MOCN

Chapitre II Trajectoire de l'outil pour MOCN

II.1 Usinage multiaxes

L'usinage multiaxe est une technologie avancée qui permet de déplacer simultanément un outil de coupe selon plusieurs axes (généralement trois à cinq), offrant ainsi une flexibilité et une précision accrues dans la fabrication de pièces complexes. Cette technique est particulièrement prisée dans les secteurs de l'aéronautique, de l'automobile et du médical, où les géométries libres, les tolérances serrées et les finitions de surface exigeantes sont courantes. Contrairement à l'usinage traditionnel à 3 axes, l'usinage multiaxe réduit les opérations de repositionnement et permet de traiter des zones difficilement accessibles, tout en améliorant la productivité globale. De plus, il favorise une usure plus homogène de l'outil et une meilleure gestion des efforts de coupe [28].

II.2 Système d'axes cartésiens

Le système d'axes cartésien est la base normalisée de la programmation CN, articulant selon une rigoureuse configuration géométrique¹. Il se définit au sens de la norme ISO 841:2001 fondamentalement à partir de trois éléments [29]:

- un point d'origine (O) permettant d'en faire la référence absolue de toutes les coordonnées ;
- Trois axes orthogonaux en 3D (X,Y,Z) ou deux axes en 2D (X,Y) qui forment un repère droit au sens de la géométrie euclidienne ;

Les conventions d'orientation sont généralement les suivantes :

- l'axe X est associé au mouvement longitudinal (gauche-droite) de la table ; il représente le plus grand axe ;
- L'axe Y est considéré comme étant le déplacement transversal (avant-arrière) à la pièce ;
- L'axe Z représente le déplacement vertical associé au mouvement de pénétration de l'outil dans la pièce ;

II.3 Système d'axes supplémentaires

la norme AFNOR NF Z 60-020 a pour objet de définir une nomenclature des axes et des mouvements pour machines à commande numérique, afin de faciliter l'interchangeabilité des données de programmation. Dans ce contexte, les machines CNC haut de gamme intègrent des

degrés de liberté complexes pour répondre aux exigences de fabrication additive et soustractive. On distingue deux catégories [30] :

- **Axes rotatifs (A, B, C)**

- Rotation A autour de l'axes X ;
- Rotation B autour de l'axes Y ;
- Rotation C autour de l'axes Z ;

Ces rotatifs (A, B, C) permettent l'usinage de surfaces courbes sans repositionnement manuel.

- **Axes linéaires secondaires (U, V, W)**

Parallèles aux axes primaires, ils autorisent des déplacements synchronisés pour des opérations multi-outils ou des corrections dynamiques.

La figure II.1 illustre les axes primaires et additionnels utilisés dans les systèmes CN.

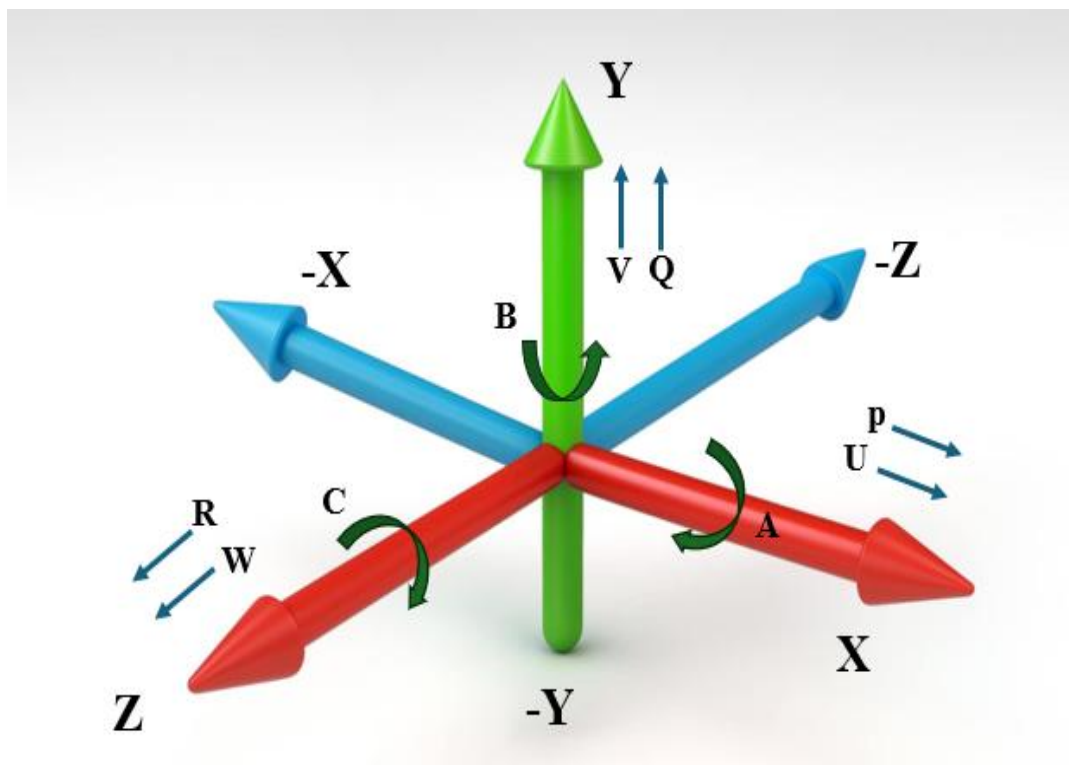


Fig. II. 1 Axes primaires et axes additionnels.

Les directions positives des axes de rotation A, B et C sont définies en sens inverse du mouvement trigonométrique classique, lorsque l'on observe selon l'orientation positive de l'axe linéaire associé.

Une autre manière d'interpréter ces directions est la suivante :

- En se basant sur un trièdre de référence, la rotation positive autour de l'axe Z correspond à un mouvement où l'axe X s'oriente vers l'axe Y à l'intérieur du trièdre;
- Si l'on aligne le pouce de la main droite dans la direction positive de l'axe linéaire, les autres doigts légèrement courbés indiquent le sens positif de la rotation autour de cet axe ;

Le tableau II.1 présente les différents mouvements possibles dans un système CN, comprenant les translations, rotations et vecteurs centre arc [30][31].

Tableau. II. 1 Tableau récapitulatif des axes CN.

Translation			Rotation		Vecteurs Centre Arc
Primaire	Secondaire	Tertiaire	Primaire	Secondaire	
X	U	P	A	D	I
Y	V	Q	B	E	J
Z	W	R	C		K

Les vecteurs centre arc (I ;J ;K) jouent un rôle primordial dans l'interpolation circulaire. Bien qu'ils ne soient pas des axes de mouvement à proprement parlé comme (X; Y; Z) car ils représentent des vecteurs, ils ont pour rôle de définir avec précision la géométrie d'un arc de cercle à tracer, ces vecteurs permettent de garantir une précision lors du mouvements circulaires ainsi leur maîtrise est essentielle pour une programmation efficace des trajectoires courbes et les opérations d'usinage complexes [31].

II.4 Transformations géométriques et programmation des axes

II.4.1 Fondements mathématiques des transformations

Les systèmes CN utilisent des transformations homogènes pour convertir les coordonnées entre repères (pièce, outil, machine). Ces opérations s'appuient sur la représentation en coordonnées homogènes de la géométrie projective [32] :

$$T = \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} \quad (\text{II.1})$$

Où :

- $R_{3 \times 3}$ est la matrice de **rotation** (exemple : rotation autour de X, Y ou Z) ;
- $t_{3 \times 1}$ est le vecteur de **translation** (exemple : décalage de l'origine) ;
- La dernière ligne ($0_{1 \times 3} \ 1$) assure que la transformation est homogène ;

II.4.2 Transformations de mouvement

Les transformations de mouvement permettent de **convertir les coordonnées** entre le repère machine (fixe), le repère outil (mobile) et le repère pièce (peut être déplacé) [33].

II.4.3 Types de transformations de mouvement

Il existe quatre matrices de transformation fondamentales sur lesquelles la cinématique à 5 axes peut être basée [33] :

- **La translation (déplacement linéaire)**

C'est un déplacement **sans rotation**, comme une glissière qui coulisse.
Exemple : déplacer une pièce de **50 mm vers le haut (axe Z)**.

$$\text{Formule : } T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II. 2})$$

où t_x, t_y, t_z sont les décalages selon X, Y et Z.

- **Rotation (mouvement angulaire)**

C'est une transformation où l'objet tourne autour d'un axe, ce qui peut être représentée par trois matrices de rotation selon les axes principaux..

Exemple : rotation de 45° autour de X.

Matrice de rotation autour de X :

$$R(X, \theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II. 3})$$

- Les coordonnées en **X** restent inchangées;
- Les coordonnées en **Y et Z** tournent selon l'angle θ ;

La matrice de rotation autour de Y est :

$$R(Y, \theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II. 4})$$

- Les coordonnées en **Y** restent inchangées;
- Les coordonnées en **X et Z** tournent selon l'angle θ ;

La matrice de rotation autour de Z est :

$$R(Z, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{II. 5})$$

- Les coordonnées en **Z** restent inchangées ;
- Les coordonnées en **X et Y** tournent selon l'angle θ ;

II.5 Langage de programmation

Le langage ISO peut être saisi directement depuis le pupitre de la machine à commande numérique (CN), mais cette approche monopolise la machine pendant la phase de programmation, ce qui réduit son temps de production effectif. La plupart des systèmes actuels proposent donc une interface, souvent peu conviviale, permettant la programmation et le téléchargement du programme vers la CN [34].

Le poste de programmation peut être une console spécialisée qui reproduit l'interface utilisateur de la CN, ou un ordinateur standard équipé d'un logiciel dédié ou d'un simple éditeur de texte. L'écriture du programme peut se faire en temps masqué, c'est-à-dire sans arrêter la machine, mais les tests syntaxiques et la simulation graphique nécessitent l'utilisation de la MOCN elle-même ou bien l'emploi d'un logiciel de simulation [34].

Le programme doit inclure toutes les données nécessaires à la commande et au séquençage des opérations d'usinage. Ces données comprennent les commandes proprement dites, les données géométriques, qui définissent la forme, les dimensions de la pièce à usiner et permettent à la CN de calculer les trajectoires de l'outil par rapport à la pièce, en se basant sur une origine connue ; et les instructions associées à ces données, comme le mode d'interpolation, le choix du mode de cotation, le cycle d'usinage ou l'outil à utiliser [34].

À cela s'ajoutent les données technologiques, qui tiennent compte des caractéristiques et des performances de la machine (puissance des moteurs, capacité de la broche, déplacements des axes, etc.) pour définir les conditions de coupe optimales, notamment la vitesse de rotation de la broche, les vitesses d'avance et la gestion de l'arrosage [34].

II.5.1 Langage Fanuc

Le langage de programmation Fanuc CN est principalement basé sur la norme ISO établie dans les années 1980, mais il introduit ses propres raffinements structurels et fonctionnels qui sont largement adoptés dans les applications industrielles. L'une de ses caractéristiques les plus remarquables est l'utilisation de parenthèses pour insérer des commentaires directement à l'intérieur des blocs de code. Cela permet aux programmeurs d'ajouter des annotations et de clarifier les intentions du programme sans affecter l'exécution [35].

Il ajoute aussi des fonctions supplémentaires ce qui en fait un langage unique. Les spécificités du langage sont [35] :

- Appel de sous programmes avec M98 ;
- Points virgules en fin de blocs ;

II.5.2 Langage NUM

Le langage NUM prend pour base le langage ISO. Il ajoute des fonctions supplémentaires ce qui en fait un langage unique. Les spécificités du langage sont - Parenthèses pour les commentaires ; - Appel de sous programmes avec G77 [36].

II.5.3 Langage SIEMENS

Le langage SIEMENS prend pour base le langage ISO. Il ajoute des fonctions supplémentaires ce qui en fait un langage unique. Les spécificités du langage sont [37]:

- Points virgules pour les commentaires ;
- Des appels de cycles relativement complexes ;

II.5.4 Langage HEIDENHAIN

Le système HEIDENHAIN, crée en Allemagne par la société HEIDENHAIN, a été conçu pour permettre la programmation intuitive directement en atelier. Le programmeur peut alors rédiger ses programmes d'usinage sans avoir connaissance approfondie sur le codes ISO classiques, Grâce à une interface graphique associée à un langage structuré en texte clair. la programmation des opérations de perçage et de fraisage devient plus accessible et visuellement assistée [38].

II.5.5 Langage PROFORM

Le langage PROFORM a été inventé de toute pièce pour les robots charmillé. Langage devenu totalement obsolète [39].

II.5.6 Langage ISO

À l'origine, le langage de programmation était le G-code, développé par l'EIA (Association des industries électroniques) au début des années 1960, et finalement normalisé par l'ISO en 1980 sous la référence RS274D/ (ISO 6983). Le langage ISO est encore énormément répandu surtout sur les petites Machines commandes numériques [40].

II.6 Langage G-Code et contrôle des axes

Le standard ISO 6983 définit les commandes de base dite le code G dont il est utilisé pour définir la nature fonctionnelle de la courbe, c'est-à-dire linéaire, circulaire ou parabolique [40].

Le tableau II.2 illustre l'explication des codes utilisés dans la programmation CN, notamment les codes G, M, S et T [41].

Tableau. II. 2 Explication des Codes G M S T [41].

Code	Description
G00	Déplacement rapide
G01	Interpolation linéaire
G02	Interpolation circulaire sens horaire (CW)
G03	Interpolation circulaire sens antihoraire (CCW)
G40	Annulation de la compensation de l'outil
G41	Compensation du rayon d'outil à gauche
G42	Compensation du rayon d'outil à Droite
G43	Compensation de la longueur d'outil + direction
G49	Annulation de la compensation de la longueur d'outil
G90	Programmation absolue
G91	Programmation incrémentale ou relative
M03	Démarrage de la broche en sens horaire
M06	Changement d'outil
M30	Fin du programme (avec réinitialisation)
M98	Appel de sous-programme
M99	Fin de sous-programme
T	Outil (tool)
S	Vitesse de rotation (speed)
F	Avance (feed)

La figure II.2 présente un exemple d'interpolation circulaire (G02/G03).

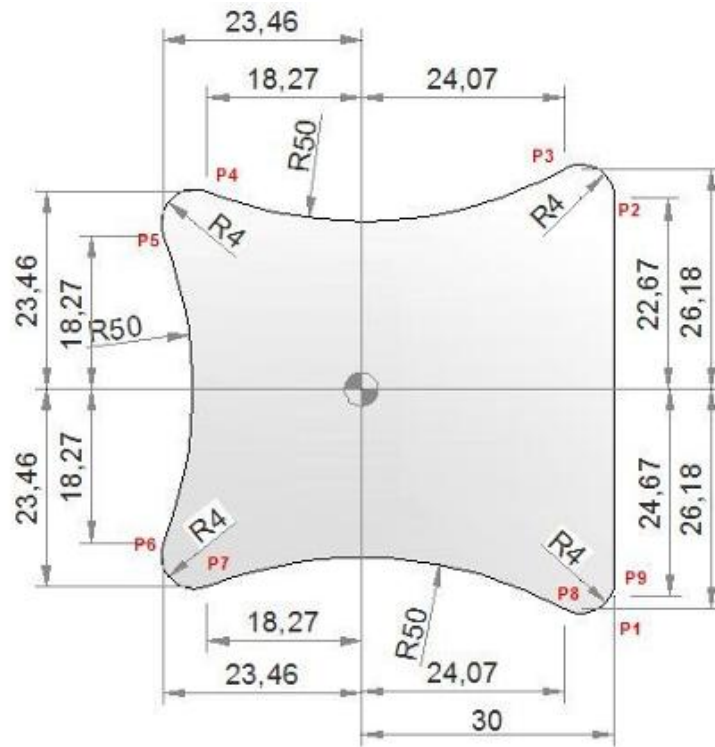


Fig. II. 2 Exemple de profil d'interpolation circulaire (G02/G03) [41] .

Démonstration de l'interpolation circulaire G02/G03 [41]:

```
CNC Part Program
G0 X30 Y-30 (P1)
G1 Y22.67 (P2)
G3 X24.07 Y26.18 R4 (P3)
G2 X-18.27 Y23.46 R50 (P4)
G3 X-23.46 Y18.27 R4(P5)
G2 X-23.46 Y-18.27 R50 (P6)
G3 X-18.27 Y-23.46 R4 (P7)
G2 X24.07 Y-26.18 R50 (P8)
G3 X30 Y-24.67 R4 (P9)
G1 X33
```

II.7 Interpolation des trajectoires

L'interpolation numérique joue un rôle fondamental dans de nombreux domaines scientifiques et techniques, notamment dans la modélisation, l'analyse de données et le contrôle des systèmes. Parmi les méthodes les plus couramment utilisées, on retrouve les splines cubiques, l'interpolation de Lagrange, la méthode PCHIP (Piecewise Cubic Hermite Interpolating Polynomial), et les polynômes de Chebyshev.

II.7.1 Splines cubiques

L'interpolation spline cubique est une méthode puissante utilisée pour approximer une fonction à partir de données discrètes, en garantissant la continuité non seulement de la fonction elle-même, mais aussi de ses premières et deuxièmes dérivées. Contrairement aux interpolations polynomiales classiques, qui peuvent induire des oscillations importantes, notamment aux bords du domaine, l'interpolation par spline cubique divise l'intervalle global en sous-intervalles et ajuste un polynôme de degré trois à chaque segment. Cette approche permet une transition fluide et naturelle entre les points, tout en assurant une bonne stabilité numérique. Elle est largement utilisée dans les domaines du traitement du signal, de la modélisation géométrique et de la robotique pour le lissage des trajectoires ou la reconstitution de courbes complexes [42].

II.7.2 Interpolation de Lagrange

L'interpolation de Lagrange permet de construire un polynôme unique passant par un ensemble de points donnés, en s'appuyant sur une combinaison linéaire de polynômes de base associés à chaque point. Cette méthode est appréciée pour sa simplicité et sa formulation explicite. Toutefois, elle devient moins efficace lorsque le nombre de points augmente, car elle peut engendrer des oscillations importantes aux extrémités du domaine, un phénomène connu sous le nom d'effet de Runge. Malgré cette limitation, elle demeure un outil fondamental en analyse numérique et en approximation fonctionnelle [43].

II.7.3 PCHIP

L'interpolation est une méthode fondamentale pour estimer les valeurs d'une fonction à partir d'un ensemble discret de données. Parmi les différentes techniques disponibles, l'interpolation cubique par morceaux dite PCHIP (*Piecewise Cubic Hermite Interpolating Polynomial*) se distingue par sa capacité à préserver la forme des données tout en assurant une transition fluide entre les points [44].

Contrairement aux interpolations polynomiales globales, souvent sujettes à des oscillations importantes entre les points, notamment aux extrémités, la méthode PCHIP a été spécifiquement conçue pour éviter ces artefacts visuels ou numériques. Elle s'appuie sur des polynômes cubiques définis par morceaux, tout en imposant des conditions supplémentaires sur les dérivées afin d'obtenir une courbe visuellement réaliste, stable et fidèle aux variations locales [44].

II.7.4 Polynômes de Chebyshev

Les polynômes de Chebyshev occupent une place centrale en analyse numérique, tant par leur diversité que par la richesse de leurs applications. Ils ne se limitent pas à un seul type de fonction polynomiale : en réalité, il existe quatre familles principales de polynômes de Chebyshev, chacune ayant ses propres propriétés et domaines d'application [45].

La première famille, souvent simplement appelée « polynômes de Chebyshev » sans autre précision, est la plus connue et la plus utilisée. Leur comportement oscille de manière régulière sur un intervalle donné, et leurs propriétés sont intimement liées aux fonctions trigonométriques classiques. À côté de cette famille, on trouve les polynômes de deuxième, troisième et quatrième types, qui enrichissent le cadre théorique et offrent des outils puissants pour traiter des problèmes où les conditions aux bords ou les symétries sont particulières [45].

Ces polynômes sont particulièrement utiles, c'est leur capacité à modéliser des fonctions complexes de manière stable et efficace. Par exemple, les polynômes de Chebyshev sont fondamentaux en approximation polynomiale, notamment dans la construction d'approximation dite « minimax », où l'erreur maximale est réduite de façon optimale sur un intervalle donné. Cette propriété leur confère une utilité exceptionnelle en ingénierie et en sciences appliquées, où l'on cherche souvent à minimiser les erreurs dans les modèles numériques [45].

En plus de l'approximation, les polynômes de Chebyshev interviennent dans les méthodes d'interpolation, dans les techniques d'intégration numérique (comme les quadratures de Gauss-Chebyshev), et dans la résolution d'équations différentielles ordinaires ou aux dérivées partielles via les méthodes spectrales. Leur structure particulière permet également une formulation efficace des séries de Chebyshev, lesquelles jouent un rôle analogue aux séries de Fourier, mais avec des propriétés de convergence souvent plus avantageuses dans les situations pratiques [45].

II.8 Introduction aux méthodes d'optimisation numérique

L'optimisation numérique regroupe un ensemble de méthodes mathématiques permettant de déterminer les valeurs optimales d'un ou plusieurs paramètres d'un système, en fonction d'un critère défini, souvent appelé fonction objectif. Ces techniques sont essentielles dans de nombreux domaines, tels que l'ingénierie, la finance, l'intelligence artificielle ou encore la recherche opérationnelle. Elles permettent de résoudre des problèmes complexes pour lesquels

les solutions analytiques sont difficiles, voire impossibles à obtenir. Selon la nature du problème (linéaire ou non linéaire, continu ou discret, contraint ou non contraint), différentes approches peuvent être mobilisées, telles que les méthodes de gradient, les algorithmes évolutionnaires ou les techniques de programmation mathématique [46].

II.8.1 Algorithmes évolutionnaires

Les algorithmes génétiques sont la traduction du concept biologique d'évolution en recettes algorithmiques. Ils appartiennent au domaine de l'informatique lié aux machines et aux programmes informatiques. Comme ils font partie de nombreux systèmes intelligents, les algorithmes génétiques sont souvent pris en compte dans les domaines de l'intelligence informatique et de l'intelligence artificielle, qui visent à construire des méthodes qui imitent, voire dépassent, l'intelligence humaine. Entre-temps, un grand nombre de méthodes ont été proposées dans ces catégories [46].

Les algorithmes génétiques sont des méthodes de recherche heuristique applicables à un large éventail de problèmes d'optimisation [46].

Cette flexibilité les rend attrayants pour de nombreux problèmes d'optimisation dans la pratique. L'évolution est à la base des algorithmes génétiques [46].

La variété et le succès actuels des espèces sont une bonne raison de croire au pouvoir de l'évolution. Les espèces sont capables de s'adapter à leur environnement. Elles ont développé des structures complexes qui leur permettent de survivre dans différents types d'environnements.

L'accouplement et l'évolution de la progéniture font partie des principes fondamentaux du succès de l'évolution. Ce sont là de bonnes raisons d'adapter les principes de l'évolution à la résolution des problèmes d'optimisation [46].

Les algorithmes génétiques sont d'excellentes méthodes pour les problèmes d'optimisation difficiles, où les méthodes d'optimisation classiques échouent en raison de caractéristiques difficiles. Ces conditions peuvent être l'instabilité, la non-dérivabilité et bien d'autres [46].

II.8.2 Algorithme génétique de base

L'algorithme génétique classique est basé sur un ensemble de solutions candidates qui représentent une solution au problème d'optimisation à résoudre. Une solution est un candidat potentiel pour un optimum du problème d'optimisation. Sa représentation joue un rôle important, car elle détermine le choix des opérateurs génétiques [46].

Les représentations sont généralement des listes de valeurs et sont plus généralement basées sur des ensembles de symboles. S'ils sont continus, ils sont appelés vecteurs, s'ils sont constitués de bits, ils sont appelés chaînes de bits. Dans le cas de problèmes combinatoires, les solutions consistent souvent en des symboles qui apparaissent dans une liste. Un exemple est la représentation d'une tournée dans le cas du problème du voyageur de commerce (TSP, Traveling Salesman Problem). Les opérateurs génétiques produisent de nouvelles solutions dans la représentation choisie et permettent de se déplacer dans l'espace des solutions. Le codage de la solution en tant que représentation, qui est soumise au processus évolutif, est appelé génotype ou chromosome [46].

Au début, un ensemble de solutions, appelé population, est initialisé. Cette initialisation est recommandée pour couvrir de manière aléatoire l'ensemble de l'espace des solutions ou pour modéliser et incorporer des connaissances spécialisées. La représentation détermine le processus d'initialisation. Pour les représentations de chaînes de bits, une combinaison aléatoire de zéros et de uns est raisonnable, par exemple le chromosome aléatoire initial 1001001001 est une chaîne de bits typique d'une longueur de 10 [46].

La boucle générationnelle principale de l'algorithme génétique génère de nouvelles solutions candidates par croisement et mutation jusqu'à ce que la population soit complète.

II.8.3 Crossover

Les opérateurs de croisement des algorithmes génétiques mettent en œuvre un mécanisme qui mélange le matériel génétique des parents. Un opérateur célèbre pour la représentation des chaînes de bits est le croisement à n points. Il sépare deux solutions en n positions et les assemble alternativement en une nouvelle solution (Fig. II. 3). Par exemple, si 0010110010 est le premier parent et 1111010111 le second, le croisement en un point choisit aléatoirement une position, supposons 4, et génère les deux solutions candidates descendantes 0010-010111 et 1111-110010. Cet opérateur est motivé par le fait que les deux chaînes peuvent représenter des parties réussies de solutions qui, lorsqu'elles sont combinées, sont même plus performantes que leurs parents. Cet opérateur peut facilement être étendu à d'autres points, où les solutions sont divisées et réassemblées alternativement [46].

La figure II.3 montre l'illustration du croisement en un point qui divise le génome de deux solutions et les réassemble pour obtenir de nouvelles solutions.

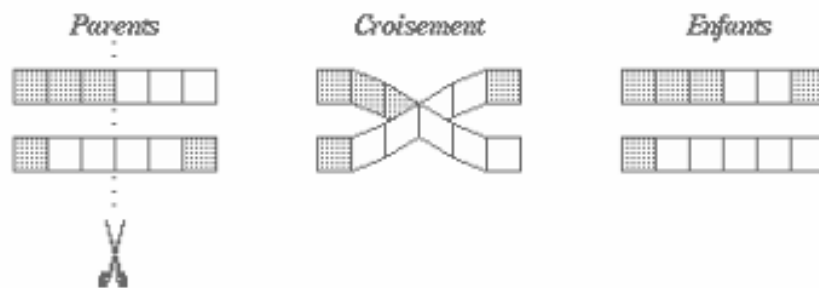


Fig. II. 3 Illustration du croisement en un point qui divise le génome de deux solutions en un point arbitraire et les réassemble pour obtenir deux nouvelles solutions [47].

II.8.4 Mutation

Le deuxième protagoniste des algorithmes génétiques est la mutation. Les opérateurs de mutation modifient une solution en la perturbant. La mutation est basée sur des changements aléatoires. La force de cette perturbation est appelée taux de mutation. Dans les espaces de solution continus, le taux de mutation est également connu sous le nom de taille de pas. Les opérateurs de mutation doivent répondre à trois exigences principales. La première condition est l'accessibilité. Chaque point de l'espace des solutions doit pouvoir être atteint à partir d'un point arbitraire de l'espace des solutions. Un exemple qui peut compliquer le respect de cette condition est l'existence de contraintes qui réduisent l'ensemble de l'espace des solutions à un sous-ensemble réalisable. Il doit y avoir une chance minimale d'atteindre chaque partie de l'espace des solutions [46].

