EDSP
École Doctorale
Sciences Pratiques

ENS
CACHAN

ENSC-2014-n°540

THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

présentée par

Virgile LACHARNAY

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Domaine :
Mécanique, Génie Mécanique, Génie Civil

Modélisation mécanique intégrant des champs répulsifs pour la génération de trajectoires 5 axes hors collision

Thèse soutenue à Cachan le 21/11/2014, devant le jury composé de :

Philippe LORONG	Professeur des Universités, Arts et Métiers ParisTech Paris	Président
Emmanuel DUC	Professeur des Universités, Institut Français de Mécanique Avancée	Rapporteur
Walter RUBIO	Professeur des Universités, Université Paul Sabatier-Toulouse 3	Rapporteur
Christian ARBER	Directeur Général, Missler Software	Examinateur
Christophe TOURNIER	Professeur des Universités, ENS de Cachan	Directeur de thèse
Sylvain LAVERNHE	Maître de Conférence, ENS de Cachan	Co-Encadrant

LURPA
Laboratoire Universitaire de Recherche en Production Automatisée
ENS de Cachan / Université de Paris Sud
61, avenue du Président Wilson – 94235 Cachan Cedex

À ma femme et ma fille qui m'ont supporté.....

Remerciements

Ces remerciements vont, dans un premier temps à mon jury de thèse et en particulier à M. **Walter RUBIO**, professeur à l'université Paul Sabatier de Toulouse et à M. **Emmanuel DUC**, professeur à l' Institut Français de Mécanique Avancée de Clermont-Ferrand pour avoir accepté d'être les rapporteurs de ce travail. Le jugement apporté à la qualité scientifique du document m'a permis d'améliorer ce dernier dans un souci de compréhensibilité. Je tiens, de même, à remercier M. **Philippe LORONG**, professeur aux Arts et Métiers Paritech de Paris, pour l'apport d'un point de vue très intéressant ainsi que l'acceptation de la présidence du jury. Je remercie à présent M. **Christian ARBER**, directeur général de Missler Software, pour avoir fait confiance, une fois de plus, au LURPA, pour un développement d'idée par forcément usuel dans le domaine industriel.

Ces travaux ont été réalisés au sein du LURPA, je tiens donc à remercier M. **Luc MATHIEU**, professeur à l'IUT de Cachan, pour son accueil au laboratoire.

Ces travaux ont été menés sous la direction de M. **Christophe TOURNIER**, professeur à l'École Normale Supérieure de Cachan, dont l'encadrement a été plus qu'essentiel... Il m'a guidé (de près ou de loin) au cours d'un stage de Master 1, d'un dossier d'agrégation, d'un stage de Master 2 et de trois ans de thèse. De plus l'encadrement apporté en tant que directeur de thèse n'est, pour le cas, pas comparable à l'existant. En effet, le nombre de réunion qui a permis une avancée de mes travaux est à relever.

Cette thèse a été co-encadrée par M. **Sylvain LAVERNHE**, maître de conférence à l'École Normale Supérieure de Cachan. Cet encadrement a été essentiel pour me permettre d'avancer avec un temps alloué à mes travaux très important. Il a su prendre le temps de m'expliquer plusieurs points alors que mon avis est parfois compliqué à modifier...

Cette thèse universitaire intéressait dans un premier temps l'entreprise Missler Software. Cette collaboration, mise en place avec M. **Pierre-Yves PÉCHARD** a été très intéressante afin de comprendre le réel intérêt du développement de mes travaux.

Merci également à Mme**Claire LARTIGUE**, qui a pris le