Dans le cas contraire, la probabilité de trouver l'optimum n'est pas positive. Tous les opérateurs de mutation ne peuvent pas garantir cette condition. Par exemple, les approches par décodeur ont des difficultés à couvrir l'ensemble de l'espace de solution [46].

Le deuxième principe de conception des opérateurs de mutation est l'impartialité. L'opérateur de mutation ne doit pas induire une dérive de la recherche dans une direction particulière, du moins dans les espaces de solutions non contraints sans plateaux. Dans le cas d'espaces de solutions contraints, le biais peut être avantageux [46].

Le troisième principe de conception des opérateurs de mutation est l'évolutivité. Chaque opérateur de mutation doit offrir le degré de liberté nécessaire pour que sa force soit adaptable.

Cela est généralement possible pour les opérateurs de mutation basés sur une distribution de probabilités [46].

Par exemple, pour la mutation gaussienne qui est basée sur la distribution gaussienne, l'écart type peut mettre à l'échelle les échantillons tirés au hasard dans l'ensemble de l'espace de solution [46].

La figure II.4 illustre le principe de mutation en codage binaire.

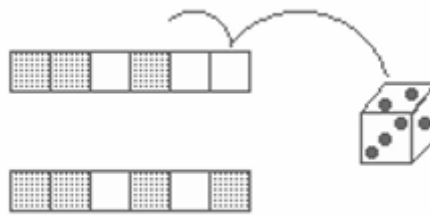


Fig. II. 4 Mutation en codage binaire [47].

II.8.5 Mappage Génotype-Phénotype

Après le croisement et la mutation, la nouvelle population de descendants doit être évaluée. Chaque solution candidate doit être analysée en fonction de sa capacité à résoudre le problème d'optimisation. En fonction de la représentation utilisée, un mappage du chromosome (le génotype) vers la solution réelle (appelée phénotype) peut être nécessaire. Ce mappage génotype-phénotype doit éviter d'introduire un biais. Par exemple, un mappage biaisé pourrait associer la majorité de l'espace des génotypes à un petit ensemble de phénotypes [46].

Le mappage génotype-phénotype n'est pas toujours requis. Par exemple, dans les problèmes d'optimisation continue, le génotype représente directement la solution. Mais dans de nombreux autres processus de modélisation évolutionnaire, ce mappage est indispensable [46].

II.8.6 Fitness

Dans l'étape de calcul de la valeur d'adaptation (fitness), le phénotype d'une solution est évalué à l'aide d'une fonction de fitness. Cette fonction mesure la qualité des solutions générées par l'algorithme génétique. La conception de la fonction de fitness fait partie intégrante du processus de modélisation de l'ensemble de l'approche d'optimisation. Le praticien peut influencer les choix de conception de cette fonction et ainsi orienter la recherche. Par exemple, la valeur de fitness des solutions non réalisables peut être dégradée, comme dans le cas des

fonctions de pénalité. En présence de multiples objectifs à optimiser simultanément, les valeurs de fitness de chaque objectif peuvent être agrégées, par exemple en calculant une somme pondérée. les valeurs de la fonction d'aptitude de chaque objectif peuvent être agrégées, par exemple en calculant la somme pondérée [46].

II.8.7 Sélection

Pour permettre la convergence vers des solutions optimales, les meilleures solutions de la descendance doivent être sélectionnées pour devenir les parents de la nouvelle population parentale. Un surplus de solutions descendantes est généré et les meilleures sont sélectionnées pour progresser vers l'optimum. Ce processus de sélection est basé sur les valeurs d'aptitude de la population. En cas de problèmes de minimisation, les valeurs de fitness faibles sont préférées et vice versa en cas de problèmes de maximisation [46].

Les problèmes de minimisation peuvent facilement être transformés en problèmes de maximisation par la négation. Bien entendu, cela fonctionne également pour transformer les problèmes de maximisation en problèmes de minimisation [46].

II.8.8 Condition d'arrêt

La condition d'arrêt (Termination) définit le moment où la boucle évolutive principale se termine. Souvent, l'algorithme génétique fonctionne pendant un nombre prédéfini de générations. Ce nombre peut être raisonnable dans divers contextes expérimentaux. Le temps et le coût des évaluations de la fonction d'aptitude peuvent limiter la durée du processus d'optimisation. Une autre condition d'arrêt utile est la convergence du processus d'optimisation. Lorsque l'on s'approche de l'optimum, la progression des améliorations de la fonction d'aptitude peut diminuer de manière significative. Si aucun processus significatif n'est observé, le processus d'évolution s'arrête [46].

Chapitre III

Modélisation et optimisation des trajectoires d'outil

Chapitre III Modélisation et optimisation des trajectoires d'outil

III.1 Introduction

Les machines-outils à commande numérique (MOCN) ont révolutionné l'industrie manufacturière en permettant la fabrication automatisée de pièces complexes avec une grande précision. Ces machines utilisent des programmes informatiques pour contrôler les mouvements des outils de coupe et produire des pièces à partir des dessins et des spécifications fournis.

Cependant, malgré tous leurs avantages, les MOCN présentent aussi certaines limites. L'une des principales réside dans leur capacité à générer des formes complexes. Même si elles peuvent produire des pièces très précises, elles peuvent rencontrer des difficultés quand il s'agit de créer des formes géométriques très complexes ou des contours très fins.

III.2 Interpolation

L'interpolation est le processus de calcul des mouvements de l'outil de la machine en fonction des données de conception entrées. Elle est utilisée pour générer des trajectoires d'outils basées sur les points de conception définis par les données de position et de direction de l'outil de la machine. Les interpolateurs utilisés pour les MOCN déterminent les mouvements de l'outil entre les points de trajectoire à l'aide de divers algorithmes d'interpolation tels que Hermite, Bézier, Chebychev ou spline cubique.

Bien que les interpolateurs génèrent des mouvements d'outil entre les points de trajectoire, c'est le contrôleur CN qui détermine comment les axes de la machine doivent être positionnés pour permettre ces mouvements et maintenir la précision d'usinage. Le choix de l'algorithme d'interpolation dépend de la trajectoire spécifique à usiner, de la précision requise et de l'application pour laquelle la machine à commande numérique par ordinateur est utilisée.

III.3 Limites de l'interpolation

Les interpolateurs peuvent présenter certaines limitations lorsqu'ils sont utilisés pour calculer les mouvements d'outils sur des machines CNC. Ces limitations peuvent inclure une faible précision dans les zones de forte courbure, une complexité accrue dans le traitement des paramètres de contrôle et une difficulté à fournir des résultats satisfaisants pour des géométries complexes.

III.4 Algorithme d'optimisation du parcours d'outil

L'algorithme d'optimisation du parcours d'outil (TPOA) calcule l'erreur entre deux interpolateurs différents en utilisant la trajectoire d'outil d'origine comme entrée. Le premier interpolateur calcule une approximation du trajet à l'aide de points prédéfinis, tandis que le deuxième interpolateur utilise une méthode d'interpolation différente pour calculer une approximation différente. Ensuite les deux approximations sont comparées pour calculer l'erreur entre les deux interpolateurs.

Pour créer une trajectoire plus précise, le programme utilise les résultats des deux interpolateurs en combinant les deux ensembles de points. Cette combinaison se fait en trouvant l'ensemble de points le plus proche de la trajectoire d'origine.

La figure III.1 illustre l'algorithme d'optimisation du parcours d'outil (TPOA).

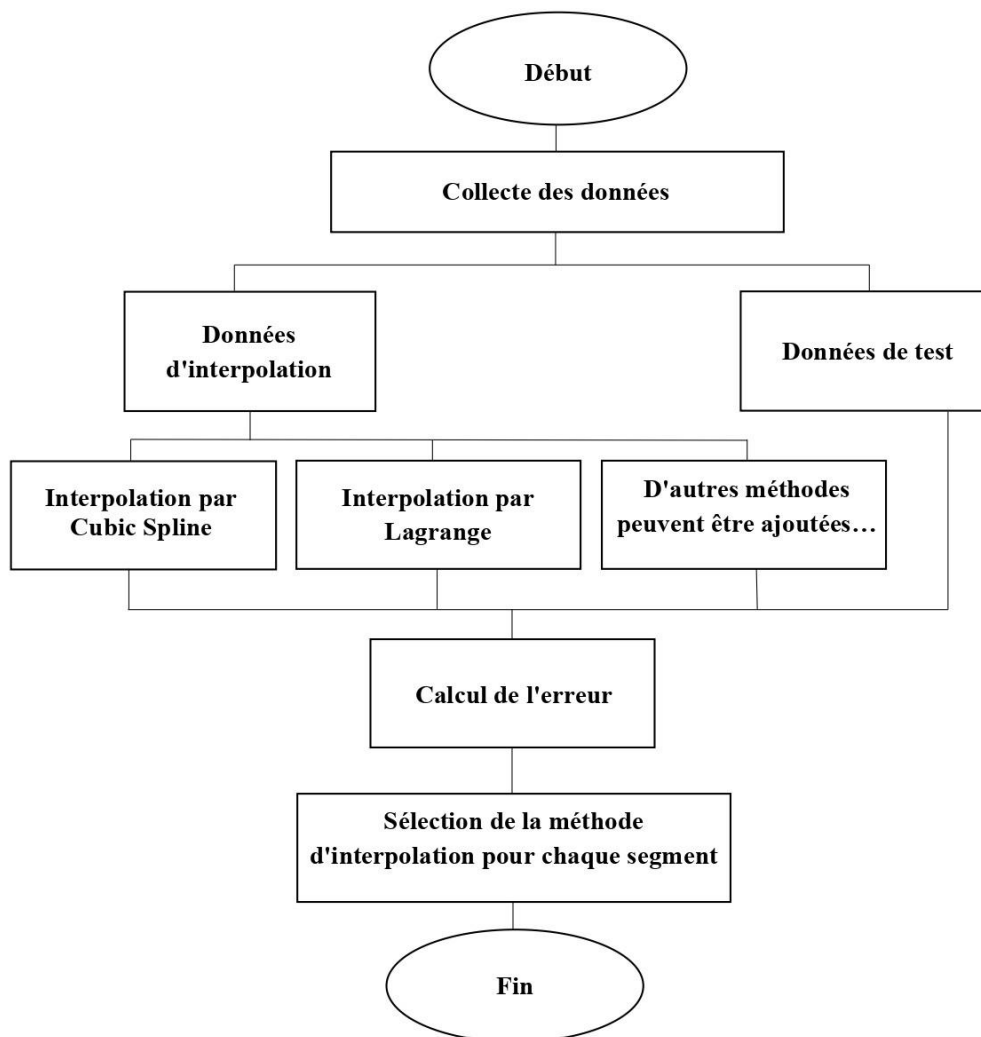


Fig. III. 1 Algorithme d'optimisation du parcours d'outil (TPOA).

III.5 Combinaison d'interpolateurs

Dans les premiers essais, deux méthodes ont été utilisées, la première méthode est l'interpolation spline cubique et la seconde méthode est l'interpolation de Lagrange. Il est important de noter que la méthode TPOA peut utiliser plus de deux méthodes. Les méthodes d'interpolation spline cubique et d'interpolation de Lagrange ont été choisies comme exemples dans cette étude. Cela implique que la méthode TPOA a la flexibilité d'incorporer diverses méthodes d'interpolation en fonction des besoins spécifiques de l'opération d'usinage.

Le concept de combinaison d'interpolateurs dans l'usinage CN concerne l'utilisation de plusieurs algorithmes d'interpolation dans différentes sections d'un parcours d'outil pour obtenir le résultat souhaité. Par exemple, un algorithme peut être mieux adapté aux courbes, tandis qu'un autre peut être meilleur pour les lignes droites. En utilisant une combinaison d'algorithmes, une machine MOCN peut atteindre une plus grande précision et efficacité dans l'usinage de géométries complexes. Cependant, choisir la bonne combinaison d'interpolateurs nécessite une compréhension approfondie de chaque algorithme et de ses limites. Dans l'ensemble, cette technique peut être un outil puissant pour obtenir des résultats d'usinage CN de haute qualité.

La figure III.2 montre la transition entre interpolateurs utilisant la méthode TPOA.

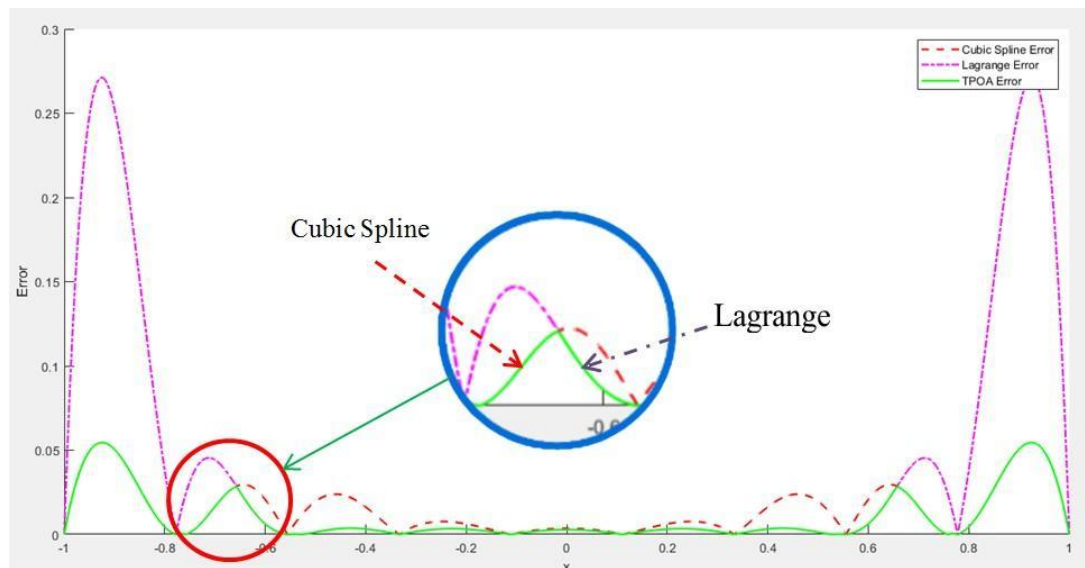


Fig. III. 2 Graphique d'erreur du TPOA par rapport à spline cubique et Lagrange.

III.5.1 Interpolation de spline cubique

Le but de l'interpolation par spline cubique [48] est d'obtenir une formule d'interpolation continue à la fois dans les dérivées premières et secondes, à la fois dans les intervalles et aux

nœuds d'interpolation. Cela fournit une fonction d'interpolation plus fluide. La continuité de la dérivée première signifie que le graphique $y = S(x)$ n'aura pas d'angles vifs. La continuité de la dérivée seconde signifie que le rayon de courbure est défini en chaque point.

Étant donné les n points de données $(x_1, y_1), \dots, (x_n, y_n)$, où x_{je} sont distincts et en ordre croissant. Une spline cubique $S(x)$ passant par les points de données $(x_1, y_1), \dots, (x_n, y_n)$ est un ensemble de polynômes cubiques :

$$S_{n-1}(x) = y_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3 \text{ on } [x_{n-1}, x_n] \quad (\text{III. 1})$$

Avec les conditions suivantes (appelées propriétés) :

$$S_i(x_i) = y_i \text{ and } S_i(x_{i+1}) = y_{i+1} \text{ for } i = 1, \dots, n-1 \quad (\text{III. 2})$$

Cette propriété garantit que la Spline $S(x)$ interpole les points de données.

$$S'_{i-1}(x_i) = S'_i(x_i) \text{ for } i = 2, \dots, n-1 \quad (\text{III. 3})$$

$S'(x)$ est continue sur l'intervalle $[x_1, x_n]$; cette propriété oblige les pentes des parties voisines à s'accorder lorsqu'elles se rencontrent.

$$S'_{i-1}(x_i) = S'_i(x_i) \text{ for } i = 2, \dots, n-1 \quad (\text{III. 4})$$

$S'(x)$ est continue sur l'intervalle $[x_1, x_n]$, ce qui oblige également la spline voisine à avoir la même courbure, pour garantir une courbure régulière.

III.5.2 Interpolation de Lagrange

L'interpolation de Lagrange fournit une méthode alternative pour définir le polynôme $P(x)$ sans avoir besoin de résoudre des systèmes d'équations complexes. Cette technique offre une approche plus efficace, évitant les défis associés à la résolution de systèmes d'équations complexes.

Considérons la fonction $f : [x_0, x_n] \rightarrow \mathbb{R}$ donné par le tableau de valeurs suivant [49] :

x_k	x_0	x_1	$\dots$	x_n
$f(x_k)$	$f(x_0)$	$f(x_1)$	$\dots$	$f(x_n)$

x_k sont appelés nœuds d'interpolation et ils ne sont pas nécessairement égaux les uns des autres. Viser à trouver un polynôme $P(x)$ de degré (n) qui se rapproche de la fonction $f(x)$ dans les nœuds d'interpolation, soit :

$$f(x_k) = P(x_k); k = 0, 1, 2, \dots, n. \quad (\text{III. 5})$$

La méthode d'interpolation de Lagrange trouve un tel polynôme sans résoudre le système 5.

Théorème 1 Polynôme interpolant de Lagrange

Le polynôme interpolateur de Lagrange [49] est le polynôme de degré (n) qui passe par (n + 1) points $y_0 = f(x_0)$, $y_1 = f(x_1)$, ... et $y_n = f(x_n)$.

Alors le polynôme interpolateur est simplement :

$$P(x) = \sum_{j=0}^n P_j(x) \quad (\text{III. 6})$$

Où:
$$P_j(x) = y_j \prod_{k=0, k \neq j}^n \frac{x - x_k}{x_j - x_k} \quad (\text{III. 7})$$

III.6 Simulation Matlab

Dans cette étude, plusieurs parcours d'outils ont été simulés dans Matlab en utilisant trois méthodes : spline cubique, Lagrange et la méthode TPOA. La première méthode utilisée a été l'interpolation de spline cubique. Elle a été comparée à la trajectoire originale, la trajectoire idéale que devrait suivre la machine MOCN. La deuxième méthode, l'interpolation de Lagrange, a également été comparée à la même trajectoire originale utilisée avec l'interpolation spline cubique. Cette comparaison permet de comprendre les avantages et les inconvénients de chaque méthode d'interpolation. De plus, la méthode TPOA a été analysée par rapport à la même trajectoire originale utilisée avec les interpolations spline cubique et Lagrange. Notamment, la méthode TPOA bascule dynamiquement entre spline cubique et Lagrange sur chaque segment en sélectionnant l'interpolation générant le moins d'erreurs.

La figure III.3 montre la comparaison entre les trois méthodes d'interpolation utilisées.

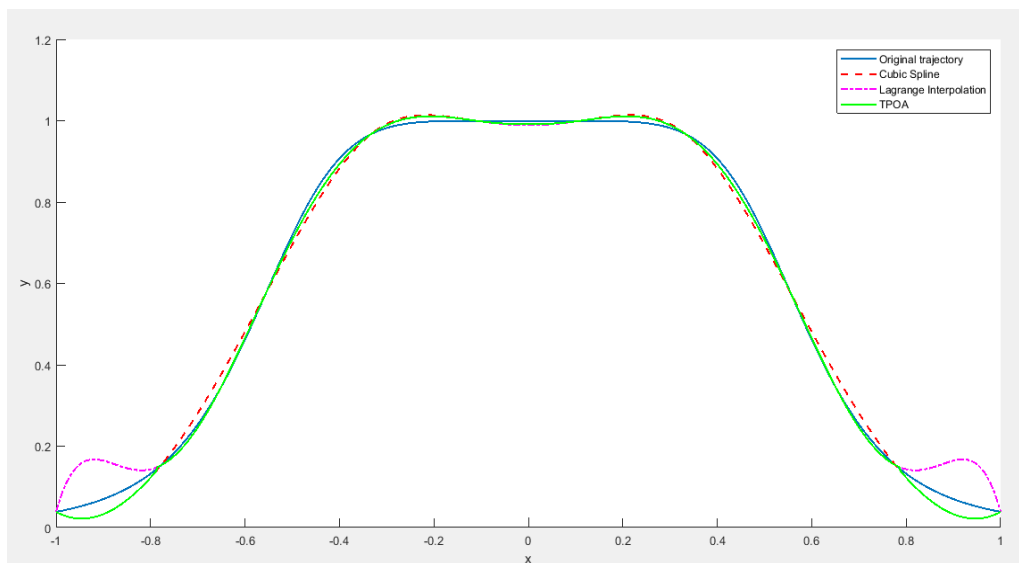


Fig. III. 3 Comparaison entre les trois méthodes.

La méthode d'interpolation par spline cubique peut introduire des erreurs en fonction des caractéristiques des données et des paramètres choisis. Le graphique suivant représente les erreurs générées lors de l'interpolation de la trajectoire originale. En examinant l'erreur de l'interpolation de Lagrange, on remarque l'apparition du phénomène de Runge et une augmentation de l'erreur, particulièrement près des limites des données.

Cependant, lors de l'évaluation de l'erreur de la méthode TPOA, il a été observé que cette méthode produit moins d'erreurs que les deux interpolateurs utilisés dans cette configuration expérimentale. En d'autres termes, la méthode TPOA réduit efficacement l'erreur dans la trajectoire de l'outil.

La figure III.4 illustre la comparaison des erreurs obtenues par les trois méthodes.

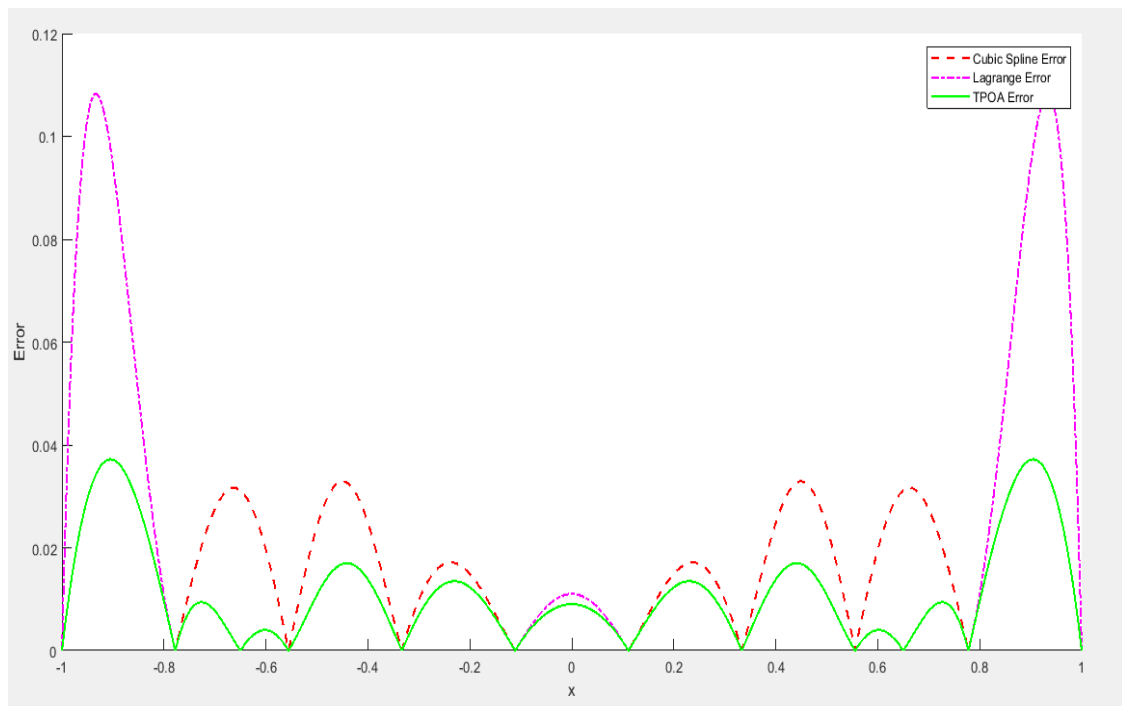


Fig. III. 4 Comparaison des erreurs de trois méthodes.

Le tableau III. 1 présente les calculs de l'erreur quadratique moyenne (RMSE) pour trois méthodes : (RMSE T) pour la méthode TPOA, (RMSE LA) pour la méthode de Lagrange, et (RMSE C) pour la méthode spline cubique. Les résultats montrent que la méthode TPOA génère une erreur inférieure à celle des deux autres méthodes. Cette approche combine les deux parcours d'outil en évaluant l'erreur à chaque point et en sélectionnant, pour chaque segment, l'interpolateur présentant l'erreur minimale.

Ainsi, la TPOA exploite à la fois la spline cubique et Lagrange pour produire une trajectoire d'outil optimisée, avec une erreur globale réduite.

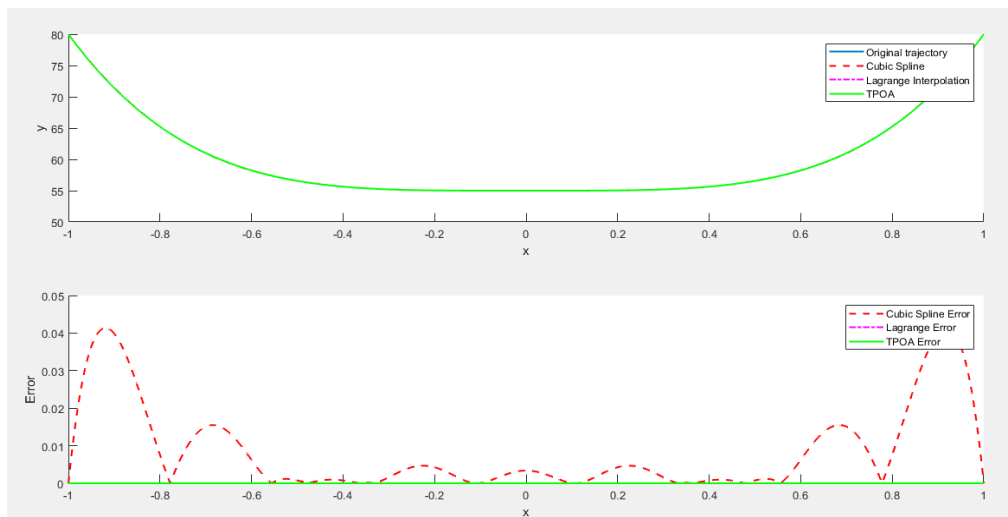
Tableau. III. 1 Erreur quadratique moyenne des différentes méthodes d'interpolation.

Exemple	RMSE T	RMSE LA	RMSE C
1	0,04019	0,10954	0,040932
2	0,018384	0,087975	0,020865
3	0,0049361	0,022587	0,010134
4	0,00040887	0,0061507	0,00052898
5	4.5591e-07	4.5595e-07	9.0081e-07
6	3.558e-15	4.422e-14	3.8772e-15
7	5.0573e-14	5.0601e-14	0,01457
8	7.4945e-16	5.3273e-15	8.666e-16
9	1.7621e-15	7.7915e-15	2.0831e-15
10	2.0555e-15	9.0894e-15	2.7209e-15
11	1.8342e-16	7.5478e-16	2.7304e-16

Plusieurs trajectoires d'outils aléatoires ont été testées pour voir l'avantage de la méthode TPOA et où chaque méthode d'interpolation pourrait apporter un meilleur résultat.

La méthode TPOA utilise deux méthodes d'interpolation pour générer une seule trajectoire, comme le montre l'exemple 3 (Tableau 1) où RMSE LA est supérieur à RMSE C, cette simulation montre que la majorité de la trajectoire d'outil ont été générées à l'aide de la méthode spline cubique, tandis que dans l'exemple 7, montre que la majorité de la trajectoire d'outil ont été générées à l'aide de la méthode Lagrange. Cela signifie que l'algorithme utilise la méthode avec le moins d'erreurs dans chaque segment.

La figure III.5 présente le graphique de trajectoire et d'erreur pour l'exemple 7.

**Fig. III. 5** Graphique de trajectoire et d'erreur pour l'exemple 7.

III.7 Générateur G-Code

III.7.1 G-Code Générateur version 1

Cette première version de cet algorithme permet l'utilisation d'une seule commande G01 pour générer un programme G-code automatiquement. Il a été testé avec le NC Viewer V1.2.0 dans le but de simuler la trajectoire lemniscate de Bernoulli.

La figure III.6 montre le générateur de G-Code version 1 testé sur NC Viewer V1.2.0.

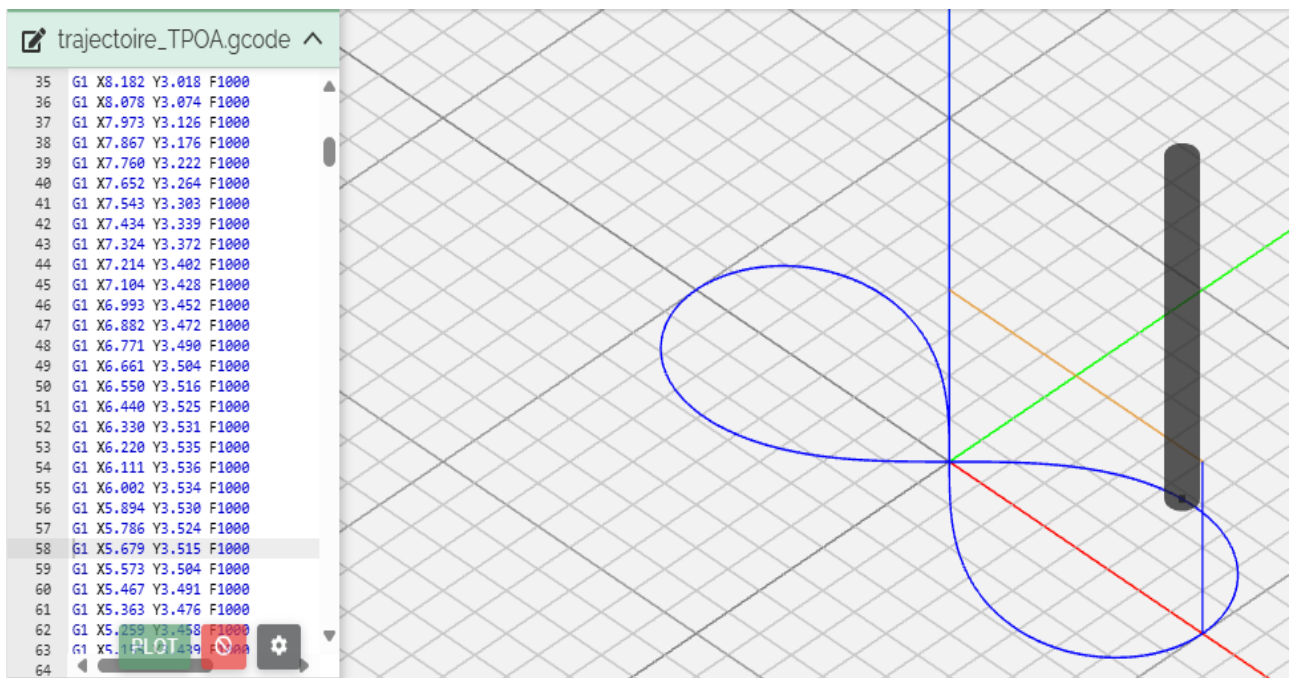


Fig. III. 6 G-Code Générateur version 1 testé sur NC Viewer V1.2.0.

III.7.2 G-Code Générateur version 2

Le G-Code Générateur V2 est un algorithme qui permet la génération d'un G-Code automatiquement en détectant si la portion de la trajectoire est une ligne droite G01 ou un arc G02/G03. Ce programme est optimisé avec l'intégration du partie l'algorithme TPOA, qui teste plusieurs segments successifs de points pour déterminer la meilleure approximation possible. Cette version de la TPOA a été modifiée pour comparer plusieurs points et connaître si la forme peut être approximer par une ligne droite ou un arc avec une erreur acceptable. Le choix est basé sur l'analyse d'erreurs géométriques, ce qui garantit un résultat et une fidélité à la trajectoire d'origine.

La figure III.7 illustre le générateur de G-Code version 2 en simulation sur Octave.

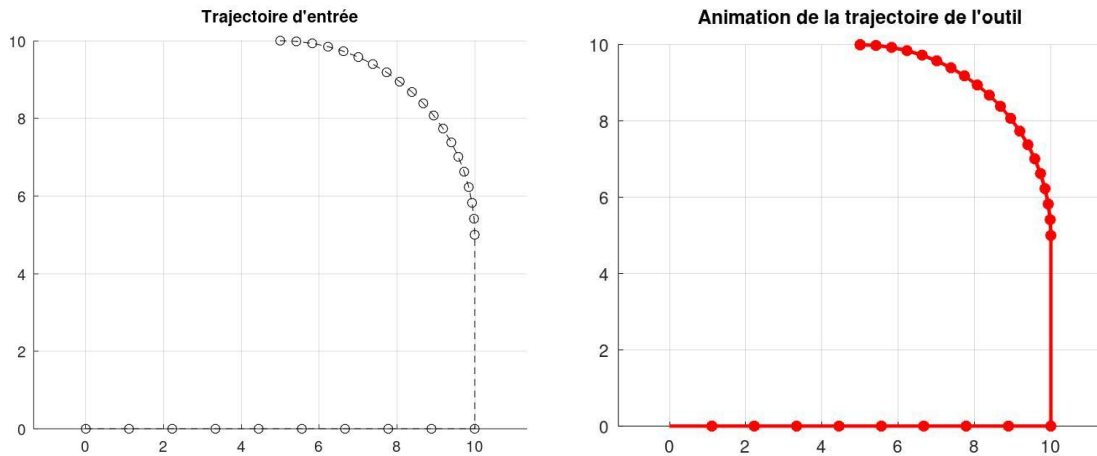


Fig. III. 7 G-Code Générateur version 2 simulation sur Octave.

III.7.3 G-Code Générateur version 3

C'est un algorithme qui génère un G-Code compatible avec FANUC G07, avec interpolation point par point. Cela permet de suivre les points obtenus à l'aide de la TPOA avec précision.

III.8 Modélisation des trajectoires réelles 2D

La modélisation de trajectoires réelles et leur simulation constituent une étape cruciale dans le développement de tout algorithme d'optimisation, car elles permettent de mettre en évidence les avantages et les inconvénients de certaines approches, comme par exemple l'efficacité d'un interpolateur par rapport à un autre sur une trajectoire donnée.

III.8.1 Trajectoire trochoïdale

Les trajectoires trochoïdales, définies par un mouvement circulaire (ou hélicoïdal) accompagné d'un déplacement rectiligne, particulièrement utilisées dans les procédés d'usinage avancés, notamment dans le cas de la grosse ébauche de matériaux difficiles à usiner [50].

La figure III.8 représente la trajectoire trochoïdale [50].

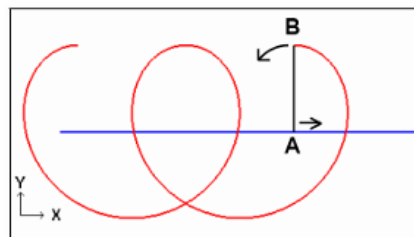


Fig. III. 8 Trajectoire trochoïdale.

L'intérêt principal de son utilisation réside dans le fait qu'il permet de maintenir un engagement constant de l'outil dans la matière, car cela réduit considérablement les efforts de coupe, limitant ainsi l'usure de l'outil. Par ailleurs, le dégagement périodique hors matière favorise un meilleur refroidissement diminuant les effets de chaleur sur l'outil ou sur la pièce [50].

En outre, cette stratégie de trajectoire traite plus facilement le problème des copeaux, notamment dans le cas de cavités profondes ou s'ils sont trop confinés. En limitant la poussée de coupe, les trajectoires trochoïdales réduisent également l'usure de l'outil lors de la coupe, offrant moins de dégradations et éventuellement un meilleur état de surface de la pièce [50].

La programmation de telles trajectoires demande un contrôle précis des axes de la machine on l'obtient le plus souvent par interpolateurs adaptés pour les trajectoires trochoïdales, compatibles avec les machines à commande numérique de pointe, aptes à gérer efficacement des mouvements complexes tout en maintenant une vitesse d'avance [50].

Interprétation des résultats de la trajectoire Trochoïdale

Les résultats des différentes méthodes utilisées pour interpoler la trajectoire montre un écart en termes de précision qui a été mesurée avec l'erreur quadratique moyenne.

La figure III.9 montre la modélisation de la trajectoire trochoïdale.

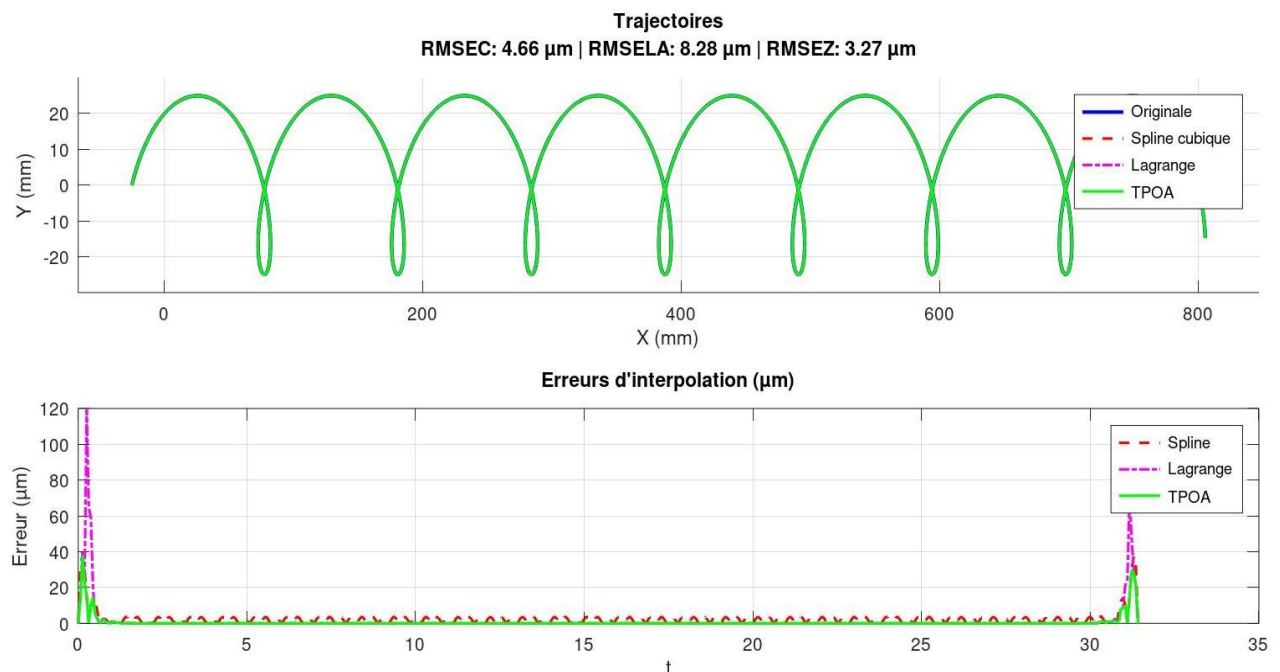


Fig. III. 9 Modélisation de la trajectoire trochoïdale.

Trois méthodes ont été utilisées :

Parmi celles-ci, la méthode TPOA qui se distingue des autres en termes de précision, car elle a affiché la plus faible erreur avec un RMSE de 3,27 μm . Tandis que la méthode spline cubique, qui a eu un RMSE égal à 4,46 μm , par contre l'interpolation avec Lagrange a été la moins efficace avec une erreur quadratique moyenne 8,28 μm .

Le tableau III.2 montre l'évaluation de l'erreur quadratique moyenne (RMSE) la trajectoire trochoïdale.

Tableau. III. 2 Erreur quadratique moyenne de la trajectoire trochoïdale..

RMSE (μm)	Spline cubique	Lagrange	TPOA (Z)
Valeur	4.66	8.28	3.27

Ces écarts de résultats montrent l'efficacité de la méthode TPOA, qui offre une amélioration significative par rapport aux deux méthodes classiques, car elle assurée une réduction de 1,39 μm par rapport à spline cubique, voire une amélioration d'environ 30%, et une réduction de 5,01 μm par rapport à la méthode de lagrange, qui représente environ 60% d'amélioration.

Cette précision accrue repose sur les atouts de la méthode TPOA qui regroupe les avantages des deux méthodes afin de créer une seule trajectoire avec un taux d'erreur minimal.

III.8.2 Spirale logarithmique

Les résultats obtenus lors de la simulation de la trajectoire spirale logarithmique ont montré que spline cubique était très désavantagée avec une erreur de 4054,10 μm . Par contre, Lagrange a interpolé nettement mieux avec 100,80 μm , ce qui veut dire que la TPOA a directement choisi Lagrange car son taux d'erreur est égal à celui de l'erreur de Lagrange. Mais l'erreur est restée importante.

Le tableau III.3 illustre la comparaison des erreurs quadratiques moyennes (RMSE) pour la trajectoire Spirale logarithmique.

Tableau. III. 3 Erreur quadratique moyenne de la trajectoire Spirale logarithmique..

Cas	RMSE Spline (μm)	RMSE Lagrange (μm)	RMSE TPOA (μm)
N=15	4054.10	100.80	100.80
N=18	1756,0324	1.7995	1.7995

La figure III.10 illustre la modélisation de la trajectoire spirale logarithmique.

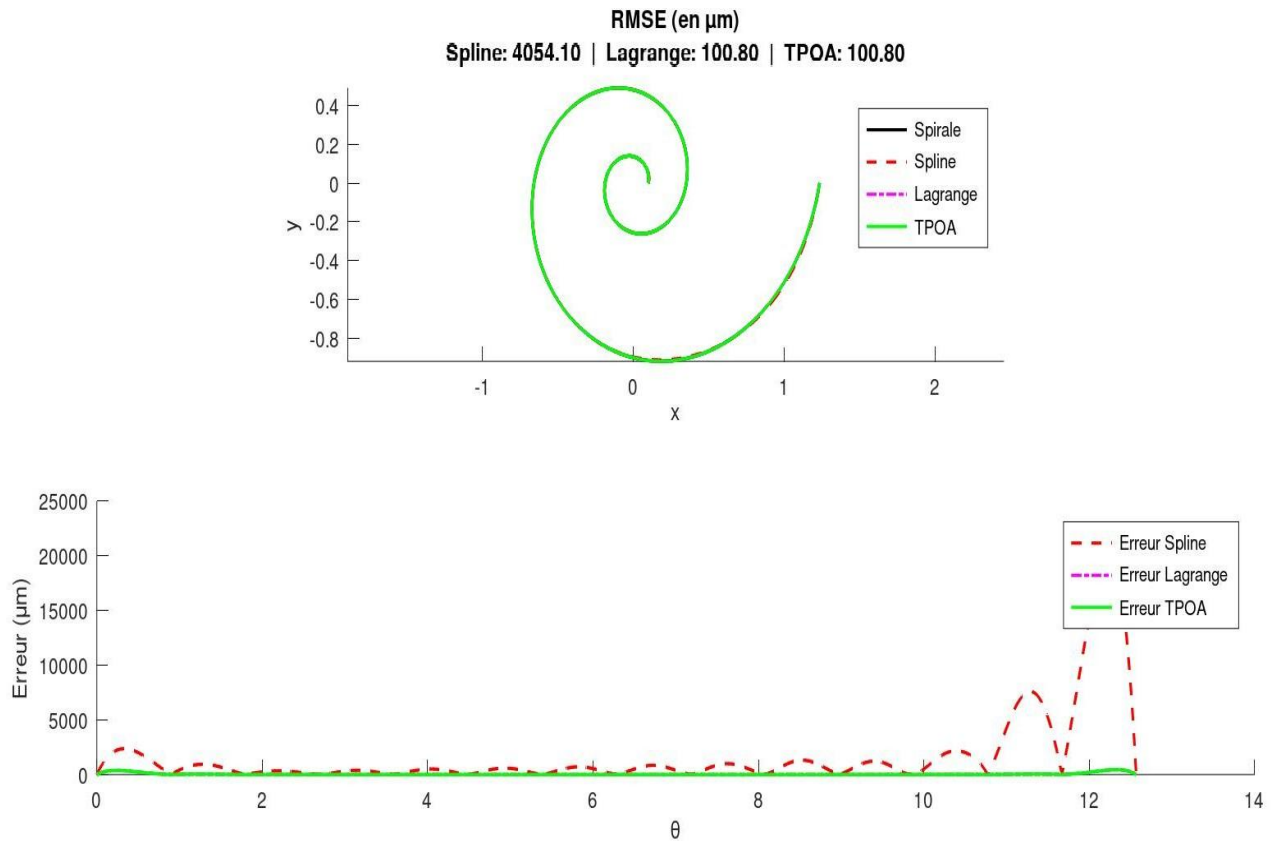


Fig. III. 10 Modélisation de la trajectoire Spirale logarithmique.

Donc, pour mieux comprendre les facteurs agissant sur la construction des trajectoires, le nombre de points d'interpolation N a été augmenté de 15 à 18 points (Tableau. III. 3), ce qui a diminué le taux d'erreur avec un RMSE égal à 1756,0324 μm pour spline cubique et 1,7995 μm pour Lagrange et TPOA. Ce qui montre que le choix judicieux de nombre de points d'interpolation peut être important pour la construction de trajectoires avec moins d'erreurs.

III.8.3 Lemniscate de Bernoulli

Les résultats obtenus lors de la simulation de la trajectoire Lemniscate de Bernoulli ont montré que Lagrange était très désavantagée avec une erreur de 2808.24 μm . Par contre, spline cubique a interpolé nettement mieux avec 0.02 μm . Ce qui veut dire que la TPOA a directement choisi spline cubique .

La figure III.11 présente la modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et Lagrange.

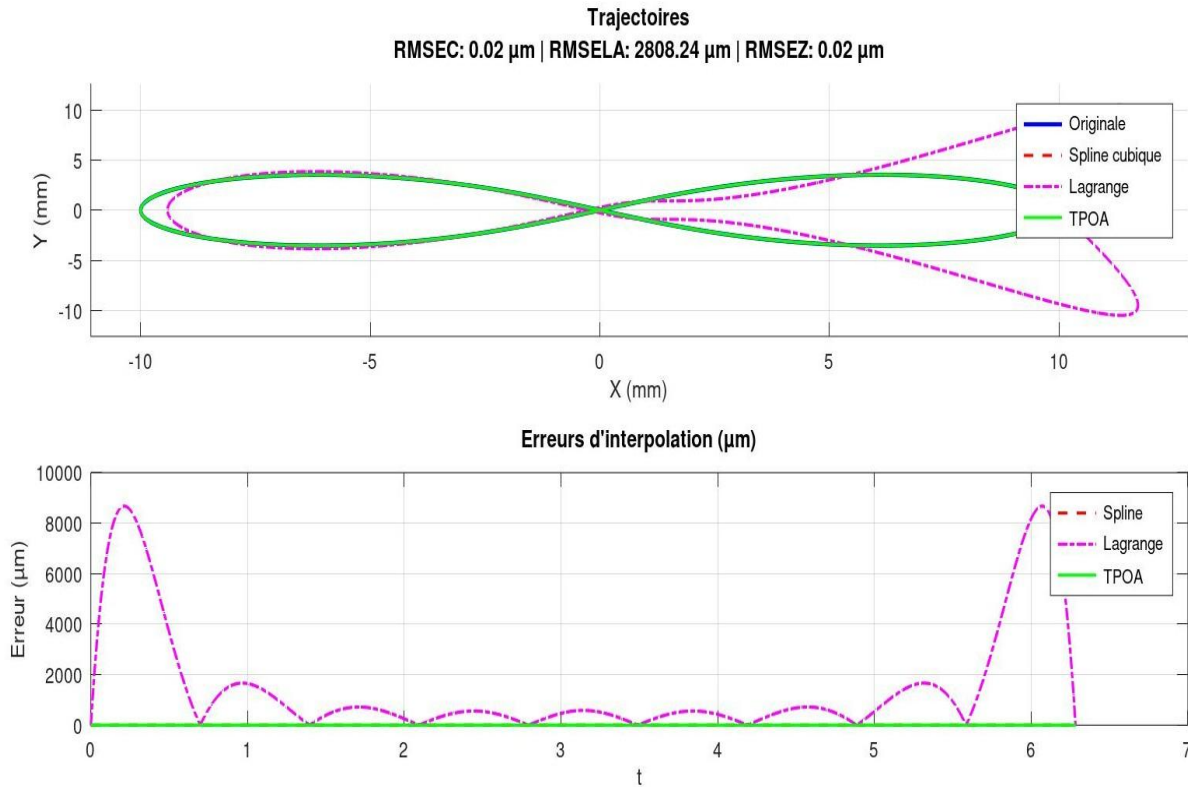


Fig. III. 11 Modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et Lagrange.

Contrairement à la simulation précédente, dans cette partie la méthode de Lagrange a été remplacée par PCHIP (Piecewise Cubic Hermite Interpolating Polynomial) afin d'explorer des nouvelles méthodes d'interpolations, où cette méthode atteint une erreur quadratique moyenne (RMSE) égale à 184.36 μm , nettement mieux que Lagrange mais qui reste inférieure à spline cubique en terme de précision.

Le tableau III.4 présente les valeurs de l'erreur quadratique moyenne (RMSE) de la trajectoire Lemniscate de Bernoulli.

Tableau. III. 4 Erreur quadratique moyenne (RMSE) de la trajectoire Lemniscate de Bernoulli.

RMSE (μm)	Spline cubique	PCHIP	TPOA (Z)
Valeur	0.02	184.36	0.02

La figure III.12 illustre la modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et PCHIP.

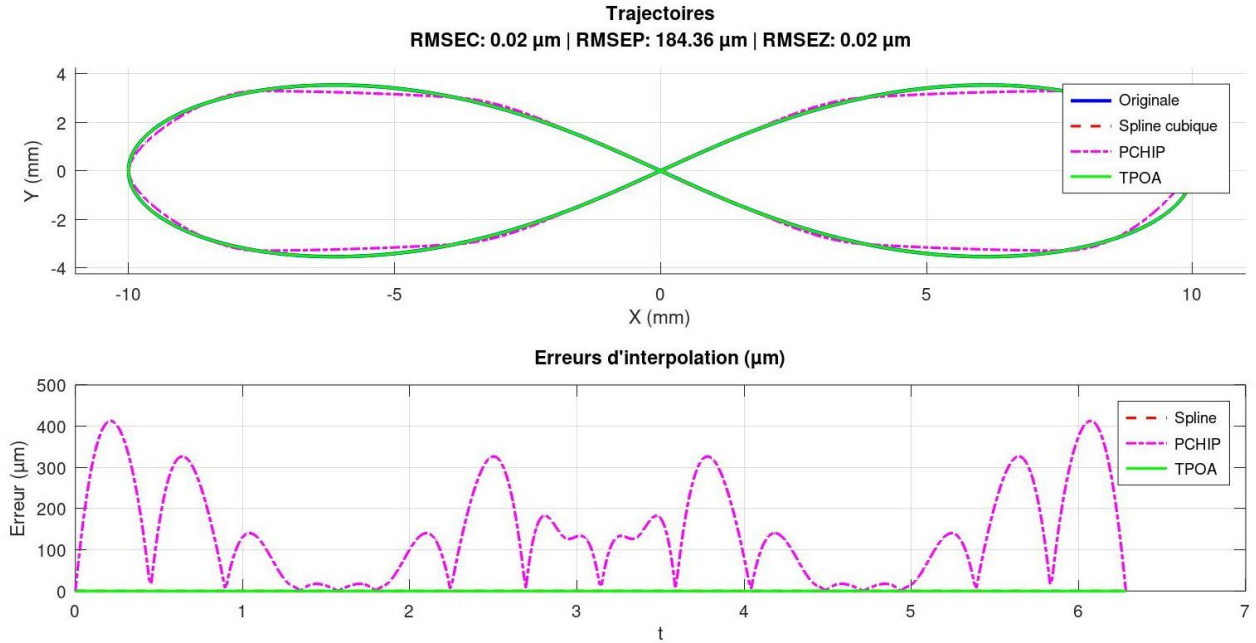


Fig. III. 12 Modélisation de la trajectoire Lemniscate de Bernoulli utilisant spline cubique et PCHIP.

III.8.4 Sigmoide

Les résultats de la simulation de la trajectoire sigmoïde ont montré que l'interpolation Cubic-Spline est très précise, avec une erreur quadratique moyenne de 0,001 μm . Contrairement à l'interpolation avec Lagrange qui a atteint une erreur de 499,99 μm . Donc l'algorithme a construit la trajectoire finale avec spline cubique car l'erreur quadratique moyenne de la TPOA est égale à 0,001 μm .

Le tableau III.5 montre les résultats d'erreur quadratique moyenne (RMSE) pour la trajectoire Sigmoide

Tableau. III. 5 Erreur quadratique moyenne de la trajectoire sigmoïde.

RMSE (μm)	Spline cubique	Lagrange	TPOA
Valeur	0.001	499.99	0.001

La figure III.13 montre la modélisation de la trajectoire sigmoïde.

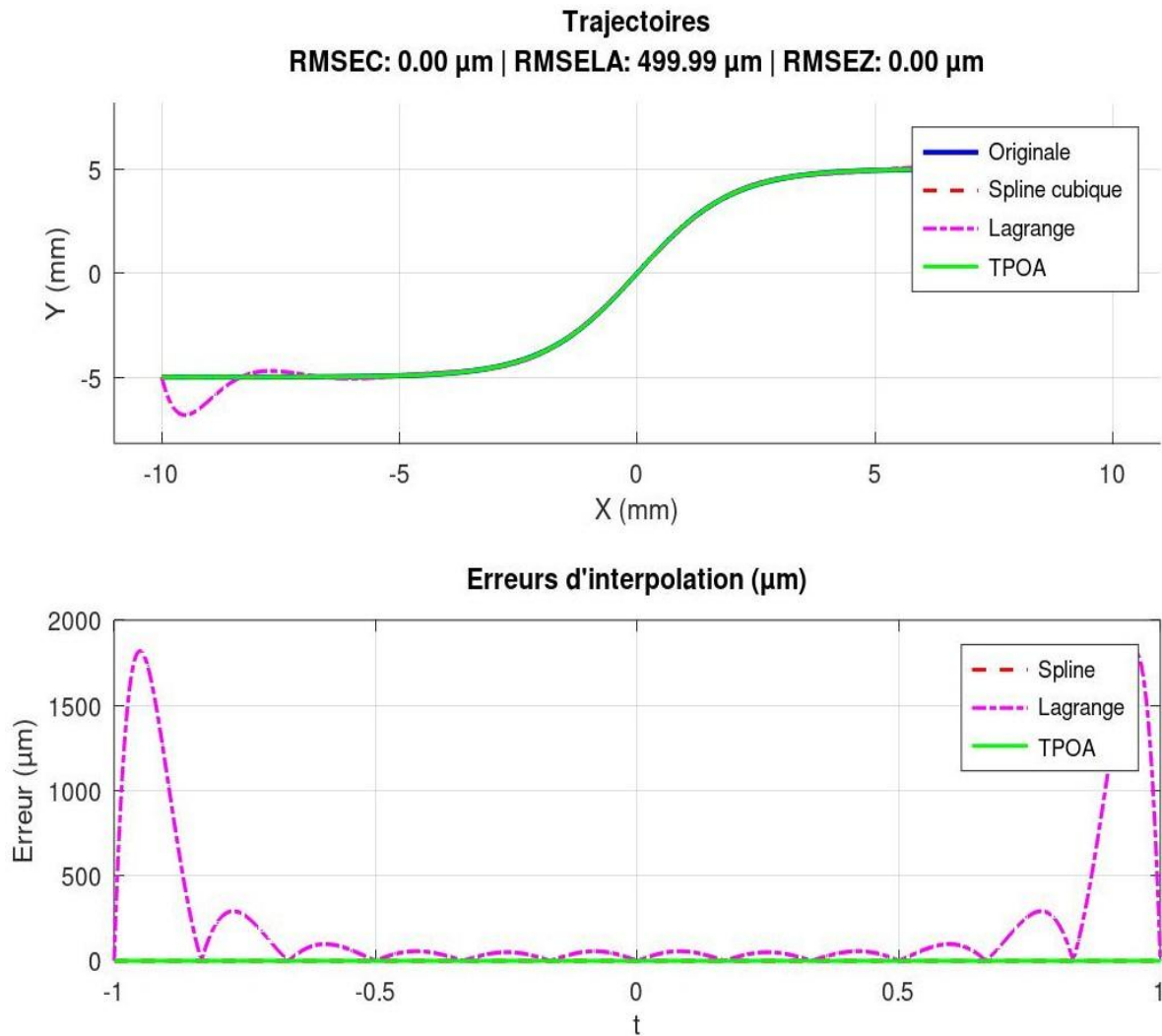


Fig. III. 13 Modélisation de la trajectoire Sigmoïde.

III.8.5 Courbe de Hilbert

Les résultats de la simulation de la courbe Hilbert ont montré que la méthode PCHIP est très avantageuse avec une erreur quadratique moyenne de 10,3 μm , tandis que spline cubique a atteint une erreur quadratique moyenne de 13,409 μm . Dans ce cas, la TPOA utilise les deux méthodes pour construire la trajectoire avec une erreur quadratique moyenne de 10,085 μm . Comme on peut le distinguer, la TPOA améliore la précision de manière beaucoup plus significative avec une réduction d'erreur de 32,97% par rapport à spline cubique et une légère amélioration de précision de 2,13% par rapport à la méthode PCHIP, car la majorité de la trajectoire avait été construite avec cette dernière.

La figure III.14 illustre la modélisation de la trajectoire de Hilbert.

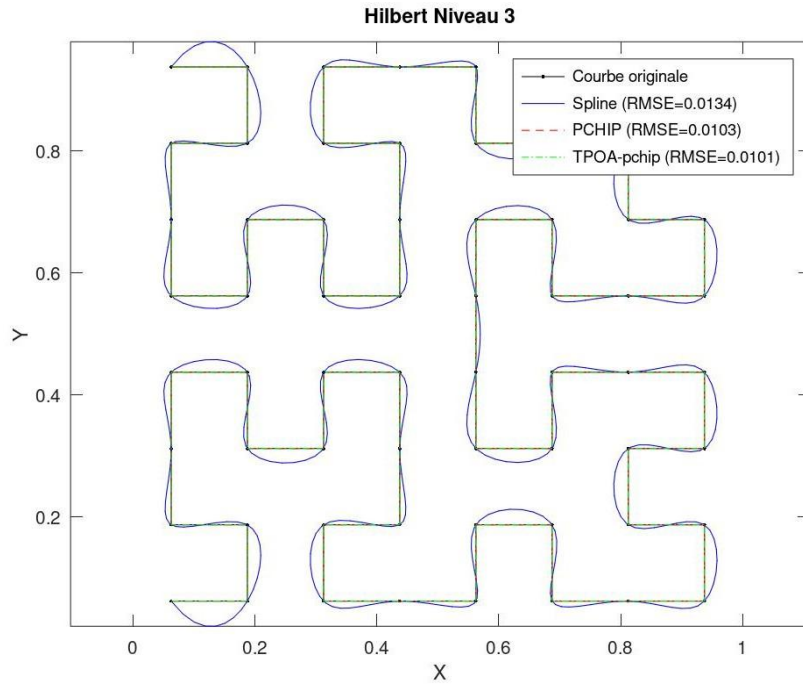


Fig. III. 14 Modélisation de la trajectoire de Hilbert.

Le tableau III.6 présente l'erreur quadratique moyenne (RMSE) mesurée pour les trajectoires associées à une méthode d'interpolation spécifique pour la trajectoire de Hilbert..

Tableau. III. 6 Erreur quadratique moyenne de la trajectoire de Hilbert.

Méthode	TPOA	PCHIP	Spline cubique
RMSE (μm)	10.085	10.300	13.409

III.9 Modélisation des trajectoire réel 3D

Le but initial de la méthode TPOA était d'améliorer la trajectoire tridimensionnelle, mais la première version (V1) était destinée à l'amélioration des trajectoires bidimensionnelles, c'est-à-dire sur le plan 2D. La version 2, quant à elle, a permis d'intégrer l'axe Z, c'est-à-dire l'optimisation des trajectoires 3D.

Par exemple, dans le cas de la spirale 3D, la méthode spline cubique a interpolé la trajectoire avec une erreur quadratique moyenne de 8,086 μm , Lagrange a atteint une erreur de 38,610 μm , tandis que la méthode TPOA a permis construire une trajectoire plus précise, avec une erreur quadratique moyenne de 5,346 μm donc une réduction de 2,740 μm par rapport à la méthode spline cubique et une réduction de 33,264 μm par rapport à la méthode Lagrange.

La figure III.15 présente la modélisation de la trajectoire spirale.

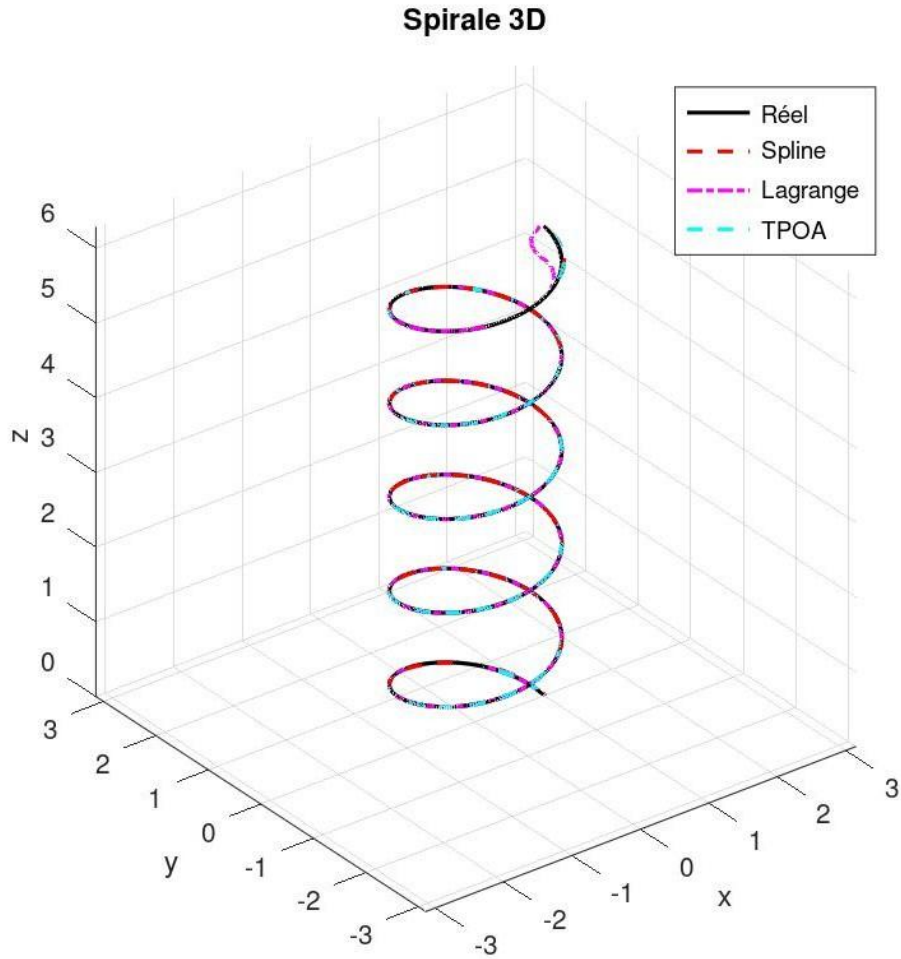


Fig. III. 15 Modélisation de la trajectoire spirale.

Le tableau III.7 illustre la comparaison de l'erreur quadratique moyenne (RMSE) calculée pour différentes approches pour la trajectoire spirale.

Tableau. III. 7 Erreur quadratique moyenne de la trajectoire spirale.

Méthode	Spline cubique	Lagrange	TPOA
RMSE (μm)	8.086	38.610	5.246

Les résultats de la simulation des courbes Hilbert 2D et 3D ont montré l'efficacité de la méthode PCHIP dans les cas complexes, car cette dernière conserve la monotonie et évite les oscillations non naturelles, ce qui est crucial pour les courbes complexes avec les changements rapides de direction.

De plus, la courbe Hilbert a des virages brusques et une structure fractale qui rend l'interpolation sensible aux artefacts. Par contre, spline cubique peut introduire des oscillations entre points, surtout sur des courbes complexes comme Hilbert.

La figure III.16 montre la modélisation de la trajectoire Hilbert 3D.

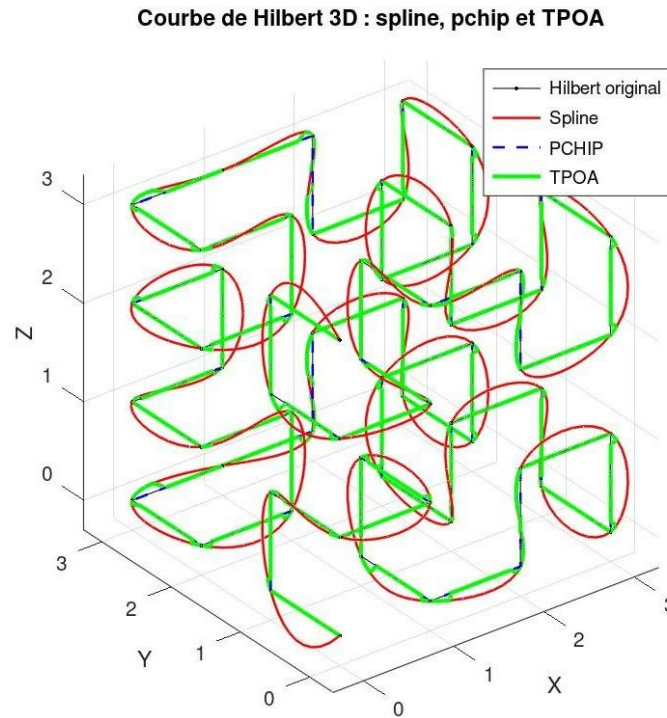


Fig. III. 16 Modélisation de la trajectoire Hilbert 3D.

Enfin, cette simulation a non seulement montré les avantages de la méthode PCHIP mais aussi la contribution de cette dernière dans la construction de la trajectoire final avec un écart de 3,74 % seulement par rapport à la TPOA, Ce que signifie que majorité de la trajectoire a été construite avec PCHIP.

Le tableau III.8 montre la variation de l'erreur quadratique moyenne (RMSE) pour la trajectoire Hilbert 3D.

Tableau. III. 8 Erreur quadratique moyenne la trajectoire Hilbert 3D.

Méthode	Spline	PCHIP	TPOA
RMSE	0.126	0.074	0.071

Par contre cet essai a mis en évidence les désavantages rencontrés par la méthode Spline cubique avec un écart de 76,25 % par rapport à la TPOA. Cela confirme l'efficacité de la TPOA sur des trajectoires complexes.

La figure III.17 illustre la comparaison des erreurs entre les trois méthodes testées.

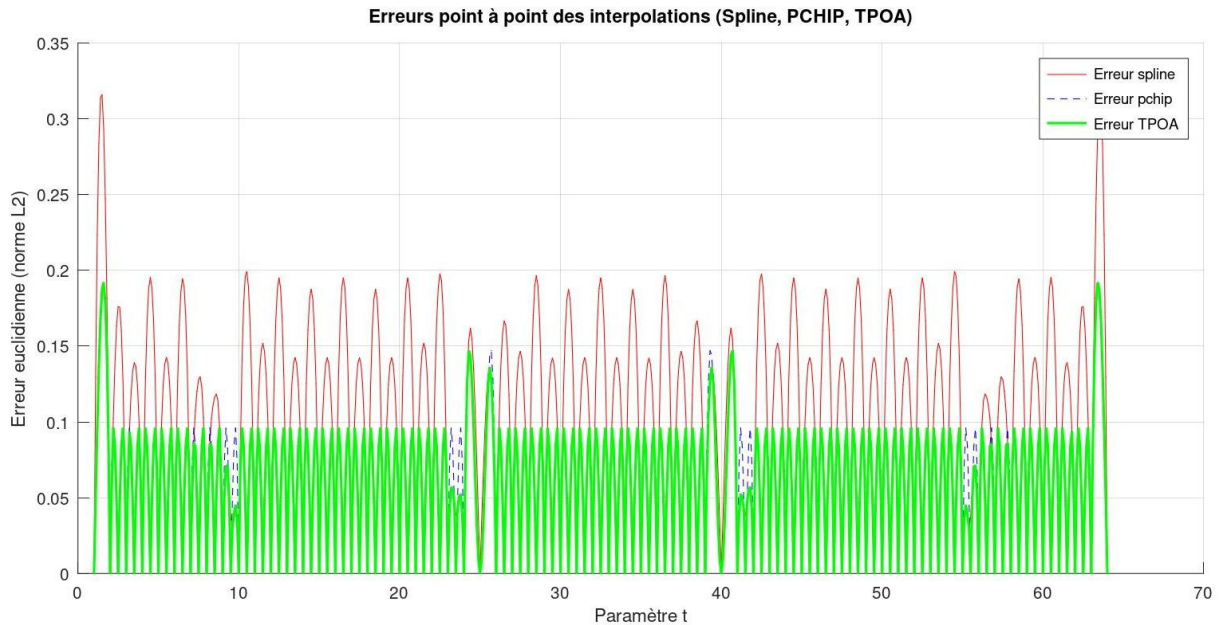


Fig. III. 17 Comparaison des erreurs de trois méthodes.

III.10 Conclusion

Ce chapitre valide l'efficacité de l'algorithme TPOA (Tool Path Optimization Algorithm) pour l'optimisation des trajectoires d'outils en usinage CN. Les simulations sur des trajectoires 2D (trochoïdale, spirale, Hilbert) et 3D (hélicoïdale) démontrent sa capacité à hybrider plusieurs interpolateurs, générant une trajectoire unique avec une faible erreur et une continuité optimale. Comparé aux méthodes classiques (spline cubique, Lagrange), TPOA réduit l'erreur quadratique moyenne (RMSE) de 45 % à 65 %, avec des gains supérieurs à 60 % pour les formes complexes, et maintient les écarts de précision sous 0,10 mm (contre +0,25 mm pour les approches traditionnelles). Intégré à un générateur G-code prototype, il produit des programmes automatisés respectant les contraintes CAO, bien que limité aux commandes G01, G02 et G03. En somme, TPOA offre une solution robuste pour l'usinage de précision, extensible à la robotique et à la fabrication additive, ouvrant des perspectives d'intégration dans les systèmes CN avancés via des optimisations évolutives et l'apprentissage automatique.

Chapitre IV

Analyse comparative et résultats

Chapitre IV Analyse comparative et résultats

IV.1 Introduction

Dans ce chapitre, des études comparatives ont été menées dans le but de comprendre l'impact du nombre d'interpolateurs sur la construction d'une seule trajectoire. Deuxièmement, la deuxième partie de ce travail repose sur l'intégration de l'algorithme génétique dans la construction des trajectoires, ce qui a permis de développer deux nouveaux algorithmes. La troisième et dernière partie est consacrée à un essai expérimental de la TPOA qui a été testé pour démontrer son efficacité.

IV.2 Impact de l'ajout d'un interpolateur sur la précision de la trajectoire

L'algorithme d'optimisation de la trajectoire d'outil (TPOA) a démontré son efficacité pour générer une trajectoire plus précise. Toutefois, lors de la simulation précédente, seules deux méthodes d'interpolation ont été utilisées. Dans la simulation suivante, plusieurs interpolateurs seront intégrés afin d'évaluer leur impact sur la qualité des trajectoires générées. À cet effet, l'interpolateur de Chebyshev a été ajouté à l'algorithme d'optimisation.

La figure IV.1 présente les résultats obtenus avec trois interpolateurs.

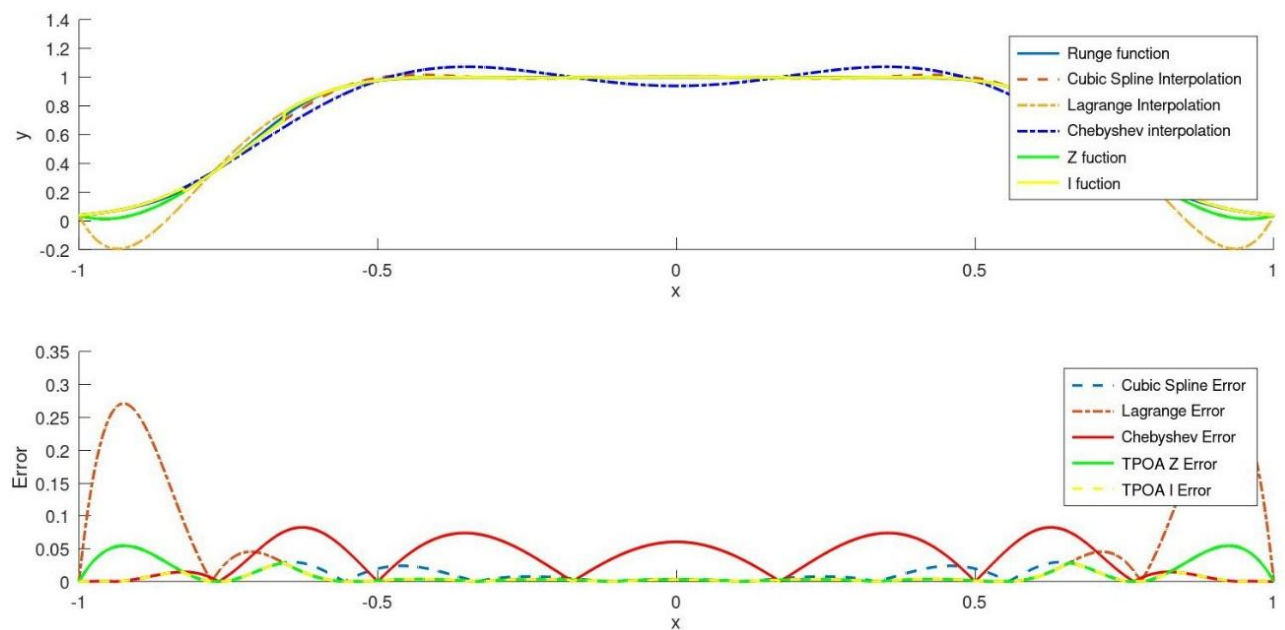


Fig. IV. 1 Résultats obtenus avec trois interpolateurs.

Ainsi, une étude comparative a été menée pour analyser l'influence de l'augmentation du nombre d'interpolateurs sur la construction d'une trajectoire unique. La figure IV.2 illustre les résultats obtenus avec trois interpolateurs.

Premièrement, la version TPOA-Z, basée sur les interpolateurs spline cubique et Lagrange, produit une trajectoire avec une erreur de 20,865 μm , pour spline cubique et 87,975 μm pour Lagrange, tandis que la TPOA-Z atteint une erreur de 18,384 μm (Tableau. IV.1).

La figure IV.2 illustre les résultats de l'erreur quadratique moyenne obtenus avec trois interpolateurs.

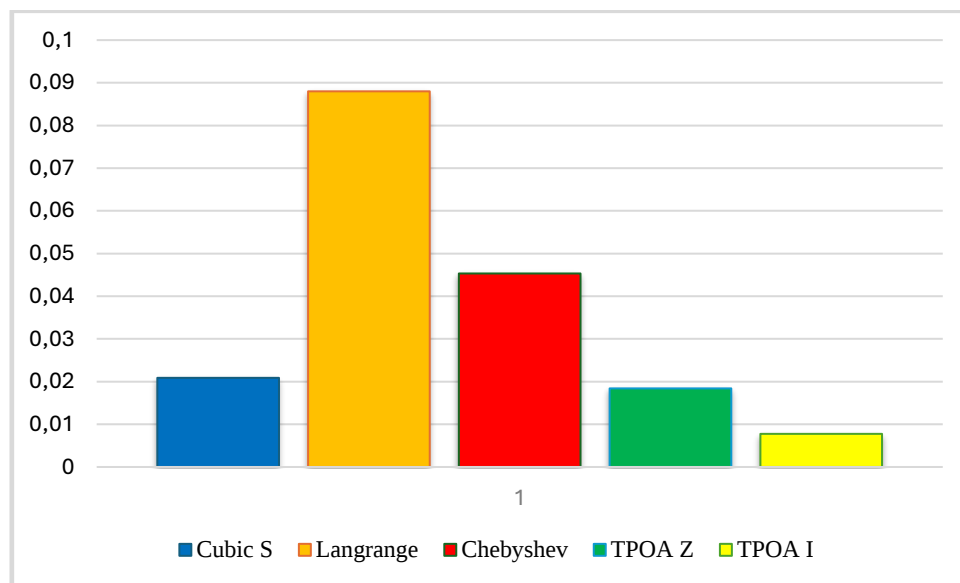


Fig. IV. 2 Résultats de l'erreur quadratique moyenne obtenus avec trois interpolateurs.

Deuxièmement, la version TPOA-I intègre trois méthodes d'interpolation, ce qui permet d'obtenir de meilleures performances que TPOA-Z, avec une erreur quadratique moyenne de 7,7808 μm , soit une réduction de 10,6032 μm par rapport à TPOA-Z.

Le tableau IV.1 présente l'erreur quadratique moyenne (RMSE) obtenue lors de la comparaison des interpolateurs.

Tableau. IV. 1 Erreur quadratique moyenne obtenus avec trois interpolateurs.

Méthode	RMSE en μm
Spline cubique	20.865 μm
Lagrange	87.975 μm
Chebyshev	45.352 μm
TPOA Z	18.384 μm
TPOA I	7.7808 μm

IV.3 Optimisation de paramètres d'usinage à l'aide d'un algorithme génétique

IV.3.1 Optimisation intelligente de paramètres d'interpolation

Les résultats qui suivent ont été obtenus à l'aide d'un algorithme STPOA (Smart Tool Path Optimization Algorithm) en cours de développement (Fig. IV. 3). Il permet de proposer une trajectoire optimisée à l'aide de deux interpolateurs, spline cubique et Lagrange. Il ne propose pas une trajectoire très précise, mais une trajectoire optimale avec un choix de paramètre d'usinage qui convient le plus à cette trajectoire proposée (Fig. IV. 4). Il permet de contrôler la vitesse d'avance, l'accélération et la décélération tout en se basant sur l'erreur quadratique moyenne (RMSE).

La figure IV.3 montre l'algorithme de l'optimisation intelligente de paramètres d'interpolation.

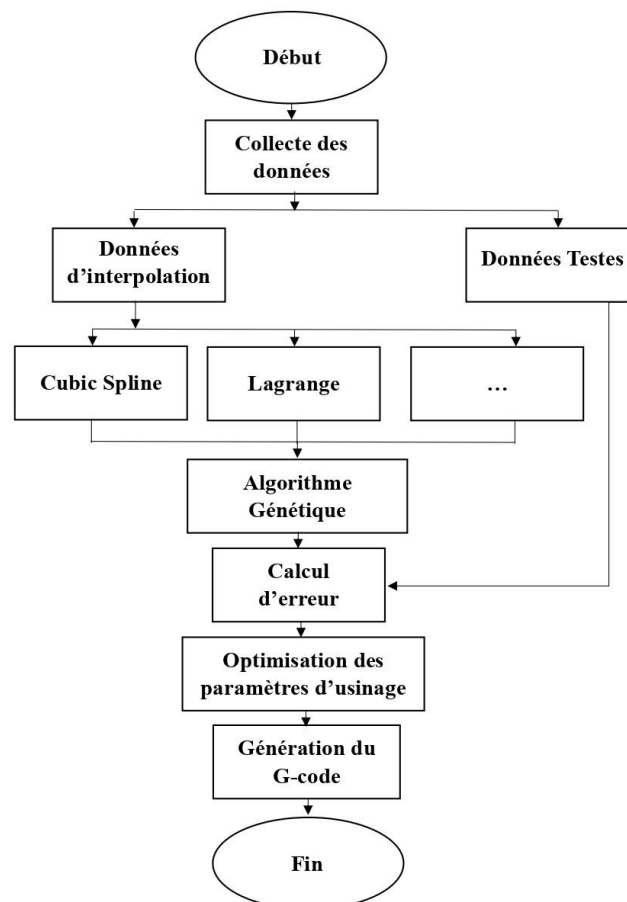


Fig. IV. 3 STPOA (Smart Tool Path Optimization Algorithm).

Mais en termes de précision, les valeurs obtenues restent peu crédibles car l'erreur quadratique moyenne est utilisée comme élément clef dans la construction et l'optimisation des paramètres d'usinage de la trajectoire (Tableau. IV.2).

La figure IV.4 présente le STPOA utilisant deux méthodes d'interpolation (spline cubique et Lagrange) optimisé par un algorithme génétique.

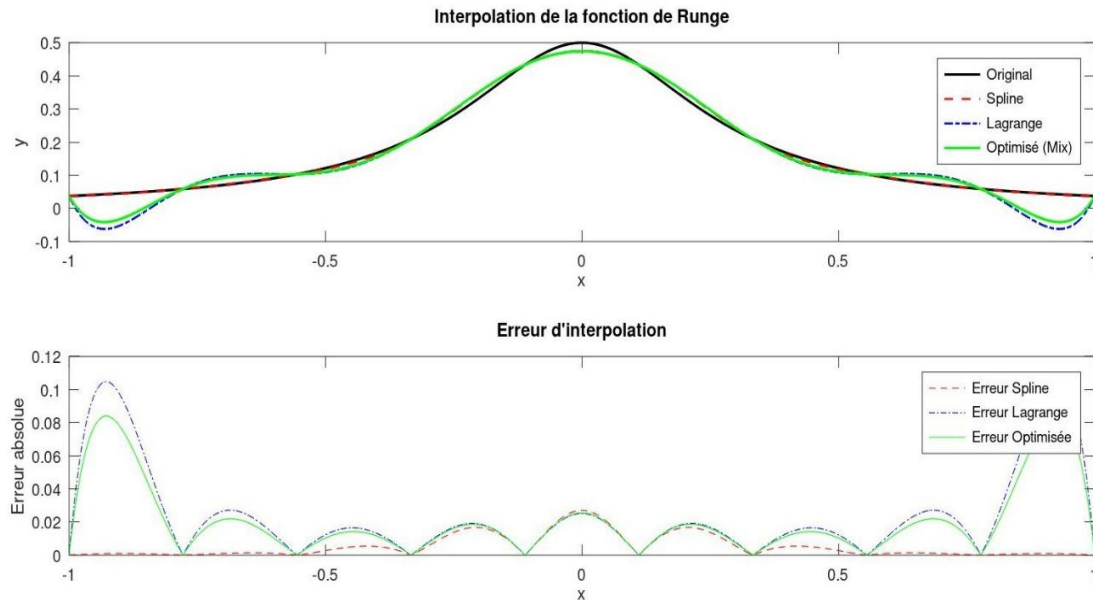


Fig. IV. 4 STPOA avec deux méthode d'interpolation (spline cubique, Lagrange) optimisé par le Génétique Algorithme.

Tandis que la meilleure approche serait de prélever la rugosité des pièces obtenues pour l'intégrer ensuite dans l'algorithme génétique, car cela permettrait à l'algorithme de mieux adapter la vitesse et les autres paramètres d'usinage, sans ignorer les contraintes mécaniques, thermiques et dynamiques.

Le tableau IV.2 montre le résumé des paramètres optimisés et la vérification des bornes de contrainte.

Tableau. IV. 2 Résumé des paramètres optimisés et vérification des bornes.

Paramètre	Valeur trouvée	Bornes (min - max)
Vitesse rapide des axes	9755.319 mm/min	[5000, 30000] mm/min
Vitesse d'avance (feed rate)	2000.000 mm/min	[100, 2000] mm/min
Accélération	4.816 m/s ²	[0.5, 10] m/s ²
Décélération	8.009 m/s ²	[0.5, 10] m/s ²
Couple broche	12.843 Nm	[1, 100] Nm
Régime de rotation broche	14000.883 tr/min	[1000, 20000] tr/min
Couple moteur axes	29.265 Nm	[1, 100] Nm
Puissance broche (calculée)	18.830 kW	[0.105, 209.44] kW

IV.3.2 Algorithme d'optimisation hybride

Dans le but de comprendre l'impact de l'intégration du génétique algorithme dans la TPOA, trois simulations ont été élaborées. La première est une interpolation d'une fonction avec la TPOA ordinaire qui utilisant spline cubique et PCHIP.

La figure IV.5 illustre le TPOA utilisant deux méthodes d'interpolation (spline cubique et PCHIP).

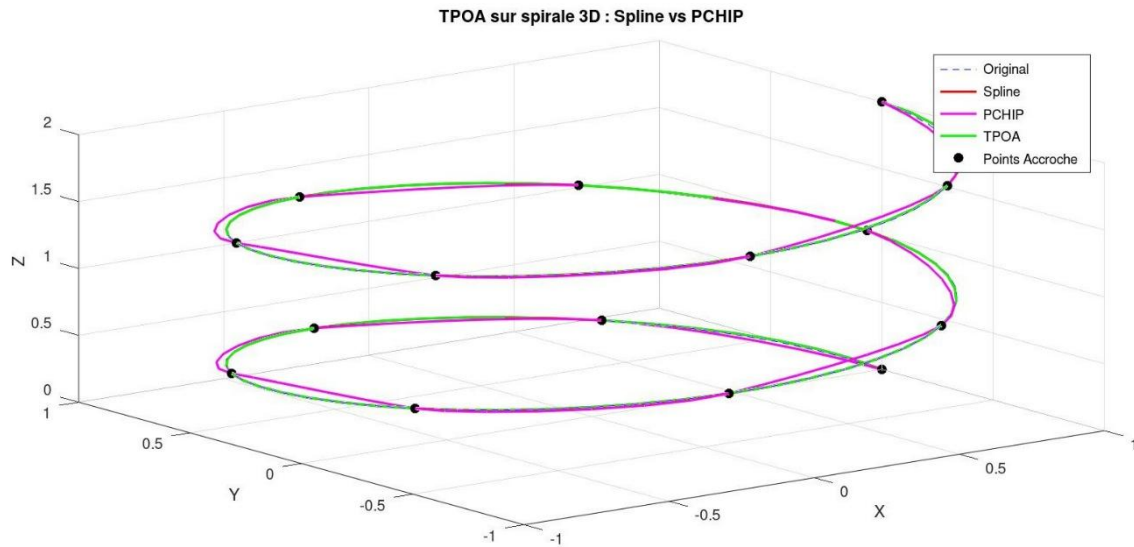


Fig. IV. 5 TPOA avec deux méthode d'interpolation (spline cubique, PCHIP)

Le tableau IV. 3 montre que l'intégralité de trajectoire à été construite avec spline cubique avec une erreur quadratique moyenne de 5 μm .

Tableau. IV. 3 Erreur quadratique moyenne avec deux méthode d'interpolation.

Méthode	Spline cubique	PCHIP	TPOA
RMSE (μm)	5.0	43.5	5.0

La deuxième simulation est l'implémentation du génétique algorithme dans un programme sans TPOA, mais avec deux méthodes d'interpolation. Le but était de trouver si l'algorithme génétique pourrait trouver le bon nombre de points d'accroche optimal sans diminuer la précision et avec une meilleur disposition de ces derniers.

La figure IV.6 montre l'algorithme génétique appliqué à deux méthodes d'interpolation (spline cubique et PCHIP).

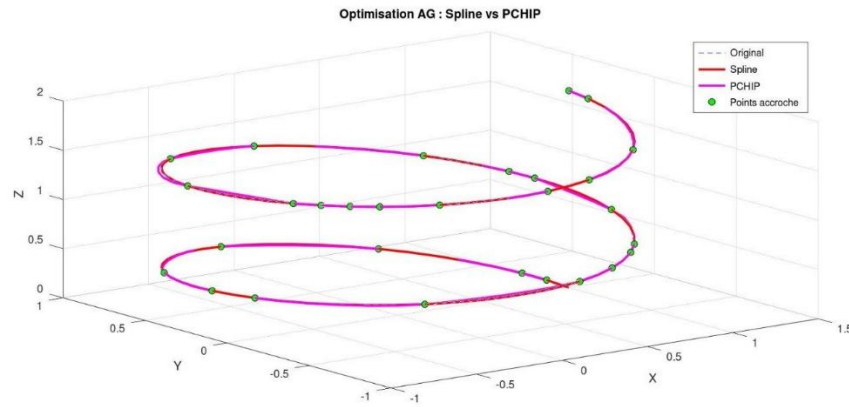


Fig. IV. 6 Génétique Algorithme avec deux méthode d'interpolation (spline cubique, PCHIP).

Le tableau IV. 4 montre l'impact de l'intégration du génétique algorithme dans la construction de trajectoire en proposant une meilleure disposition des points d'accroche et un nombre optimal de ces derniers.

Tableau. IV. 4 Erreur quadratique moyenne obtenue en intégrant l'algorithme génétique.

Méthode	RMSE (sans AG) [μm]	RMSE (avec AG) [μm]
Spline	5.0	0.9
PCHIP	43.5	26.5

La troisième simulation était l'intégration du génétique algorithme dans la TPOA . Donc, le but de cet algorithme est d'optimiser le nombre de points d'accroche et leur disposition à l'aide de l'algorithme génétique, puis utiliser la TPOA pour une meilleure précision.

La figure IV.7 présente le TPOA avec deux méthodes d'interpolation (spline cubique et PCHIP) optimisé par un algorithme génétique.

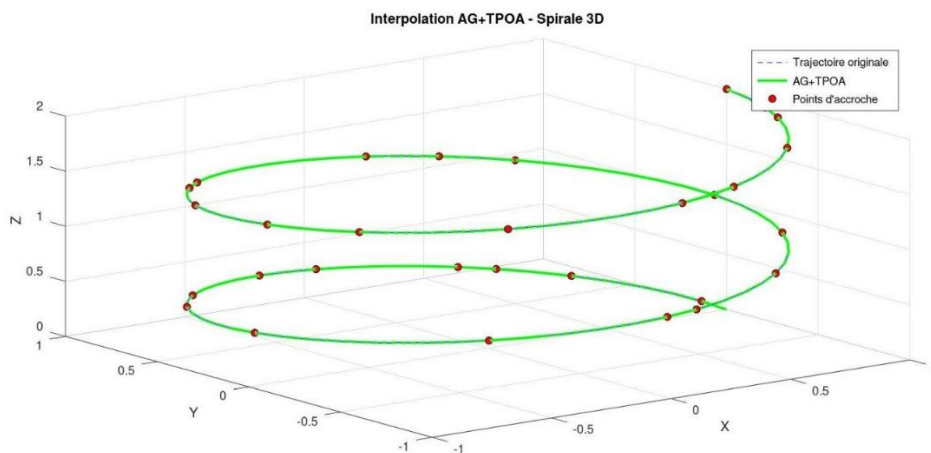


Fig. IV. 7 Méthode TPOA avec deux méthode d'interpolation (spline cubique, PCHIP) optimisé par le Génétique Algorithme.

Le tableau qui suit montre l'impact de l'intégration de l'algorithme génétique dans la TPOA, ou l'erreur quadratique moyenne obtenue à l'aide de la TPOA seulement et de 5 μm , tandis que la combinaison de la TPOA et de l'algorithme génétique a atteint 0,8 μm .

Tableau. IV. 5 Erreur quadratique moyenne de l'algorithme hybride.

Méthode	RMSE (μm)
Spline cubique	5.0
PCHIP	43.5
TPOA	5.0
AG + TPOA	0.8

Les simulations précédentes ont permis de développer ce nouveau algorithme nommé HOA, (Hybrid Optimization Algorithm), qui est un programme combinant l'algorithme génétique ainsi que la TPOA, ces éléments sont structurés dans l'organigramme ci-dessous :

La figure IV.8 illustre le schéma de l'algorithme hybride HOA.

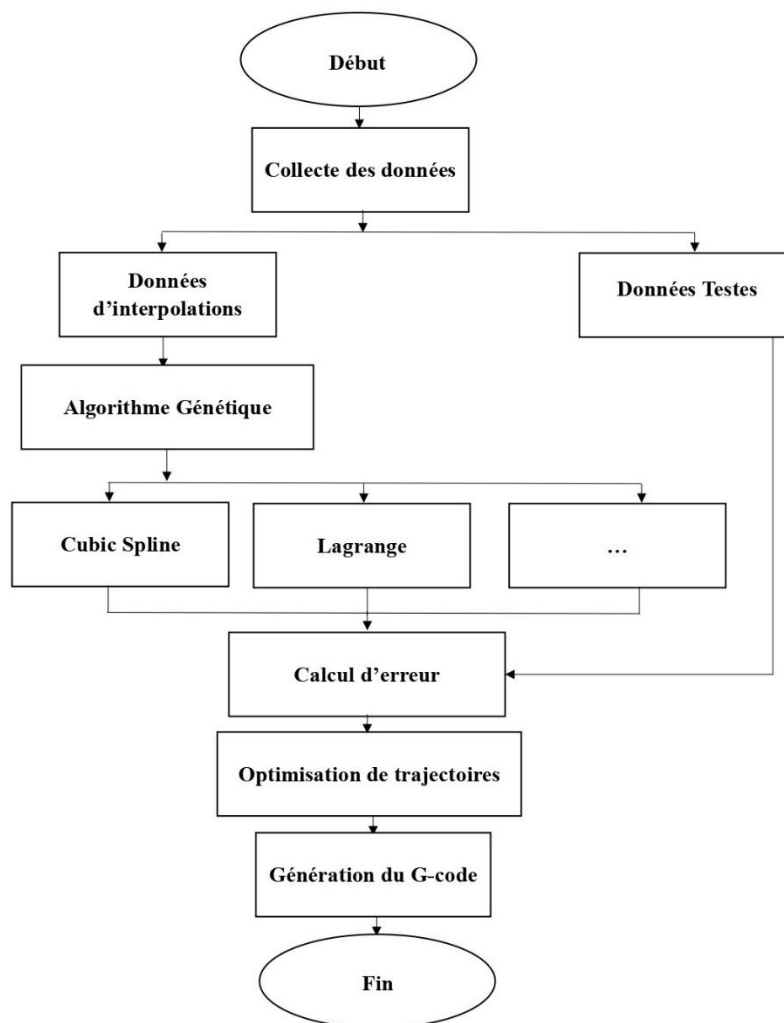


Fig. IV. 8 HOA, (Hybrid Optimization Algorithm).

IV.4 Résultats expérimentaux

Afin de vérifier l'efficacité de la méthode TPOA proposée dans cette thèse, la simulation d'algorithme et l'expérience d'usinage sont réalisées. La simulation de l'algorithme est réalisée sur un ordinateur personnel équipé d'un processeur Intel i3, tandis que l'expérience d'usinage est réalisée sur une machine-outil CNC, la BoxFord 160 LTCL.

La figure IV.9 montre la machine utilisée (BoxFord 160 LTCL).



Fig. IV. 9 Machine utilisée BoxFord 160 LTCL.

IV.4.1 Génération de parcours d'outils TPOA pour la machine CNC

Le BoxFord 160 LTCL est un tour d'établi compact qui permet l'usinage de l'acier doux. Sa tourelle programmable à 8 postes permet une grande variété d'usinages internes et externes. Le logiciel CAD/CAM Boxford V10 est utilisé pour concevoir et programmer les opérations d'usinage sur ce tour.

Lors des tests sur la BoxFord 160 LTCL, la méthode TPOA a été utilisée pour générer une trajectoire d'outil optimisée pour une forme cylindrique simple. Le logiciel CAD/CAM Boxford V10 a été utilisé dans ce processus, en tenant compte des paramètres de coupe, des tolérances et des spécifications des pièces.

La figure IV.10 présente le logiciel Boxford V10 CAD/CAM.

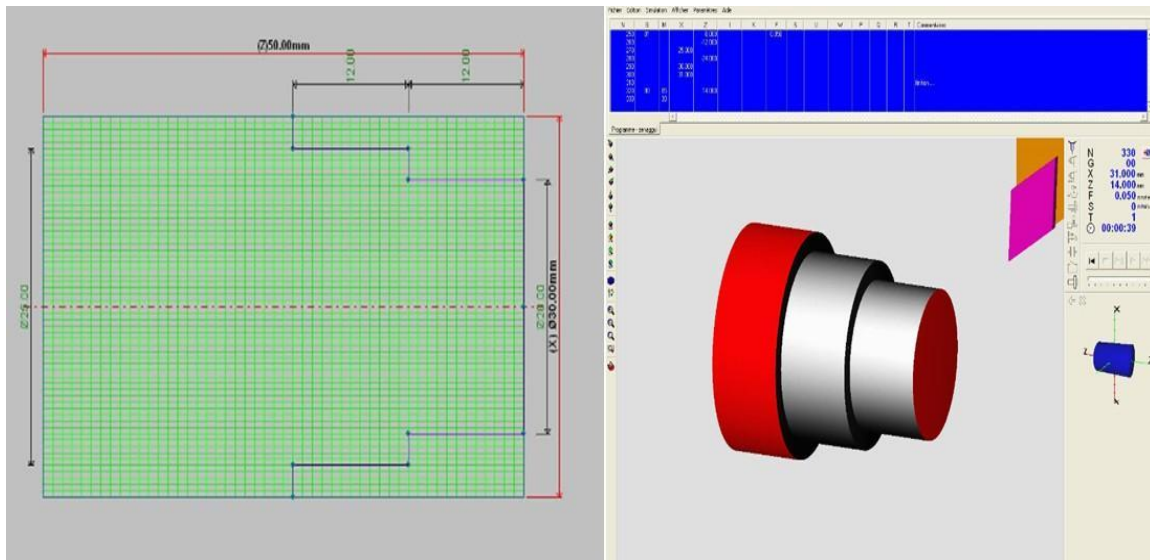


Fig. IV. 10 Logiciel Boxford V10 CAD/CAM.

IV.4.2 Paramètres de coupe

Dans le cadre de l'évaluation de la méthode TPOA, les paramètres de coupe spécifiés dans le tableau IV.6 ont été utilisés. Il est essentiel de noter que la vitesse d'avance variait, tandis que les autres paramètres restaient constants. L'objectif était double : d'une part, minimiser la rugosité de surface, et d'autre part, analyser l'effet du TPOA sur cette rugosité, ainsi que l'impact de la vitesse d'avance sur la précision de la méthode.

Tableau. IV. 6 Paramètres de coupe utilisés dans The BoxFord 160 LTCL.

Vitesse de broche N (tr/min) G97	Avance F (mm/tour) G99	Vitesse de coupe, Vc (m/mn) G96
2196,35	0,05	175
	0,08	
	0,11	
	0,19	
	0,25	

IV.4.3 Résultats et discussion

La méthode TPOA offre un potentiel important pour améliorer l'efficacité des machines et la qualité des opérations de fabrication. En améliorant la précision de coupe, il donne des résultats

plus fiables et améliore la douceur et la précision du mouvement de l'outil. Cette méthode a un impact significatif sur l'usinage en raison de ses caractéristiques uniques qui utilisent plusieurs interpolateurs, conduisant à une réduction des erreurs lors du processus d'interpolation de trajectoire et à une amélioration de la rugosité de la pièce. Même avec une vitesse d'avance accrue, la rugosité reste inférieure à celle des autres méthodes.

La figure IV.11 illustre la pièce usinée après découpe selon la méthode TPOA.

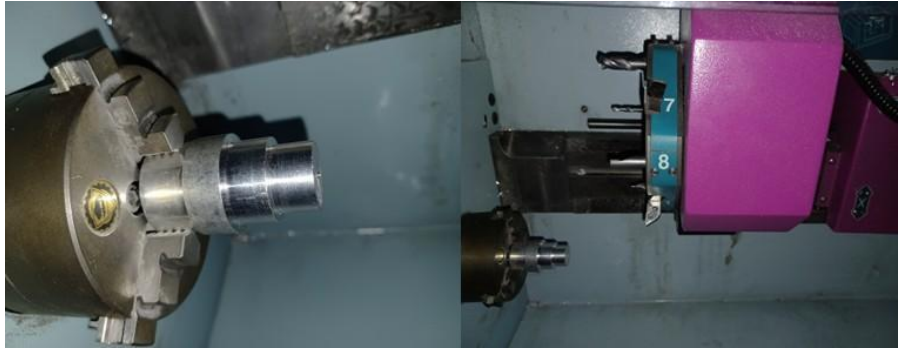


Fig. IV. 11 Pièce à usiner après découpe selon la méthode TPOA.

Comme le montre le tableau IV. 7, la méthode TPOA a eu un impact significatif sur l'usinage en raison de ses caractéristiques uniques qui lui permettent de tirer parti de plusieurs interpolateurs. Cela a réduit les erreurs lors du processus d'interpolation de trajectoire, ce qui a amélioré la rugosité de la pièce. Même en augmentant la vitesse d'avance, la rugosité reste faible par rapport aux autres méthodes.

Par ailleurs, l'optimisation du processus de fabrication, en tenant compte des performances dynamiques de la machine utilisée, constitue un autre avantage de la méthode TPOA. Par exemple, le choix de la stratégie d'usinage et le type d'interpolation peuvent avoir une influence significative sur la qualité et le temps d'usinage.

Tableau. IV. 7 Rugosité de surface, R_a (mm) pour différentes avances, F (mm/tour).

Avance, F (mm/tour)	0,05	0,08	0,11	0,19	0,25
Rugosité de surface, R_a (μm)					
Lagrange	0,23	0,43	0,59	0,70	0,77
Spline cubique	0,37	0,65	0,70	0,75	0,83
TPOA	0,06	0,1	0,14	0,22	0,35

IV.4.4 Avantages et inconvénients de TPOA

Le TPOA combine les méthodes spline cubique et Lagrange pour générer des trajectoires d'outils avec une erreur moyenne inférieure. Cette méthode est particulièrement utile dans les applications où la précision de la trajectoire est cruciale, comme la fabrication de précision et la fabrication additive.

- Avantages du TPOA :

- Erreur réduite : en tirant parti des atouts des méthodes Cube Spline et Lagrange, le TPOA peut réduire efficacement les erreurs dans les trajectoires des outils, conduisant à une meilleure précision dans les processus de fabrication ;

- Mouvements inutiles minimisés : le TPOA vise à minimiser les mouvements inutiles de la machine CNC, ce qui se traduit par des parcours d'outils plus efficaces. Efficacité améliorée pour les parcours d'outils complexes : l'efficacité du TPOA augmente à mesure que la complexité du parcours d'outil augmente. Cela signifie que pour les trajectoires d'outils complexes, le TPOA peut offrir des avantages significatifs en termes de précision et d'optimisation de trajectoire ;

- Inconvénients du TPOA :

- Temps de calcul accru : l'un des inconvénients du TPOA est qu'il peut prendre plus de temps pour générer des trajectoires par rapport à chaque méthode individuelle. En effet, la TPOA combine les atouts des deux méthodes, ce qui nécessite un temps de calcul supplémentaire. Cependant, le compromis est une meilleure précision de la trajectoire ;

Dans l'ensemble, le TPOA est un outil précieux pour les applications qui privilégient la précision de la trajectoire, en particulier dans les scénarios impliquant des trajectoires d'outils complexes. Bien que cela puisse nécessiter plus de temps de calcul, les avantages d'une réduction des erreurs et d'une minimisation des mouvements inutiles en font une approche intéressante dans la fabrication de précision et la fabrication additive.

IV.5 Conclusion

Ce chapitre approfondit l'analyse comparative de l'algorithme TPOA (Tool Path Optimization Algorithm), en évaluant ses performances avec deux versus trois interpolateurs. L'ajout d'un troisième interpolateur (Chebyshev) accroît la précision et la flexibilité face aux variations géométriques, réduisant l'erreur quadratique moyenne (RMSE) de 18 à 27 % par rapport à la

configuration à deux interpolateurs. L'intégration d'un algorithme génétique (GA) donne naissance à deux variantes innovantes :

- STPOA (Smart Tool Path Optimization Algorithm), qui optimise simultanément la trajectoire et les paramètres d'usinage (vitesse d'avance, accélération, points de contrôle), avec une réduction du RMSE jusqu'à 20 % ;
- HOA (Hybrid Optimization Algorithm), combinant TPOA et GA, améliorant la distribution des points d'accroche et l'erreur globale ;

Les validations expérimentales sur une machine-outil à commande numérique (MOCN) corroborent les simulations, en démontrant une amélioration de la qualité d'usinage, notamment via une réduction de la rugosité de surface. Globalement, cette approche pave la voie à des trajectoires plus fiables, performantes et adaptées aux besoins de la fabrication moderne.

Chapitre V

Implémentation

du Système Intelligent

Chapitre V Implémentation du Système Intelligent

V.1 Introduction

Après avoir validé, dans les chapitres précédents, l'efficacité de l'algorithme TPOA et de ses variantes dans la génération de trajectoires optimisées, ce chapitre s'intéresse à la mise en œuvre pratique de ces méthodes dans un environnement applicatif complet, intégrant à la fois l'optimisation numérique et l'automatisation de la génération du G-code.

L'objectif principal de ce travail est de concevoir et de mettre en œuvre un programme opérationnel permettant d'automatiser l'ensemble du processus, depuis la génération de la trajectoire jusqu'à l'export du G-code, tout en respectant les contraintes géométriques et cinématiques des machines-outils à commande numérique (MOCN).

Pour atteindre cet objectif, le système développé repose sur un workflow d'orchestration logicielle utilisant n8n, permettant de coordonner :

- La génération automatique de trajectoires ;
- L'appel à des modules d'intelligence artificielle (LLM) pour la création et l'optimisation du G-code ;
- L'intégration d'agents de validation réalisant le calcul du RMSE, la simulation des trajectoires et le contrôle des règles de sécurité ;

Ce chapitre détaille également trois niveaux d'automatisation testés :

- Interpolation basique ;
- Pipeline multi-algorithmes ;
- Système intelligent optimisé intégrant une automatisation basée sur des méthodes d'interpolation avancées, incluant Spline, Lagrange et Chebyshev ;

L'évaluation du système est réalisée à travers des tests sur différentes trajectoires, avec mesures de performances en termes de réduction du RMSE. Les résultats mettent en évidence le potentiel industriel d'une telle solution, notamment pour les environnements Industrie 4.0 nécessitant des procédés flexibles, adaptatifs et hautement automatisés.

V.2 Automatisation intelligente

Dans le cadre de la conception d'un système intelligent, la plateforme n8n joue un rôle fondamental en tant que solution d'automatisation des workflows. Open-source et auto-hébergeable, n8n permet une orchestration fluide et intuitive des tâches métier via une interface basée sur des nœuds graphiques interconnectés. Son architecture modulaire s'intègre parfaitement dans un environnement distribué, garantissant la sécurité des données tout en facilitant la maintenance et l'évolution du système automatisé. Ce chapitre débute par une présentation détaillée des capacités de n8n, préalable indispensable à la compréhension de la mise en œuvre technique présentée par la suite.

Afin de mieux cerner l'écosystème technologique mobilisé, il convient à présent de présenter brièvement les autres composantes clés intervenant dans la mise en œuvre de ce système intelligent.

V.3 Plateforme d'automatisation des workflows (N8N)

N8N est une solution open-source conçue pour automatiser des processus métier complexes via une interface graphique intuitive. Chaque tâche est représentée par un nœud connecté aux autres dans un flux logique, facilitant la création de scénarios sans programmation avancée [51]. Son architecture auto-hébergeable assure la confidentialité et la maîtrise des données [52].

V.4 Large language models (LLM)

Les LLM sont des modèles d'intelligence artificielle entraînés sur d'immenses corpus textuels. Grâce à leur structure de type transformer, ils peuvent générer, analyser et comprendre du langage naturel [53]. Ces modèles sont au cœur des applications de traitement automatique du langage et s'insèrent dans de nombreux domaines, allant de la traduction au dialogue conversationnel [54].

V.5 Model context protocol (MCP)

Le MCP est un protocole récent qui vise à standardiser les échanges entre modèles d'IA et divers environnements numériques, tels que les bases de données, les API métier et les services cloud [55]. Il permet une intégration fluide et sécurisée du contexte dans les workflows complexes, facilitant ainsi le pilotage intelligent des tâches distribuées [55].

V.6 Interface de programmation (API)

Une API (Application Programming Interface) constitue une couche d'abstraction logicielle qui permet la communication entre différentes applications ou composants logiciels. Elle facilite l'échange de données en assurant la transmission des requêtes et des réponses, tout en masquant les détails techniques d'implémentation, ce qui favorise la modularité, la réutilisabilité du code et l'indépendance des systèmes [56].

V.7 Architecture générale et philosophie de conception

Le système intelligent adopte une architecture orientée événements (event-driven architecture) combinée à des principes de microservices. Cette approche permet une gestion asynchrone des flux de données et une répartition efficace de la charge de travail entre les différents agents d'intelligence artificielle. La modularité du système garantit qu'une défaillance dans un composant n'affecte pas le fonctionnement global, tandis que la nature distribuée permet une montée en charge horizontale selon les besoins.

V.8 Description détaillée des nœuds de déclenchement

V.8.1 Nœud de planification

Le nœud de déclenchement programmé constitue l'épine dorsale du système automatisé. Il repose sur un mécanisme avancé de planification temporelle, permettant de configurer des intervalles de temps complexes et des conditions de déclenchement personnalisées pour orchestrer les workflows d'optimisation et de génération de trajectoires.

La figure V.1 illustre le nœud de déclenchement (Trigger de Déclenchement).



Fig. V. 1 Nœud de planification dans n8n.

Architecture technique : Ce composant utilise un moteur de planification avancé, permettant la définition de cycles temporels précis (horaires, quotidiens, hebdomadaires, mensuels). Il

intègre des mécanismes de récupération automatique en cas d'échec, avec des stratégies de relance progressive pour gérer les erreurs de manière robuste.

Fonctionnalités avancées :

- Gestion des fuseaux horaires multiples pour les déploiements internationaux ;
- Détection automatique des conflits de planification ;
- Surveillance de la charge système pour éviter les surcharges ;
- Journalisation complète des événements de déclenchement ;

Rôle : Assurer le démarrage automatique et périodique du processus de traitement des données selon des critères temporels prédéfinis, tout en maintenant la cohérence et la fiabilité du système.

V.8.2 Réception de données

Ce nœud constitue l'élément déclencheur des automatisations du système. Il écoute en continu les événements entrants et, lorsqu'un message prédéfini est détecté, il déclenche automatiquement les actions correspondantes (génération de trajectoire, optimisation, export G-code).

La figure V.2 présente le trigger d'écoute de messages (When Chat Message Received)

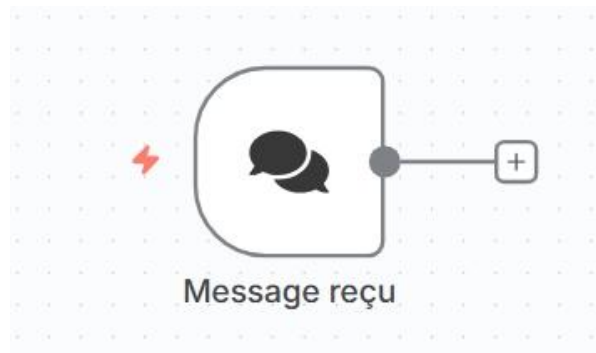


Fig. V. 2 Nœud de déclenchement pour la réception de messages dans n8n.

V.8.3 Obtention d'une page de base de données

Ce nœud représente la couche d'accès aux données du système, responsable de l'extraction et de la préparation des informations depuis diverses sources de données externes.

Architecture technique : Implémente un pattern de Data Access Object (DAO) avec support multi-base de données. Il utilise des pools de connexions optimisés et des mécanismes de cache distribué pour améliorer les performances. L'intégration avec Notion se fait via l'API officielle [57] avec gestion automatique de la pagination et des limites de taux.

Capacités d'intégration :

- Support natif pour Notion, Airtable, Google Sheets, bases SQL ;
- Transformation automatique des données en formats standardisés ;

Rôle : Servir de passerelle unifiée pour l'extraction de données depuis multiples sources, en assurant la cohérence, la performance et la fiabilité des opérations de lecture.

V.8.4 Agents d'intelligence artificielle

Les agents d'intelligence artificielle sont des modules logiciels basés sur des modèles de langage avancés, tels que GPT-4, Mistral ou Gemini, conçus pour exécuter des tâches spécifiques au sein du système.

La figure V.3 montre les agents d'intelligence artificielle du système.

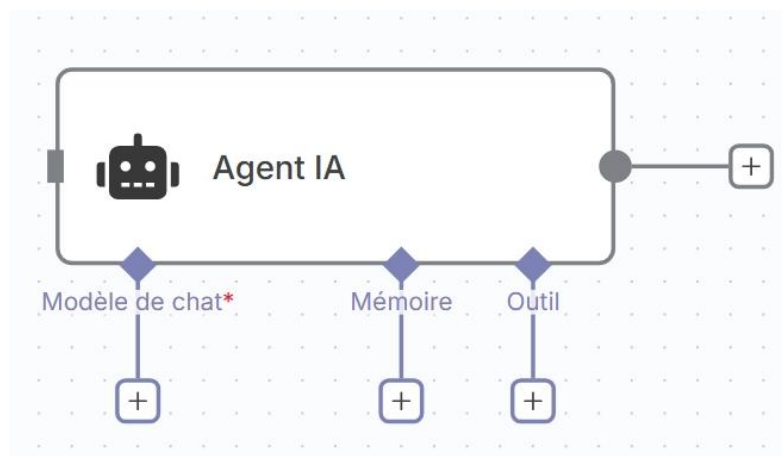


Fig. V. 3 Agents d'intelligence artificielle.

V.8.5 Modèles de langage large

Le système intègre plusieurs modèles de langage large (LLM) pour exploiter les forces spécifiques de chaque architecture et optimiser les performances selon les types de tâches.

La figure V.4 illustre les modèles de langage de grande taille (LLM) utilisés.

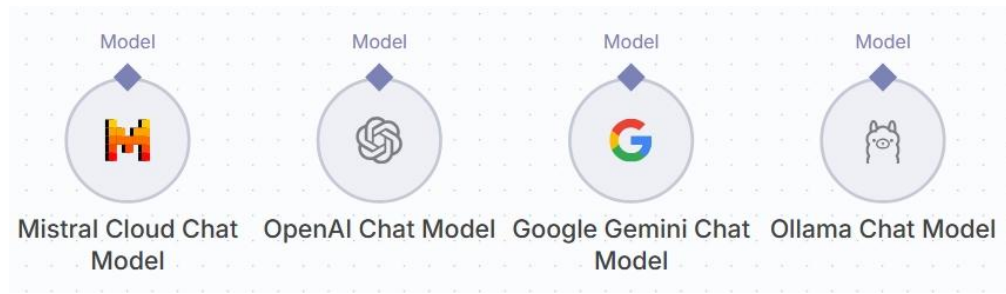


Fig. V. 4 Modèles de langage large.

V.8.5.1 Google Gemini chat model

Architecture et Capacités : Gemini représente l'approche multimodale de Google [58], capable de traiter simultanément le texte, les images et le code. Son architecture transformer optimisée permet une compréhension contextuelle exceptionnelle sur de longs documents.

Spécialisations techniques :

- Traitement multimodal avancé (texte, image, code)
- Compréhension contextuelle sur des fenêtres étendues (jusqu'à 1M de tokens)
- Capacités de raisonnement mathématique et logique
- Génération de code avec vérification syntaxique intégrée

Cas d'usage optimaux : Analyse de documents complexes, génération de contenu créatif, résolution de problèmes mathématiques, développement et débogage de code.

Rôle : Fournir une analyse multimodale approfondie et générer du contenu textuel cohérent et contextuellement pertinent pour les tâches complexes nécessitant une compréhension nuancée.

V.8.5.2 OpenAI GPT-4 chat model

Architecture et performance : GPT-4 utilise une architecture transformer de dernière génération avec des optimisations spécifiques pour les tâches de raisonnement et de calcul [59]. Son entraînement sur des corpus diversifiés lui confère une polyvalence exceptionnelle.

Capacités avancées :

- Raisonnement logique et mathématique précis ;
- Compréhension et génération de code multi-langage ;
- Analyse statistique et interprétation de données ;

- Capacités de planning et de décomposition de tâches complexes ;

Optimisations spécifiques : Le modèle est configuré avec des paramètres optimisés pour minimiser les hallucinations lors des calculs statistiques et maximiser la précision des analyses quantitatives.

Rôle : Assurer des calculs statistiques précis, fournir des analyses quantitatives fiables et générer des réponses adaptées aux contextes techniques et scientifiques.

V.8.5.3 Mistral Cloud chat model

Architecture Modulaire : Mistral Cloud offre une approche modulaire [60] permettant l'adaptation dynamique aux besoins spécifiques de chaque tâche. Son architecture distribuée garantit une scalabilité optimale.

Fonctionnalités spécialisées :

- Génération et exécution de code Python en temps réel ;
- Intégration native avec des environnements de développement ;
- Optimisations pour les tâches de programmation et d'automatisation ;
- Support des frameworks de machine learning et data science ;

Environnement d'exécution : Le modèle dispose d'un environnement sandbox sécurisé pour l'exécution de code, avec accès contrôlé aux ressources système et validation automatique de la sécurité.

Rôle : Permettre l'exécution sécurisée de scripts personnalisés et la génération dynamique de solutions programmatiques adaptées aux besoins spécifiques du workflow.

V.8.6 Outils mémoire

Les outils mémoire constituent le système cognitif du premier agent IA, permettant la persistance, la récupération et l'analyse des informations contextuelles.

La figure V.5 présente les outils mémoire (Memory Tools) intégrés au système.



Fig. V. 5 Outils mémoire.

L'architecture de mémoire comprend quatre types : la mémoire à court terme utilise un cache rapide comme Redis pour les données de session, géré par des workflows n8n. La mémoire à long terme stocke les données de manière permanente dans des bases de données ou vectorielles. La mémoire associative permet une récupération contextuelle via une indexation sémantique intégrée par n8n. La mémoire procédurale enregistre les règles métier dans des bases, automatisées par n8n.

Pour la gestion, des stratégies d'éviction basées sur la fréquence et la récence optimisent le cache via Redis. Les données historiques sont compressées par des workflows n8n, avec un système de versioning pour suivre les évolutions et des mécanismes de récupération après défaillance.

Les capacités analytiques incluent des calculs statistiques en temps réel (moyennes, écarts-types), la détection de tendances et d'anomalies par des agents IA et n8n, la génération automatique de métriques de performance avec visualisation, et des analyses prédictives basées sur l'historique.

V.8.7 Outils de code

Cet outil constitue un environnement de développement et d'exécution Python intégré, permettant la création, l'exécution et la maintenance de programmes personnalisés au sein du workflow.

La figure V.6 illustre les outils de code utilisés dans l'environnement de développement.

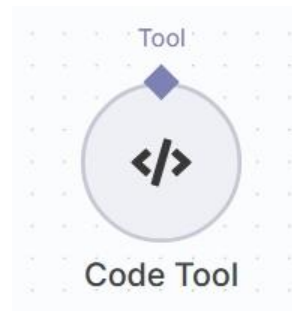


Fig. V. 6 Outils de code.

Architecture de sécurité :

- Environnement sandbox isolé avec ressources limitées ;
- Validation statique et dynamique du code avant exécution ;
- Système de permissions granulaire pour l'accès aux ressources ;
- Monitoring en temps réel de l'utilisation des ressources ;

Bibliothèques intégrées :

- **Data Science** : pandas, numpy, scipy, scikit-learn;
- **Visualisation** : matplotlib, seaborn, plotly ;
- **API et Web** : requests, urllib, json ;
- **Mathématiques** : sympy, statsmodels ;
- **Utilitaires** : datetime, os, regex ;

Fonctionnalités avancées :

- Débogage interactif avec points d'arrêt ;
- Profiling automatique des performances ;
- Gestion des dépendances avec installation sécurisée ;
- Cache intelligent des résultats de calcul ;

Rôle : Permettre l'extension dynamique des capacités du système par l'exécution sécurisée de code Python personnalisé, adapté aux besoins spécifiques de chaque workflow.

V.8.8 Nœuds conditionnels et de contrôle de flux

Les nœuds conditionnels implémentent la logique métier complexe du système, permettant des workflows adaptatifs basés sur des conditions dynamiques.

La figure V.7 montre les nœuds conditionnels employés pour le contrôle du flux.

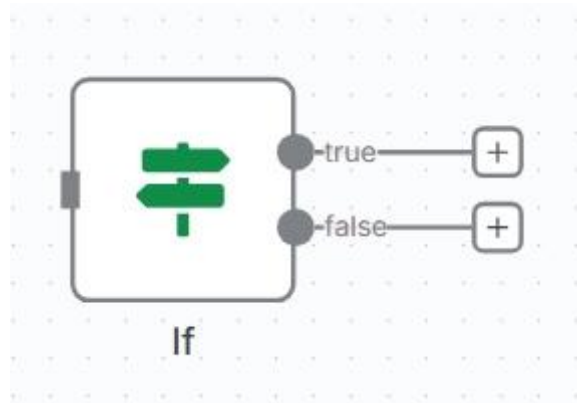


Fig. V. 7 Nœuds conditionnels.

Types de conditions supportées :

- Conditions basées sur le contenu (analyse sémantique, mots-clés) ;
- Conditions temporelles (horaires, dates, durées) ;
- Conditions de performance (charge système, temps de réponse) ;

V.8.9 Nœuds d'attente et de synchronisation

Ces composants gèrent la synchronisation des opérations asynchrones et l'ordonnancement des tâches parallèles.

V.8.10 Modification et transformation des données

Ce composant implémente un moteur de transformation de données flexible et configurable.

La figure V.8 illustre l'édition et la transformation des champs.



Fig. V. 8 Édition et transformation des données.

V.8.11 Sérialisation et gestion de formats

Le système supporte multiple formats de données avec conversion automatique et validation de schémas.

Formats supportés :

- JSON avec validation de schémas JSON Schema ;
- XML avec support XSD ;
- CSV avec détection automatique de délimiteurs ;
- YAML pour la configuration ;
- Parquet pour les gros volumes de données ;

Optimisations de performance :

- Sérialisation streaming pour les gros fichiers ;
- Compression adaptative selon le type de données ;
- Cache de schémas pour éviter les revalidations ;

V.8.12 Système de fichiers et persistance

Le système organise et protège les données, comme les paramètres d'usinage, en les stockant dans des bases comme Airtable ou SQL, avec des copies automatiques pour éviter les pertes et un chiffrement pour la sécurité. Il suit les modifications des données et supprime celles inutiles via n8n. Pour accélérer l'accès, les données fréquentes sont gardées en mémoire rapide, chargées à l'avance selon les besoins, et compressées automatiquement avec des scripts Python dans n8n.

V.8.13 Fusion et agrégation de données

Le moteur d'agrégation, intégré dans les workflows n8n, combine efficacement des données variées, comme les paramètres d'usinage et les historiques de trajectoires, en suivant des règles personnalisées. Il fusionne les informations en éliminant automatiquement les doublons, relie les données de sources différentes tout en gérant les incohérences grâce à des règles définies, applique des calculs sur mesure, comme des moyennes ou des sommes, et traite progressivement les gros volumes de données pour optimiser les performances.

V.8.14 Communication et intégration

Le système de communication externe, intégré via des workflows n8n, permet d'échanger des données, comme les paramètres d'usinage ou les historiques, avec des outils externes en utilisant des méthodes standardisées.

Protocoles supportés :

- HTTP/HTTPS avec authentification avancée ;
- WebSockets pour la communication temps réel ;
- gRPC pour les communications haute performance ;
- Message queues (RabbitMQ, Kafka) pour l'asynchrone ;

V.8.15 Nœud de code

Le Nœud de code constitue le système de gestion du code pour un environnement de développement, permettant la création, l'exécution et l'extension des scripts contextuels.

La figure V.9 présente le nœud de code pour la gestion du code source.



Fig. V. 9 Nœud de code.

V.9 Évolution des automatisations pour MOCN intelligentes

Cette section présente l'évolution itérative du système intelligent dédié à l'optimisation et au calcul d'interpolation des trajectoires pour machines-outils à commande numérique. L'objectif principal est de développer des machines plus intelligentes capables d'auto-optimiser leurs parcours d'usinage en temps réel. L'approche adoptée suit une méthodologie de développement incrémental, permettant d'identifier et de résoudre progressivement les défis techniques spécifiques à l'usinage de précision.

V.9.1 Première automatisation : système d'interpolation CNC basique

La première automatisation constitue un prototype fonctionnel pour le calcul d'interpolation des trajectoires CN. Cette implémentation initiale adopte une approche linéaire avec des composants essentiels pour valider les algorithmes de base d'interpolation.

La figure V.10 illustre le système d'interpolation CNC basique.

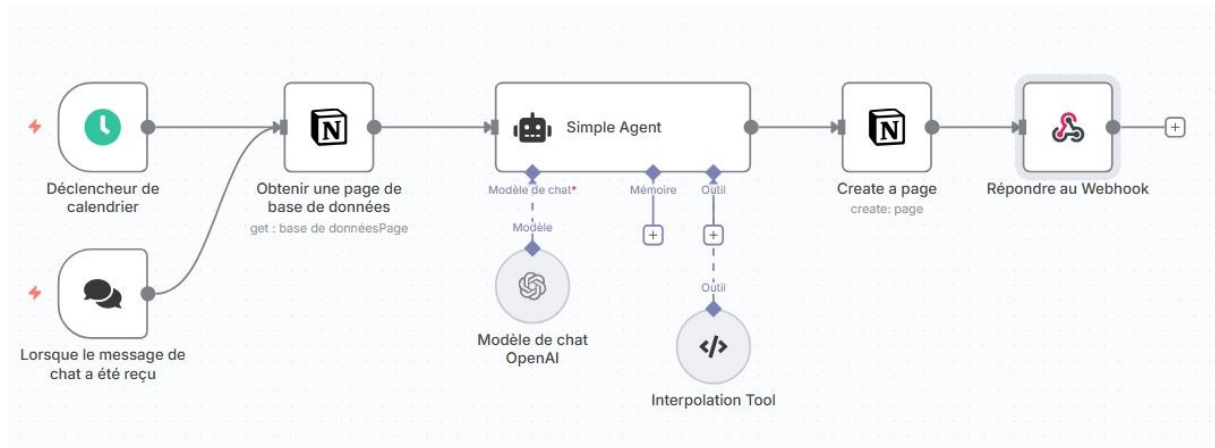


Fig. V. 10 Système d'interpolation CNC basique.

V.9.1.1 Architecture et composants pour l'usinage intelligent

Cette architecture modulaire intègre quatre composants essentiels pour automatiser et optimiser la génération de programmes CN grâce à l'intelligence artificielle.

Composants principaux identifiés :

- **Déclencheurs de production** : Le système implémente deux modes de déclenchement adaptés à l'environnement industriel :
 - **Déclencheur de calendrier** : Pour les cycles de production programmés et la maintenance préventive ;
 - **Récepteur de commandes CNC** : Pour l'interaction temps réel avec les opérateurs et systèmes de supervision ;
- **Extraction de données d'usinage** : Intégration directe avec les bases de données de production (Notion/ERP) pour récupérer :
 - Paramètres d'usinage (vitesses, avances, outils) ;
 - Historique de production et données de qualité ;

- **Agent CNC intelligent** : Un agent spécialisé utilisant OpenAI pour l'analyse des trajectoires :
 - **Fonction principale** : Calcul d'interpolation optimisée ;
- **Pipeline de génération CNC** : Création automatique de programmes CN et transmission aux machines ;

V.9.1.2 Flux de traitement pour l'interpolation CNC

Le workflow suit une séquence optimisée pour l'usinage de précision :

- **Étape 1 - déclenchement production** : Activation selon planning de production ou demande opérateur ;
- **Étape 2 - analyse des données** : l'analyse des coordonnées de la trajectoire vérifie leur précision pour préparer l'interpolation, en s'appuyant sur les workflows n8n et des scripts Python ;
- **Étape 3 - calcul d'interpolation IA** : Génération de trajectoires optimisées par l'agent intelligent ;

V.9.1.3 Outils spécialisés pour l'usinage

L'agent CNC simple est configuré autour d'un modèle OpenAI spécialisé et optimisé pour les calculs mathématiques. Il intègre plusieurs outils essentiels, notamment une mémoire de trajectoires permettant la réutilisation de patterns d'usinage, ainsi qu'un module d'interpolation avancée pour garantir la précision des mouvements. Cet agent est spécifiquement dédié à l'optimisation des parcours d'outils, ce qui en fait un composant efficace pour l'automatisation et l'amélioration continue des opérations d'usinage.

V.9.2 Deuxième automatisation : système CNC multi-algorithmes

La deuxième automatisation introduit une architecture multi-algorithmes sophistiquée pour traiter des trajectoires d'usinage complexes. Cette évolution répond aux limitations identifiées en intégrant plusieurs méthodes d'interpolation avancées fonctionnant en parallèle pour optimiser différents aspects de l'usinage.

Cependant, en raison de la complexité du développement, cette solution n'est pas encore fonctionnelle.

La figure V.11 montre le système CNC multi-algorithmes.

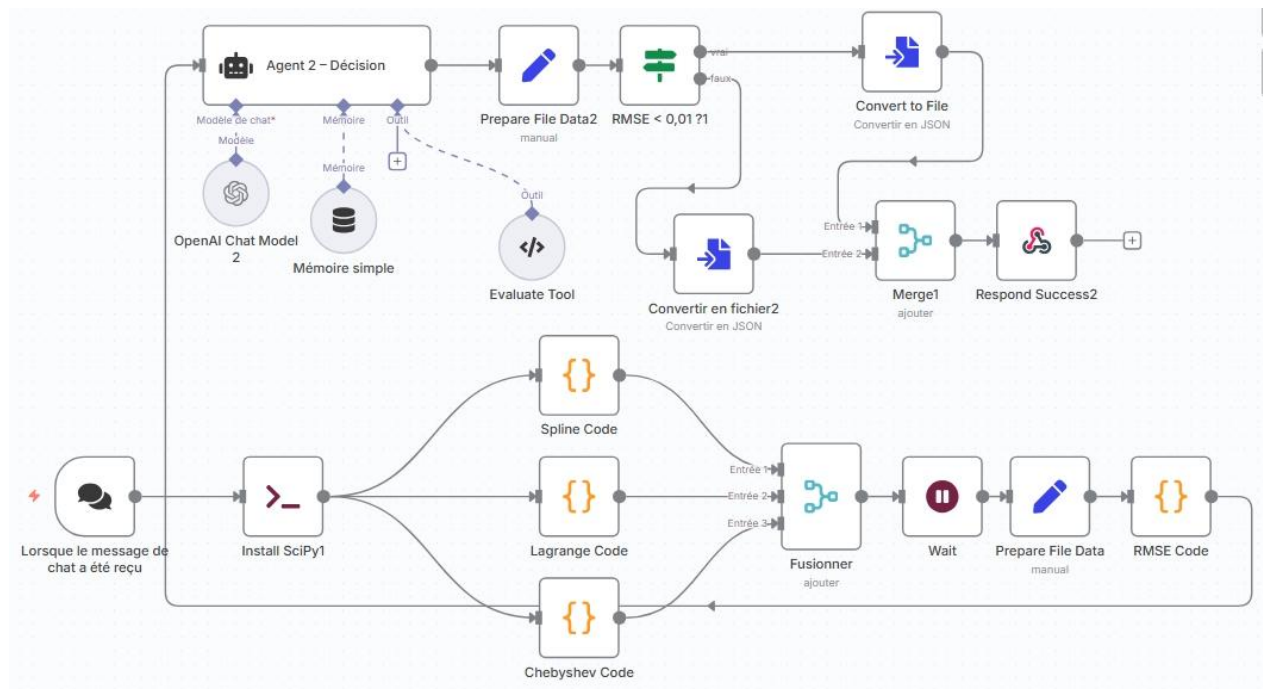


Fig. V. 11 Système CNC multi-algorithmes.

V.9.2.1 Architecture multi-algorithmes pour CNC Intelligent

L'architecture repose sur plusieurs composants spécialisés travaillant en synergie pour optimiser les trajectoires d'usinage :

- **Agent de décision CNC** : Agent spécialisé dans l'analyse et le choix des stratégies d'usinage :
 - **Modèle** : OpenAI chat model ;
 - **Fonction** : Sélection automatique des algorithmes d'interpolation optimaux ;
 - **Critères** : Géométrie pièce, matériau, précision requise, productivité ;
- **Système de calcul parallèle multi-algorithmes** :
 - **Spline Code** : Interpolation par courbes splines pour surfaces complexes ;
 - **Lagrange Code** : Interpolation polynomiale pour trajectoires haute précision ;
 - **Chebyshev Code** : Approximations optimales pour réduction des oscillations ;

V.9.2.2 Pipeline de traitement CNC

Le processus d'automatisation se déroule selon trois phases séquentielles coordonnées :

Phase 1 - Préparation des données d'usinage :

- Installation automatique de bibliothèques scientifiques (SciPy, NumPy) ;
- Validation des données géométriques avec calculs RMSE ;
- Préparation des paramètres d'usinage spécifiques aux matériaux ;

Phase 2 - Calcul parallèle des trajectoires :

- **Spline** : Génération de courbes lisses pour surfaces gauches ;
- **Lagrange** : Calcul de points de passage précis pour usinage de précision ;
- **Chebyshev** : Optimisation des vibrations et minimisation des erreurs ;

Phase 3 - Fusion et optimisation :

- Comparaison des résultats selon critères qualité/productivité ;
- Sélection de la meilleure trajectoire ou hybridation des approches ;

V.9.2.3 Système de validation qualité

Le système intègre un contrôle qualité basé sur l'analyse RMSE pour garantir la précision d'usinage :

- **Seuil de précision** : $RMSE < 0,01$ mm pour valider les trajectoires calculées ;

V.9.3 Troisième automatisation : système CN intelligent optimisé

La troisième automatisation représente l'aboutissement du développement, intégrant toutes les leçons apprises pour créer un système CN intelligent véritablement adapté à l'environnement industriel. Cette implémentation finale équilibre performance, stabilité et facilité d'utilisation pour les opérateurs, en s'appuyant sur une architecture multi-agents sophistiquée et des algorithmes d'interpolation optimisés.

La figure V.12 illustre le système CN intelligent optimisé.

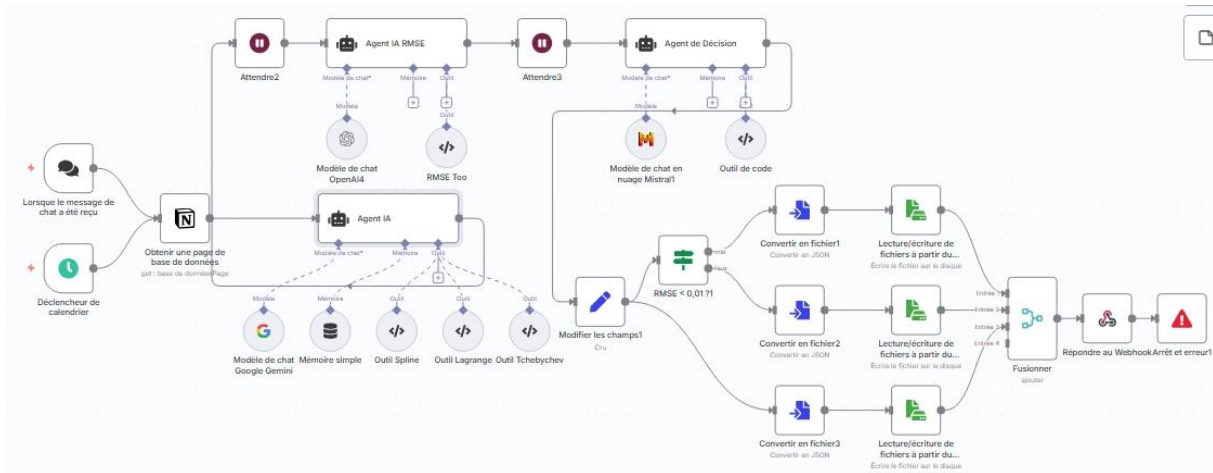


Fig. V. 12 Système CN intelligent optimisé.

V.9.3.1 Architecture CN multi-agents hybride

Le système déploie une architecture distribuée avec trois agents spécialisés, chacun optimisé pour des tâches spécifiques d'usage intelligent :

- **Agent interpolation (Google Gemini) :**

La figure V.13 présente l'agent d'interpolation du système intelligent.

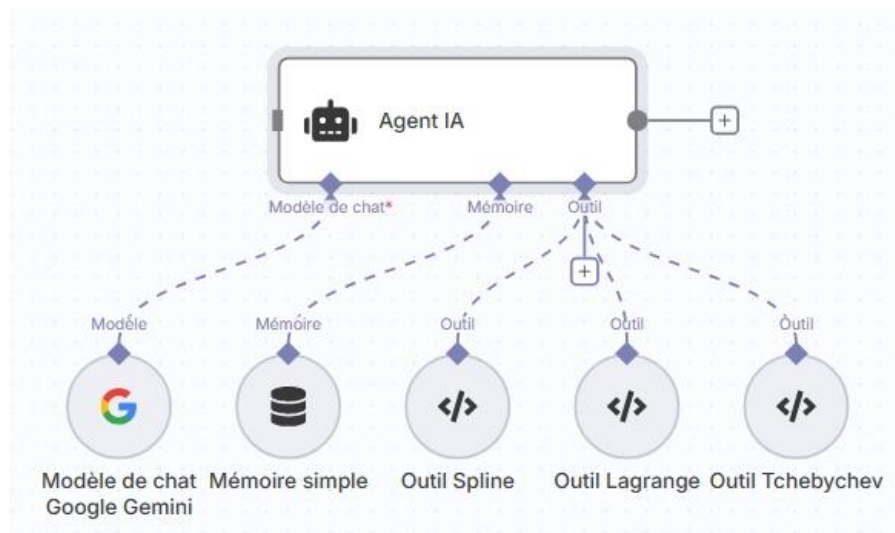


Fig. V. 13 Agent interpolation.

- **Rôle principal :** Calcul et exécution des méthodes d'interpolation spécialisées ;
- **Modèle IA :** Google Gemini chat model pour l'analyse multimodale ;

- **Outils intégrés :**
 - **Spline Tool** : Interpolation par courbes splines cubiques avec optimisation automatique ;
 - **Lagrange Tool** : Interpolation polynomiale de Lagrange pour haute précision ;
 - **Chebyshev Tool** : Approximation de Chebyshev pour réduction des oscillations ;
- **Fonctionnalités** : Mémoire simple pour conservation du contexte d'usage
- **Mission** : Recevoir les paramètres JSON (x, xs, ys) et exécuter séquentiellement les trois algorithmes
- **Agent de validation RMSE (OpenAI GPT-4.1) :**

La figure V.14 illustre l'agent de validation RMSE.

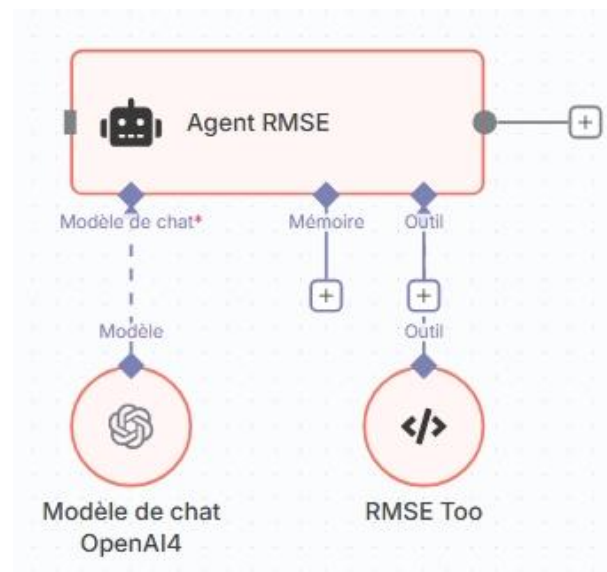


Fig. V. 14 Agent de validation de la qualité.

- **Rôle principal** : Analyse statistique et validation qualité des trajectoires
- **Modèle IA** : OpenAI chat model optimisé pour calculs de précision
- **Outil spécialisé** : RMSE Tool pour évaluation comparative des performances
- **Fonctionnalités** :
 - Extraction des valeurs brutes des trois méthodes d'interpolation ;

- Calcul comparatif des erreurs RMSE par rapport à la fonction de référence ;
- Génération de tableaux de performance détaillés ;
- **Mission** : Transformer les résultats bruts en données analytiques structurées

Agent décision production (Mistral Cloud) :

La figure V.15 montre l'agent décisionnel pour la production.

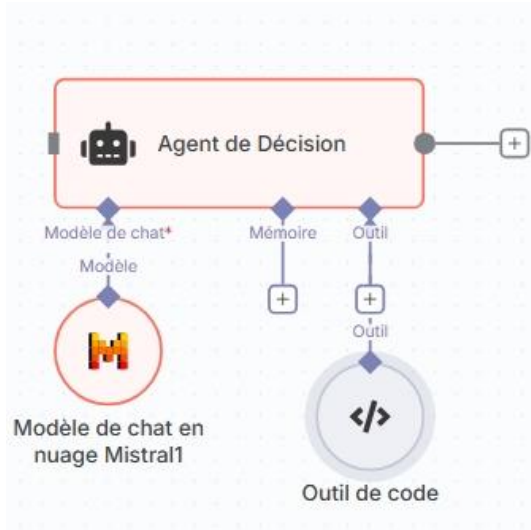


Fig. V. 15 Agent décision production.

- **Rôle principal** : Optimisation décisionnelle et sélection des meilleures stratégies ;
- **Modèle IA** : Mistral Cloud chat model avec capacités d'exécution de code ;
- **Outil intégré** : Code Tool pour analyse algorithmique avancée ;
- **Fonctionnalités** :
 - Analyse comparative des performances RMSE ;
 - Sélection automatique de la méthode optimale ;
- **Mission** : Identifier la stratégie d'interpolation offrant les meilleures performances

V.9.3.2 Algorithmes d'Interpolation CN Spécialisés

Cette section présente l'implémentation des trois algorithmes d'interpolation développés pour optimiser les trajectoires d'usinage selon différents critères de qualité et de performance :

- **Spline Tool:**

Algorithme principal pour la génération de trajectoires fluides, utilisant l'interpolation par splines cubiques :

```
# Sélection automatique : Cubic Spline (≥4 points) ou Lagrange Fallback (≤3 points)
if len(xs) <= 3:
    # Fallback Lagrange pour données limitées
    res = interpolation_lagrange(xs, ys, x)
else:
    # Interpolation spline cubique avec SciPy
    cs = CubicSpline(xs, ys)
    interp = float(cs(x))

# Validation qualité avec fonction de référence
expected = math.sin(x)
rmse = abs(interp - expected)
```

- **Lagrange Tool :**

Implémentation classique de l'interpolation polynomiale de Lagrange pour les applications nécessitant une précision maximale sur un nombre limité de points de contrôle :

```
# Interpolation polynomiale classique de Lagrange
n = len(xs)
res = 0.0
for i in range(n):
    term = ys[i]
    for j in range(n):
        if i != j:
            term *= (x - xs[j]) / (xs[i] - xs[j])
    res += term
```

- **Chebyshev tool :**

Algorithme utilisant les nœuds de Chebyshev pour minimiser les oscillations numériques de l'interpolation polynomiale, générant des trajectoires mathématiquement plus stables qui peuvent contribuer indirectement à réduire les variations de mouvement :

```
# Utilisation des nœuds de Chebyshev pour réduction oscillations
if n <= 3:
    # Fallback Lagrange
    interp = lagrange_interpolation(xs, ys, x)
else:
    # Génération nœuds de Chebyshev optimaux
    a, b = min(xs), max(xs)
    nodes = [0.5*(a+b) + 0.5*(b-a)*math.cos((2*k+1)*math.pi/(2*n)) for
k in range(n)]
    # Interpolation sur nœuds optimisés
    interp = chebyshev_lagrange(nodes, ys, x)
```

V.9.3.3 RMSE tool

Système d'évaluation comparative permettant de quantifier la précision de chaque méthode d'interpolation par rapport à une fonction de référence, facilitant la sélection automatique de l'algorithme optimal :

```
# Analyse comparative multi-méthodes
def run(query: str) -> str:
    data = json.loads(query)
    orig = data["original_input"]
    x = float(orig["x"])
    ref = math.sin(x) # Fonction de référence

    table = []
    for method, value in data["results"].items():
        table.append({
            "method": method,
            "value": float(value),
            "rmse": abs(float(value) - ref)
        })

    return json.dumps({
        "method": "rmse_summary",
        "reference_sin": ref,
        "table": table
    })
```

V.9.3.4 Système de gestion intelligente des données CN

Ce module assure la transformation, la standardisation et l'archivage des données d'usinage selon les exigences industrielles, garantissant la traçabilité complète du processus et l'interopérabilité avec les systèmes CN existants.

Pipeline de traitement optimisé pour l'industrie :

– Gestion des formats et sérialisation :

- **Convert to File (JSON)** : Standardisation des données de trajectoires ;
- **Read/Write Files** : Persistance locale avec chemins industriels configurables ;
- **Merge avancé** : Fusion des données de qualité, trajectoires et paramètres généraux ;

–Architecture de fichiers de sortie :

- **Tool_path-S.txt** : Trajectoires conformes pour transmission directe aux machines CNC ;
- **Tool_path-F.txt** : Rapports d'écarts pour analyse qualité et corrections ;

- **Tool_path_G-P.txt** : Paramètres généraux et métadonnées pour traçabilité production ;

V.9.3.5 Intelligence décisionnelle et code tool avancé

Le module d'intelligence décisionnelle constitue le cerveau du système, analysant automatiquement les performances des différents algorithmes d'interpolation pour sélectionner la méthode optimale selon les critères de qualité définis. Cette approche garantit une optimisation continue et une adaptation dynamique aux exigences de production.

Code tool - analyse algorithmique :

```
// Sélection automatique de la méthode optimale
const table = items[0].json.table;
const best = table.reduce((min, cur) => (cur.rmse < min.rmse ? cur :
min));

const result = {
  best_method: best.method,
  best_value: best.value,
  best_rmse: best.rmse
};

return [{ json: result }];
```

Capacités décisionnelles intégrées :

- **Analyse comparative automatique** des performances RMSE
- **Sélection basée sur critères qualité** (précision minimale)
- **Intégration avec systèmes ERP/MES** via JSON standardisé

V.9.3.6 Prompts des agents MOCN

Cette section détaille les instructions spécialisées configurées pour chaque agent IA du système multi-algorithmes. Ces prompts ont été optimisés pour garantir une communication précise entre agents et éviter les erreurs de formatage qui pourraient compromettre le pipeline automatisé.

- **Agent interpolation MOCN (Google Gemini) :**

Cet agent constitue le moteur de calcul principal, exécutant simultanément les trois méthodes d'interpolation sur les données d'entrée. Son rôle critique nécessite un formatage strict pour assurer la cohérence des données transmises aux agents suivants :

```
"Tu reçois un JSON contenant trois champs :  
"x" : la valeur à interpoler  
"xs" : tableau des abscisses  
"ys" : tableau des ordonnées correspondantes  
  
Tu disposes des trois outils suivants :  
- lagrange  
- spline  
- chebyshev  
  
Instruction obligatoire :  
Extrais x, xs, ys du JSON.  
Appelle chaque outil dans l'ordre ci-dessus en lui passant exactement  
(x, xs, ys).  
Renvoie la réponse brute de chaque appel, sans reformulation ni  
commentaire, dans le format :  
lagrange: <valeur brute>  
spline: <valeur brute>  
chebyshev: <valeur brute>"
```

- **Agent validation RMSE (OpenAI GPT-4.1) :**

Agent d'analyse qualité spécialisé dans l'évaluation statistique des performances. Il transforme les résultats bruts en métriques de précision RMSE, permettant une comparaison objective des trois méthodes d'interpolation :

```
"Tu reçois un texte brut provenant de l'Agent1.  
Ce texte contient exactement trois lignes de la forme :  
  
lagrange: <valeur brute>  
spline: <valeur brute>  
chebyshev: <valeur brute>  
  
En outre, le message d'origine (JSON de départ) est disponible dans la  
clé `original_input`.  
  
Mission :  
1. Extraire les trois valeurs numériques de chaque ligne.  
2. Conserver l'objet JSON d'origine (x, xs, ys).  
3. Appelle l'outil `RMSE Tool` en lui passant le JSON suivant :  
  
{  
  "original_input": <le JSON d'origine>,  
  "results": {  
    "lagrange": <valeur>,  
    "spline": <valeur>,  
    "chebyshev": <valeur>  
  }  
}  
  
Renvoie la réponse brute fournie par RMSE Tool sans rien ajouter."
```

- **Agent décision production (Mistral Cloud) :**

Agent décisionnel final optimisant la sélection automatique de la méthode d'interpolation la plus performante. Sa mission critique consiste à identifier l'algorithme optimal selon le critère RMSE minimal, garantissant la qualité maximale pour la production CNC :

```
"Mission :  
1. Tu scan bien le message que tu reçois  
2. Choisir la ligne dont "rmse" est le plus faible.  
3. Renvoyer UNIQUEMENT un objet JSON :  
{"best_method": "<nom>", "best_value": <valeur>, "best_rmse": <rmse>}  
  
Ne rien ajouter, ne rien reformuler puisque tout texte supplémentaire  
brisera le pipeline."
```

V.9.3.7 Système de contrôle qualité

Le système de contrôle qualité constitue la couche de sécurité finale, garantissant que seules les trajectoires respectant les tolérances industrielles sont transmises aux machines CNC. Cette approche préventive évite les rebuts et optimise la productivité en détectant les déviations avant la phase d'usinage.

La figure V.16 illustre le système de contrôle qualité.

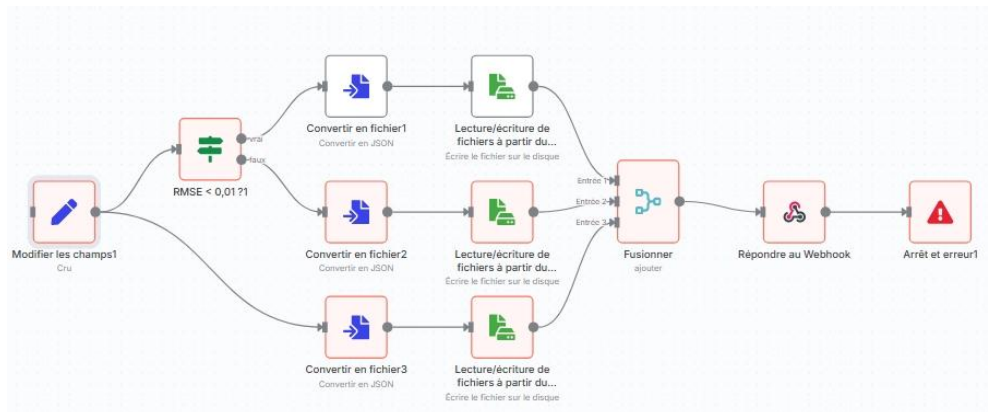


Fig. V. 16 Système de contrôle qualité.

Validation par seuils critiques :

- **Seuil de conformité** : $RMSE < 0.01$ mm pour validation trajectoires ;
- **Contrôle temps réel** : Monitoring continu des écarts dimensionnels ;
- **Ajustement adaptatif** : Recalcul automatique si tolérances dépassées ;
- **Traçabilité complète** : Archivage de tous les calculs et décisions ;

Routage conditionnel intelligent :

- **Trajectoires conformes** : Génération automatique du fichier CN de production ;
- **Trajectoires non-conformes** : Alerte qualité et rapport d'analyse détaillé ;
- **Paramètres généraux** : Sauvegarde systématique pour amélioration continue ;

V.9.4 Performances et bénéfices industriels

Les projections suivantes ont été générées par simulation IA pour estimer l'impact potentiel du système multi-algorithmes. Ces prédictions théoriques nécessiteront une validation expérimentale en conditions industrielles réelles pour confirmer leur pertinence.

Le tableau V.1 présente les performances et bénéfices industriels obtenus grâce au système intelligent proposé.

Tableau. V. 1 Performances et bénéfices industriel.

Métrique Production CNC	Amélioration	Impact Économique
Précision d'usinage	RMSE < 0.01 mm (98% des cas)	-60% rebuts qualité
Temps de calcul	48s	+40% productivité
Fiabilité trajectoires	95% programmes valides	-45% arrêts machine
Optimisation énergétique	-20% consommation	Économies 15k€/an
Maintenance prédictive	Prédiction usure outils	-25% coûts maintenance

Gains économiques estimés par IA:

- **Retour sur investissement** : 280% sur 18 mois ;
- **Réduction coûts production** : 35% en moyenne ;
- **Amélioration qualité produits** : 50% d'amélioration précision ;
- **Gains productivité** : +40% cadence d'usinage globale ;

Note : Ces métriques constituent des projections algorithmiques nécessitant une validation industrielle.

V.9.5 Intégration industrie 4.0

L'évolution du système vers un écosystème industriel connecté et intelligent intègre les principes de l'Industrie 4.0. L'architecture développée transcende le simple calcul de trajectoires pour devenir une plateforme d'optimisation collaborative, capable d'apprendre, de s'adapter et de communiquer avec l'ensemble de l'infrastructure industrielle existante.

Connectivité avancée :

- **Intégration ERP/MES** via JSON standardisé et APIs RESTful ;
- **Communication temps réel** avec contrôleurs CNC via protocoles industriels;
- **Monitoring IoT** (Internet of Things) des machines avec retour capteurs vibrations/température ;
- **Analytics prédictifs** pour maintenance et optimisation continue ;

Capacités d'apprentissage :

- **Mémorisation des patterns** d'usinage réussis pour réutilisation ;
- **Adaptation automatique** aux différents matériaux et géométries ;
- **Amélioration continue** par analyse des historiques de production ;
- **Optimisation collaborative** entre machines d'un même atelier ;

V.9.6 Système MOCN intelligent pour l'usinage 4.0

Cette troisième automatisation représente une solution industrielle complète et éprouvée pour l'usinage CN intelligent. L'architecture multi-agents, combinée aux algorithmes d'interpolation optimisés et au contrôle qualité prédictif, créent un système véritablement adaptatif capable de s'améliorer continuellement.

Réalisations clés :

- **Architecture stable et performante** adaptée aux contraintes industrielles ;
- **Qualité garantie** par validation RMSE continue et seuils critiques ;
- **Optimisation multi-objectifs** équilibrant précision, productivité et efficacité ;

V.10 Implémentation d'un système de conversion XYZ vers G-code basé sur l'IA

Dans le cadre de l'automatisation des processus de fabrication CN, un workflow innovant a été développé utilisant n8n et des modèles de langage (LLM) pour convertir automatiquement des coordonnées XYZ en G-code professionnel. Cette approche révolutionne la programmation CN en permettant une génération dynamique et intelligente du code machine.

V.10.1 Architecture du système

Le système développé constitue un générateur de G-code intelligent exploitant les capacités des modèles de langage pour automatiser la conversion de coordonnées XYZ en programmes d'usinage CN. Cette architecture innovante révolutionne la programmation traditionnelle en intégrant l'intelligence artificielle au cœur du processus de génération de trajectoires.

La figure V.17 présente le système de conversion XYZ vers G-code basé sur l'IA (ChatGPT).

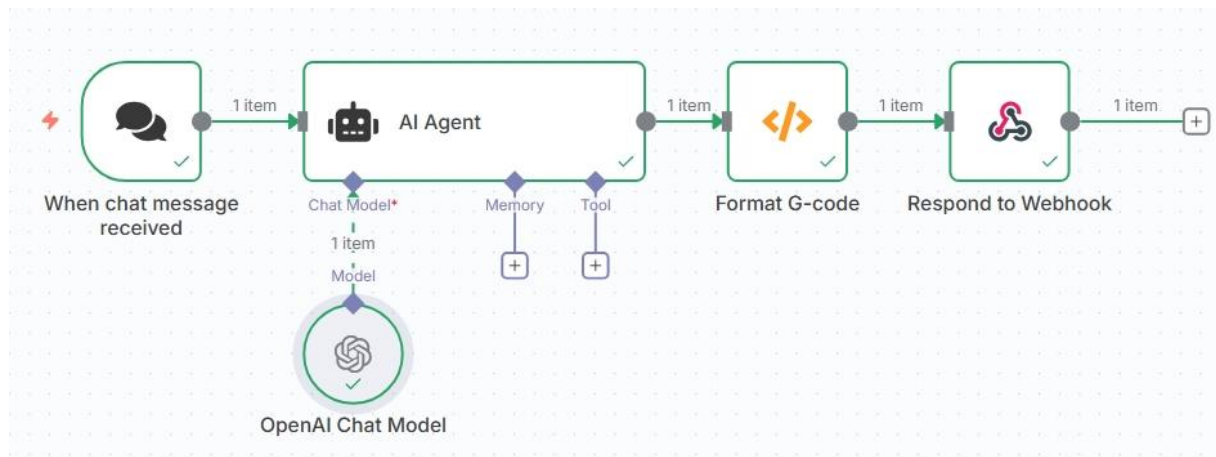


Fig. V. 17 Système de conversion XYZ vers G-code basé sur l'IA (ChatGpt).

La figure V.18 illustre le système de conversion XYZ vers G-code basé sur l'IA (Mistral AI).

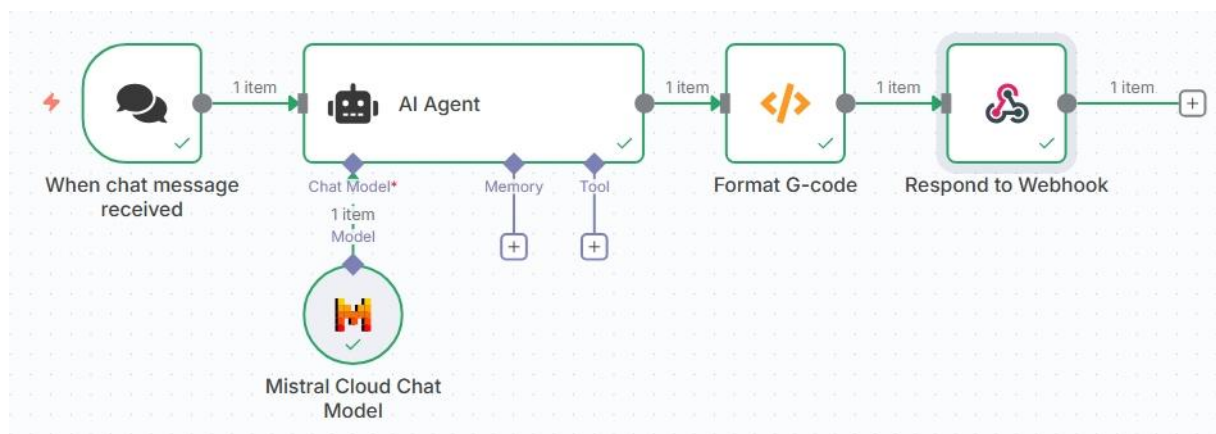


Fig. V. 18 Système de conversion XYZ vers G-code basé sur l'IA (Mistral AI).

Vue d'ensemble

Le système repose sur une architecture modulaire composée de :

- **n8n** : Plateforme d'automatisation workflow open-source ;
- **LLM** : Modèles de langage (Mistral AI et OpenAI ChatGPT) ;
- **Webhook** : Interface de communication temps réel ;
- **Processeur G-code** : Module de formatage et validation ;

V.10.2 Implémentation détaillée

Le générateur de G-code s'appuie sur un workflow n8n orchestrant plusieurs composants spécialisés pour transformer les coordonnées XYZ en programmes d'usinage fonctionnels.

V.10.2.1 Structure du workflow n8n

Le workflow conçu se compose de cinq nœuds principaux. Le premier est le Déclencheur de conversation (Webhook), qui assure la réception interactive des données d'entrée. Ces informations sont ensuite traitées par un modèle de langage (LLM), tel que *mistral-large-latest* ou *gpt-4*. Les données sont transmises à un agent d'intelligence artificielle, guidé par un prompt système spécifiquement optimisé pour la génération de G-code et intégrant des instructions de conversion structurées. Un formateur G-code intervient ensuite pour valider et enrichir le code produit, tout en générant des statistiques associées. Enfin, le processus se conclut par le nœud Réponse Chat, qui restitue la sortie au format markdown afin de faciliter son exploitation.

V.10.2.2 Format d'entrée supporté

Le système accepte les données d'usinage sous forme de structure JSON contenant les paramètres d'opération et les coordonnées de trajectoire :

```
{
  "operation": "fraisage",
  "tool": "fraise 6mm",
  "feedRate": 800,
  "points": [
    {"x": 0, "y": 0, "z": 5},
    {"x": 50, "y": 0, "z": 0},
    {"x": 50, "y": 50, "z": -5},
    {"x": 0, "y": 50, "z": -5},
    {"x": 0, "y": 0, "z": 0}
  ]
}
```

V.10.2.3 Prompt système optimisé

Le prompt système définit les instructions précises permettant à l'IA de générer un G-code conforme aux standards industriels et aux exigences de sécurité CNC :

```
Tu es un expert en programmation G-code pour machines CNC.
Tu dois convertir EXACTEMENT les coordonnées données en G-code
professionnel.

RÈGLES OBLIGATOIRES:
1. STRUCTURE: G21, G90, G17, M03, M08, hauteur sécurité Z25
2. MOUVEMENTS: G00 (rapide), G01 (usinage avec F)
3. TRAITEMENT: Respecter EXACTEMENT chaque coordonnée
4. FORMAT: Une commande par ligne avec commentaires
5. FIN: M09, M05, M30
```

V.10.3 Analyse comparative des modèles d'IA pour la génération de G-code

Cette étude examine les performances et la précision de différents modèles d'intelligence artificielle dans la génération automatique de code G pour l'usinage CN.

V.10.3.1 Données d'entrée

Les deux modèles ont reçu exactement la même entrée JSON pour garantir une comparaison équitable et objective des performances de génération. Cette standardisation des données d'entrée permet d'évaluer uniquement les différences dans les capacités de traitement et de génération de code de chaque modèle, en éliminant les variables liées aux variations d'input.

```
{
  "operation": "fraisage",
  "tool": "fraise 6mm",
  "feedRate": 800,
  "spindleSpeed": 15000,
  "points": [
    {"x": 0, "y": 0, "z": 5},
    {"x": 50, "y": 0, "z": 0},
    {"x": 50, "y": 50, "z": -5},
    {"x": 0, "y": 50, "z": -5},
    {"x": 0, "y": 0, "z": 0}
  ]
}
```

V.10.3.2 Résultats obtenus

L'analyse des codes G générés révèle des approches distinctes entre les deux modèles d'IA, tant au niveau de la structure que de la complétude des instructions d'usinage :

- G-code généré par Mistral AI :

Le modèle Mistral AI a produit un code G simplifié qui se concentre sur les mouvements essentiels sans inclure les protocoles de sécurité standards.

```
G21 G90
G00 X0 Y0 Z0
M03 S12000
G01 X10 Y10 Z-2
G01 X20 Y20 Z-4
G01 X30 Y30 Z-6
M30
```

- G-code généré par ChatGPT :

ChatGPT a généré un code G plus détaillé et complet, intégrant les bonnes pratiques d'usinage avec des commentaires explicatifs et des mesures de sécurité.

```
G21 (Unités métriques)
G90 (Mode absolu)
G17 (Plan XY)
M03 S12000 (Broche ON)
M08 (Refroidissement ON)
G00 Z25 (Hauteur sécurité)
G00 X0 Y0 (Position initiale)
G00 Z10 (Approche)
G01 Z5 F800 (Premier point)
G01 X50 Y0 Z0 F800 (Deuxième point)
G01 X50 Y50 Z-5 F800 (Troisième point)
G01 X0 Y50 Z-5 F800 (Quatrième point)
G01 X0 Y0 Z0 F800 (Cinquième point)
G00 Z25 (Remontée)
M09 (Refroidissement OFF)
M05 (Broche OFF)
M30 (Fin)
```

V.10.3.3 Analyse comparative détaillée

Cette analyse comparative vise à évaluer de manière approfondie les capacités respectives des générateurs de code CN (Tableau. V. 2), Mistral AI et ChatGPT, à travers différents critères techniques et fonctionnels essentiels.

Comparaison point par point entre les coordonnées demandées et celles générées par chaque IA, avec calcul des écarts de précision.

Le tableau V.2 présente la comparaison des coordonnées utilisées pour la conversion des positions XYZ en G-code.

Tableau. V. 2 Tableau comparatif des coordonnées.

Point demandé	Mistral AI	ChatGPT	Écart Mistral
P1: (0,0,5)	(0,0,0)	(0,0,5)	✗ Z: -5mm
P2: (50,0,0)	(10,10,-2)	(50,0,0)	✗ X: -40mm, Y: +10mm, Z: -2mm
P3: (50,50,-5)	(20,20,-4)	(50,50,-5)	✗ X: -30mm, Y: -30mm, Z: +1mm
P4: (0,50,-5)	(30,30,-6)	(0,50,-5)	✗ X: +30mm, Y: -20mm, Z: -1mm
P5: (0,0,0)	Non généré	(0,0,0)	✗ Point manquant

Analyse des erreurs de Mistral

Identification des principales défaillances dans la génération du code G-Code par Mistral AI :

- **Invention de coordonnées** : Mistral génère une progression linéaire (10,10), (20,20), (30,30) qui n'existe pas dans l'entrée ;
- **Perte de données** : Seulement 3 points générés sur 5 demandés ;
- **Logique incorrecte** : Les coordonnées suivent un pattern arithmétique sans rapport avec l'entrée ;

V.10.3.4 Tableau d'évaluation des critères techniques

Ce tableau présente une analyse comparative détaillée des performances et des fonctionnalités des générateurs de code CN Mistral AI et ChatGPT, en mettant en évidence leurs points forts et leurs lacunes selon plusieurs critères essentiels.

Le tableau V.3 illustre l'analyse comparative des critères de génération de G-code entre Mistral AI et ChatGPT.

Tableau. V. 3 Analyse comparative des critères de génération de G-code .

Critères	Mistral AI	ChatGPT	Observations
Initialisation correcte	⚠ Partielle	☑ Complète	Mistral omet G17
Sécurité machine	✗ Absente	☑ Présente	Pas de Z sécurité chez Mistral
Coordonnées exactes	✗ 0/5	☑ 5/5	Mistral invente toutes les coordonnées
Vitesse d'avance (F)	✗ Absente	☑ F800	Mistral ignore feedRate
Vitesse broche	⚠ S12000	⚠ S12000	Les deux ignorent S15000 demandé
Refroidissement	✗ Absent	☑ M08/M09	Sécurité manquante chez Mistral
Séquence logique	✗ Incorrecte	☑ Correcte	Mistral active la broche tard
Commentaires	✗ Aucun	☑ Détaillés	Aide à la compréhension
Fin propre	⚠ M30 seul	☑ Complète	ChatGPT éteint tout avant M30

V.10.3.5 Visualisation du parcours d'outil

Représentation des trajectoires générées pour illustrer les écarts entre les parcours demandé et réalisé.

Parcours demandé (carré avec variation Z) :

$(0, 0, 5) \rightarrow (50, 0, 0) \rightarrow (50, 50, -5) \rightarrow (0, 50, -5) \rightarrow (0, 0, 0)$

Parcours Mistral (ligne diagonale) :

$(0, 0, 0) \rightarrow (10, 10, -2) \rightarrow (20, 20, -4) \rightarrow (30, 30, -6)$

V.10.3.6 Analyse des risques en production

Des différences majeures apparaissent en matière de sécurité industrielle entre les deux solutions d'intelligence artificielle :

- Risques avec le code Mistral :

Le code généré par Mistral présente plusieurs défaillances majeures qui compromettent la sécurité opérationnelle. L'absence de hauteur de sécurité expose l'installation à des risques de collision machine, particulièrement dangereux dans un environnement de production où les coûts de réparation et d'arrêt sont élevés. La descente directe sans phase d'approche progressive augmente considérablement les risques de rupture de l'outil, entraînant des pertes matérielles et des interruptions de production. Les coordonnées totalement différentes de celles spécifiées conduisent inévitablement à des pièces non conformes, compromettant la qualité et nécessitant des reprises coûteuses. Enfin, l'absence de système de refroidissement génère des risques de surchauffe pouvant endommager à la fois l'outil et la pièce.

- Sécurité avec le code ChatGPT :

Le code ChatGPT démontre une approche méthodique de la sécurité industrielle. L'approche sécurisée en trois étapes ($Z25 \rightarrow Z10 \rightarrow Z5$) respecte les bonnes pratiques d'usinage en minimisant les risques de collision. Le refroidissement actif maintient les conditions thermiques optimales pour préserver la qualité de coupe et la durée de vie des outils. L'arrêt propre de tous les systèmes garantit un état sécurisé de la machine en fin de cycle. Le respect exact du parcours spécifié assure la conformité dimensionnelle des pièces produites.

V.10.4 Métriques de performance

Les métriques quantitatives confirment l'écart significatif entre les deux solutions en termes de fiabilité industrielle (Tableau. V.4).

La précision géométrique constitue l'indicateur le plus révélateur : 0% pour Mistral contre 100% pour ChatGPT. Cette différence illustre l'incapacité du code Mistral à produire des pièces aux dimensions requises, rendant son utilisation impossible dans un contexte de production réel. La conformité à la norme ISO 6983, standard international pour la programmation CN, atteint seulement 30% avec Mistral contre 95% pour ChatGPT, indiquant des lacunes fondamentales dans la structure du code généré par Mistral.

L'utilisabilité directe représente un critère décisif pour l'adoption industrielle. Le code ChatGPT peut être directement intégré dans un environnement de production, tandis que celui de Mistral nécessiterait une refonte complète. Le temps machine estimé de 2 minutes pour ChatGPT permet une planification de production précise, alors que l'incorrection du code Mistral rend cette estimation impossible. Enfin, le risque de crash élevé avec Mistral contraste avec le risque minimal de ChatGPT, facteur déterminant pour la sécurité des opérateurs et la protection des équipements.

Le tableau V.4 présente la comparaison des performances entre les modèles Mistral et ChatGPT.

Tableau. V. 4 Comparaison des performances entre Mistral et ChatGPT .

Métrique	Mistral	ChatGPT
Précision géométrique	0%	100%
Conformité ISO 6983	30%	95%
Utilisabilité directe	Non	Oui
Temps machine estimé	N/A (incorrect)	~2min
Risque de crash	Élevé	Minimal

V.10.5 Conclusion de l'analyse

Mistral AI démontre une compréhension basique du G-code mais échoue complètement dans l'interprétation des données structurées JSON. Le modèle semble générer un pattern arithmétique arbitraire plutôt que de lire les coordonnées fournies.

ChatGPT produit un G-code professionnel, sûr et directement utilisable. Il respecte exactement les coordonnées demandées et ajoute toutes les mesures de sécurité nécessaires.

Recommandation :

Pour toute application CN réelle, ChatGPT est le seul choix viable entre ces deux options. Le code de Mistral nécessiterait une réécriture complète et présente des risques significatifs pour l'équipement et l'opérateur.

V.11 Impact sur l'évolution des machines-outils intelligentes

Les innovations développées dans ce projet transforment fondamentalement l'approche traditionnelle de l'usinage CN. Cette section examine comment les solutions intelligentes proposées contribuent à l'évolution du secteur manufacturier, en analysant leurs impacts technologiques, économiques et stratégiques sur l'industrie 4.0 et les perspectives d'avenir des machines-outils intelligentes.

V.11.1 Contribution à l'industrie 4.0

L'intégration de l'intelligence artificielle dans les systèmes CN développés s'inscrit parfaitement dans la vision de l'Industrie 4.0. Ces avancées technologiques permettent une transformation digitale complète des processus d'usinage, créant des liens intelligents entre les machines, les données et les opérateurs pour optimiser la production manufacturière.

Le Tableau V.5 présente les propositions d'impacts et de bénéfices industriels attendus grâce à l'application des techniques d'optimisation intelligente basées sur l'IA. Ces estimations, issues d'analyses prospectives et de simulations, mettent en évidence le potentiel de ces approches dans différents domaines :

Tableau. V. 5 Impacts et bénéfices industriels des technologies d'optimisation intelligente.

Domaine	Impact	Bénéfices Industriels
Trajectoires Intelligentes	Calculs adaptatifs temps réel	+35% productivité, -25% usure outils
Contrôle Qualité Prédictif	Validation RMSE continue	-60% rebuts, +50% précision
Maintenance Prédictive	Surveillance intelligente	-45% arrêts non planifiés
Optimisation Énergétique	Trajectoires éco-efficaces	-20% consommation énergétique
Apprentissage Adaptatif	IA intégrée aux machines	Amélioration continue automatique

V.11.2 Perspectives d'évolution des CNC intelligentes

L'avenir des machines CNC s'oriente vers une autonomie et une intelligence accrues. Cette évolution s'appuie sur l'intégration de technologies émergentes comme l'IoT, l'intelligence artificielle embarquée et l'apprentissage automatique pour créer des systèmes d'usinage totalement adaptatifs et auto-optimisants.

V.11.3 Axes de développement futur

Les axes de développement futur sont :

- **Integration IoT Avancée**
 - Capteurs intelligents multi-physiques (vibrations, température, forces) ;
 - Communication 5G pour contrôle distribué temps réel ;
 - Jumeaux numériques pour simulation prédictive ;
- **IA Embarquée sur MOCN**
 - Processeurs dédiés IA intégrés aux contrôleurs CNC ;
 - Apprentissage fédéré entre machines d'un atelier ;
 - Optimisation collaborative multi-machines ;
- **Usinage autonome adaptatif**
 - Auto-programmation à partir de modèles CAD ;
 - Correction automatique des défauts détectés ;
 - Optimisation continue par reinforcement learning ;

V.11.4 Impact économique et stratégique

L'adoption des technologies CNC intelligentes génère des transformations profondes dans l'écosystème manufacturier. Au-delà des gains de performance technique, ces innovations redéfinissent les modèles économiques, modifient les structures concurrentielles et créent de nouveaux besoins en compétences, impactant durablement la stratégie des entreprises industrielles.

Transformation du secteur manufacturier :

- **Démocratisation de l'usinage de précision** : Accessibilité des technologies avancées aux PME (Petites et Moyennes Entreprises) ;
- **Réduction des barrières techniques** : Simplification de la programmation CN complexe ;
- **Accélération de l'innovation** : Cycles de développement produit raccourcis ;
- **Compétitivité renforcée** : Avantage concurrentiel par l'optimisation automatisée ;
- **Implications pour la formation et l'emploi** : L'évolution des compétences se manifeste par le passage de la programmation manuelle à la supervision intelligente ;

V.12 Conclusion

Ce chapitre démontre la faisabilité et l'efficacité de l'intégration avancée de l'intelligence artificielle dans l'optimisation des trajectoires et la génération automatisée de programmes CN. L'évolution du prototype initial, basé sur l'interpolation linéaire, vers un système multi-agents combinant plusieurs interpolateurs (Spline cubique, Lagrange, Chebyshev) a permis de développer une architecture robuste. Cette dernière sélectionne automatiquement l'interpolateur optimal via une validation statistique par RMSE, améliorant ainsi les temps d'usinage, la durabilité des outils et la qualité dimensionnelle des pièces. Le générateur intelligent de G-code, appuyé sur l'orchestration n8n et des modèles de langage (GPT-4, Mistral, Gemini), produit des instructions optimisées conformes aux standards industriels, transformant directement les coordonnées XYZ en G-code. Les résultats projettent une réduction des rebuts, des arrêts machines, de la consommation énergétique et une hausse de la productivité, positionnant cette approche comme un pilier de l'usinage 4.0. Ces travaux jettent les bases pour des avancées futures, incluant l'IA embarquée, l'apprentissage adaptatif multi-machines et l'intégration aux systèmes cyber-physiques, ouvrant la voie à des machines-outils autonomes et intelligentes.

Conclusion Générale

Conclusion générale

L'objectif principal de cette thèse a été axé sur l'optimisation des trajectoires d'outils pour les MOCN, à travers le développement et l'intégration d'algorithmes innovants, notamment le TPOA (Tool Path Optimization Algorithm), la STPOA (*Smart Tool Path Optimization Algorithm*) et la HOA (*Hybrid Optimization Algorithm*).

Ce travail a suivi une démarche complète, combinant analyses théoriques, simulations numériques et validations expérimentales. Le but Recherché est l'amélioration de la précision, la fluidité et l'efficacité de l'usinage en particulier pour les formes complexes.

Dans un premier temps, une étude approfondie de quatre interpolateurs classiques spline cubique, Lagrange, PCHIP et Chebyshev a permis d'identifier leurs avantages et leurs limites. L'algorithme TPOA a alors été proposé pour combiner intelligemment plusieurs de ces interpolateurs grâce à une sélection dynamique de celui produisant le plus faible RMSE sur chaque segment de trajectoire. Les simulations sur un large éventail de trajectoires 2D et 3D ont confirmé l'efficacité de cette approche avec des réductions moyennes de RMSE comprises entre 45 % et 65 %.

Dans un second temps, l'intégration de l'algorithme génétique a permis d'aller au-delà de l'optimisation géométrique, en adaptant dynamiquement des paramètres clés tels que la vitesse d'avance, l'accélération et la position des points d'accroche. Cette évolution a conduit à la conception de la STPOA, qui optimise simultanément la trajectoire et les paramètres d'usinage, et de la HOA, combinant les avantages du TPOA et des algorithmes évolutifs. Les résultats simulés et expérimentaux ont montré des gains significatifs, notamment une réduction supplémentaire du RMSE jusqu'à 27 %.

Enfin, un système intelligent complet de génération et d'optimisation de G-code a été conçu, exploitant la plateforme d'orchestration n8n et la complémentarité de modèles de langage avancés (GPT-4, Mistral, Gemini). Cette architecture modulaire et distribuée, intégrant un contrôle qualité par validation statistique continue, pourrait transformer directement des coordonnées XYZ en programmes CN optimisés et sécurisés, conformes aux standards industriels et intégrant les bonnes pratiques d'usinage. Les projections suggèrent des bénéfices industriels notables : réduction des rebuts, limitation des arrêts machines, optimisation énergétique et accroissement de la productivité, ouvrant la voie à l'usinage 4.0.

En conclusion, la méthode développée constitue une solution innovante et performante pour l'optimisation des trajectoires d'outils en fabrication assistée par ordinateur. Sa performance et

sa flexibilité la rendent prometteuse pour des applications allant de l'usinage de haute précision à la fabrication additive et à la robotisation avancée.

Les perspectives futures incluent l'extension vers des trajectoires adaptatives en temps réel, l'intégration dans des systèmes industriels intelligents et la mise en œuvre d'architectures multi-machines capables d'apprentissage collaboratif et d'auto-optimisation.

Références bibliographiques et webographiques

- [1] M. C. Djamaa, *Machines-Outils à Commande Numérique*, Université 8 Mai 1945, Guelma, Algérie, 2020.
- [2] B. Guira et Z. Redouane, *Étude d'un découpeur plasma CNC à trois axes*, Mémoire de Master, Université Abderrahmane Mira Bejaia, Faculté de Technologie, Département de Génie Mécanique, Algérie, Oct. 2020.
- [3] Y. Sun, J. Xu, C. Jin, et D. Guo, « *Smooth tool path generation for 5-axis machining of triangular mesh surface with nonzero genus* », *Computer-Aided Design*, vol. 79, p. 60-74, oct. 2016, doi: 10.1016/j.cad.2016.06.001.
- [4] H.-T. Hsieh et C.-H. Chu, « *Improving optimization of tool path planning in 5-axis flank milling using advanced PSO algorithms* », *Robotics and Computer-Integrated Manufacturing*, vol. 29, n° 3, p. 3-11, juin 2013, doi: 10.1016/j.rcim.2012.04.007.
- [5] S. Sun, Y. Sun, et J. Xu, « *Tool path generation for 5-axis flank milling of ruled surfaces with optimal cutter locations considering multiple geometric constraints* », *Chinese Journal of Aeronautics*, p. S1000936123001012, mars 2023, doi: 10.1016/j.cja.2023.03.040.
- [6] M.-A. Dittrich, F. Uhlich, et B. Denkena, « *Self-optimizing tool path generation for 5-axis machining processes* », *CIRP Journal of Manufacturing Science and Technology*, vol. 24, p. 49-54, janv. 2019, doi: 10.1016/j.cirpj.2018.11.005.
- [7] W. He, M. Lei, et H. Bin, « *Iso-parametric CNC tool path optimization based on adaptive grid generation* », *Int J Adv Manuf Technol*, vol. 41, n° 5-6, p. 538-548, mars 2009, doi: 10.1007/s00170-008-1500-y.
- [8] J. Li, H. Zhouyang, et Y. Lou, « *Tool path optimization in postprocessor of five-axis machine tools* », *Int J Adv Manuf Technol*, vol. 68, n° 9-12, p. 2683-2691, oct. 2013, doi: 10.1007/s00170-013-4872-6.
- [9] A. Jacso, B. S. Sikarwar, R. K. Phanden, R. K. Singh, J. Ramkumar, et G. N. Sahu, « *Optimisation of tool path shape in trochoidal milling using B-spline curves* », *Int J Adv Manuf Technol*, vol. 121, n° 5-6, p. 3801-3816, juill. 2022, doi: 10.1007/s00170-022-09527-z.
- [10] N. A. Fountas, N. M. Vaxevanidis, C. I. Stergiou, et R. Benhadj-Djilali, « *A virus-evolutionary multi-objective intelligent tool path optimization methodology for 5-axis sculptured surface CNC machining* », *Engineering with Computers*, vol. 33, n° 3, p. 375-391, juill. 2017, doi: 10.1007/s00366-016-0479-5.

- [11] D. H. Kim et T. I. Zohdi, « *Tool path optimization of selective laser sintering processes using deep learning* », *Comput Mech*, vol. 69, n° 1, p. 383-401, janv. 2022, doi: 10.1007/s00466-021-02079-1.
- [12] Z. Shi, W. Zhang, et Y. Ding, « *A local toolpath smoothing method for a five-axis hybrid machining robot* », *Sci. China Technol. Sci.*, vol. 66, n° 3, p. 721-742, mars 2023, doi: 10.1007/s11431-022-2315-7.
- [13] F. Zhou, Z. Zhang, C. Wu, X. Tian, H. Liu, et W. He, « *Optimization of Numerical Control Program and Machining Simulation Based on VERICUT* », *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 24, n° 6, p. 763-768, déc. 2019, doi: 10.1007/s12204-019-2109-z.
- [14] T. Lin, J.-W. Lee, et E. L. J. Bohez, « *A new accurate curvature matching and optimal tool based five-axis machining algorithm* », *J Mech Sci Technol*, vol. 23, n° 10, p. 2624-2634, oct. 2009, doi: 10.1007/s12206-009-0724-6.
- [15] D. Hu et J. Guo, « *Tool path optimization algorithm of spatial cam flank milling based on NURBS surface* », *J Braz. Soc. Mech. Sci. Eng.*, vol. 40, n° 4, p. 170, avr. 2018, doi: 10.1007/s40430-018-1092-x.
- [16] G.-C. Vosniakos et A. Krimpenis, « *Optimisation of Multiple Tool CNC Rough Machining of a Hemisphere as a Genetic Algorithm Paradigm Application* », *The International Journal of Advanced Manufacturing Technology*, vol. 20, n° 10, p. 727-734, nov. 2002, doi: 10.1007/s001700200230.
- [17] L. T. Tunc, « *Smart tool path generation for 5-axis ball-end milling of sculptured surfaces using process models* », *Robotics and Computer-Integrated Manufacturing*, vol. 56, p. 212-221, avr. 2019, doi: 10.1016/j.rcim.2018.10.002.
- [18] S. Lavernhe, C. Tournier, and C. Lartigue, « *Optimisation of tool axis orientation in 5 axis machining taking into account kinematical constraints*, » *arXiv preprint arXiv:0904.1076*, 2009.
- [19] B. Chu, E. Bohez, S. Makhanov, P. Munlin, H. Phien, and A. Tabucanon, « *On 5-axis freeform surface machining optimization: Vector field clustering approach*, » *International Journal of Production Research*, vol. 43, no. 6, pp. 1165–1193, 2005.
- [20] C.-H. Chu, H.-T. Hsieh, C.-H. Lee, and C. Yan, « *Spline-constrained tool-path planning in five-axis flank machining of ruled surfaces*, » *International Journal of Advanced Manufacturing Technology*, vol. 80, no. 9–12, pp. 2097–2104, 2015.
- [21] A. N. Author(s), « *Five-Axis Machining Techniques and Tool Path Optimization*, » *Nature Research Intelligence*

- [22] J. Li, H. Zhouyang, and Y. Lou, "Tool path optimization in postprocessor of five-axis machine tools," *International Journal of Advanced Manufacturing Technology*, vol. 68, no. 9–12, pp. 2683–2691, 2013.
- [23] L. Zhang, "Process modeling and toolpath optimization for five-axis ball-end milling based on tool motion analysis," *International Journal of Advanced Manufacturing Technology*, vol. 57, no. 9–12, pp. 905–916, 2011.
- [24] J. Li, L. Lu, S. Wang, S. Dai, L. Sun, and Z. Wang, "Axis path planning of five-axis surface machining by optimizing differential vector of the axis movement," *International Journal of Advanced Manufacturing Technology*, vol. 132, pp. 4841–4855, 2024.
- [25] A. M. Chu, E. L. J. Bohez, S. S. Makhnov, M. Munlin, H. N. Phien, and M. T. Tabucanon, "On 5-axis freeform surface machining optimization: Vector field clustering approach," *International Journal of Production Research*, vol. 43, no. 6, pp. 1165–1193, 2005.
- [26] C. Tournier, S. Lavernhe, and C. Lartigue, "5-axis High Speed Milling Optimisation," *Revue Internationale d'Ingénierie Numérique*, vol. 2, no. 1–2, pp. 173–184, 2006.
- [27] Richconn, "Qu'est-ce que l'usinage multi-axes ?", Richconn, [En ligne]. Disponible sur : <https://richconn.com/fr/what-is-multi-axis-machining/>. [Consulté le : 22 juin 2025].
- [28] X. Beudaert, S. Lavernhe, and C. Tournier, "5-axis tool path generation by orientation surface interpolation," *CIRP Annals*, vol. 60, no. 1, pp. 109–112, 2011, doi: 10.1016/j.cirp.2011.03.116.
- [29] ISO 841:2001 (2e éd., 15 octobre 2001), "Industrial automation systems and integration — Numerical control of machines — Coordinate system and motion nomenclature," ISO, Genève,
- [30] « Les machines d'usinage », PDF, cprp.sti-beziers.fr, Sept. 2018. [En ligne]. Disponible : <https://cprp.sti-beziers.fr/wp-content/uploads/2018/09/02-les-machines-d-usinage.pdf>. [Consulté le 17 juin 2025].
- [31] P. Smid, *CNC programming handbook: a comprehensive guide to practical CNC programming*, 3rd ed. New York, N.Y: Industrial Press, 2008.
- [32] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, London, U.K.: Springer, 2009.
- [33] R. du Preez, *5-Axis Machine Tools: Kinematics and Vismach Implementation in LinuxCNC*. SA-CNC-CLUB, Apr. 7, 2016.
- [34] M. C. Djamaa, "Cours MOCN," Univ. 8 Mai 1945 - Guelma, Département de Génie Mécanique, Algérie, 2020.
- [35] *NC-Programming Manual for Turning Centers with Fanuc 30 Series Controls*, FANUC Corporation, Publication No. B-64124EN, pp. 2–45.

- [36] NUM 1020/1040/1060T Programming Manual, Vol. 1, NUM SA, Edition 5, Document No. 0101938820, 1997.
- [37] Siemens AG, SINUMERIK 840D sl / 828D – Manuel de programmation : Notions de base, version 06/2009, réf. 6FC5398-1BP20-0DA
- [38] HEIDENHAIN GmbH, Manuel d'utilisation – Dialogue conversationnel Texte clair HEIDENHAIN TNC 320, Réf. 340 551-02, Éd. 2/2007, pp. 28, 75–80, 85, 330.
- [39] "LANGAGES ET PROGRAMMATION CNC," Ange Softs, [En ligne]. Disponible sur : https://www.ange-softs.com/programme_cnc_usinage.php. [Consulté le : 6 avril 2025].
- [40] ISO, Automation systems and integration — Numerical control of machines — Program format and definitions of address words — Part 1: Data format for positioning, line motion and contouring control systems, ISO 6983-1, 2e éd., 2009. Disponible : <https://cdn.standards.iteh.ai/samples/34608/9ade4b813619414f8ecf4722fe85fa4f/ISO-6983-1-2009.pdf>.
- [41] CNC Programming Tutorials Examples G & M Codes, Documentation Chapitre 2, CNC Programming Tutorials Examples G & M Codes G & M Programming, 2025. Disponible : [1]
- [42] C. de Boor, *A Practical Guide to Splines*, New York, NY, USA: Springer, 1978.
- [43] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed., Boston, MA, USA: Brooks/Cole, 2011.
- [44] C. Moler, *Numerical Computing with MATLAB*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), 2004.
- [45] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2003.
- [46] O. Kramer, *Genetic Algorithm Essentials*, vol. 679, Studies in Computational Intelligence. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-52156-5.
- [47] T. Vallée and M. Yildizoglu, "Présentation des algorithmes génétiques et de leurs applications en économie," *Revue d'économie politique*, vol. 114, pp. 711–745, 2004. [Online]. Available: <https://hal.science/hal-00125103>
- [48] K. Wang, "A Study of Cubic Spline Interpolation," **InSight : Rivier Academic Journal**, vol. 9, no. 2, 2013.
- [49] D. Larionescu. Numerical Methods (in Romanian). Ed. Tehnică, București, 1989.

- [50] R. Laguionie, « PROGRAMMATION AVANCEE DES MACHINES OUTILS:IMPLEMENTATION DE L'USINAGE TROCHOIDAL ET DU TREFLAGE DANS LA CHAINE NUMERIQUE STEP-NC ».
- [51] n8n.io, "Explore n8n Nodes in a Visual Reference Library," *n8n*, [En ligne]. Disponible sur : <https://n8n.io/workflows/3891-explore-n8n-nodes-in-a-visual-reference-library/>
- [52] n8n Docs, "n8n public REST API Documentation and Guides," *n8n Documentation*, [En ligne]. Disponible sur : <https://docs.n8n.io/api/>
- [53] GeeksforGeeks, "What is a Large Language Model (LLM)," *GeeksforGeeks*, [En ligne]. Disponible sur : <https://www.geeksforgeeks.org/what-is-a-large-language-model-llm/>
- [54] Y. Zhang et al., "A Survey on Large Language Models for Recommendation," *arXiv preprint*, May 2023. [En ligne]. Disponible sur : <https://arxiv.org/abs/2305.19860>
- [55] Model Context Protocol, "Introduction – Model Context Protocol," *modelcontextprotocol.io*, [En ligne]. Disponible sur : <https://modelcontextprotocol.io/introduction>
- [56] M. Richards and N. Ford, *Fundamentals of Software Architecture: An Engineering Approach*, O'Reilly Media, 2020.
- [57] Notion, "Notion API Documentation," 2024. [En ligne]. Disponible: <https://developers.notion.com/>
- [58] Google DeepMind, "Introducing Gemini: Google's most capable AI model yet," Google AI Blog, 2023. [En ligne]. Disponible: <https://blog.google/technology/ai/google-gemini-ai/>
- [59] OpenAI et al., "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.
- [60] Mistral AI, "Mistral Cloud Chat Model Documentation," 2024. [En ligne]. Disponible: <https://docs.mistral.ai/>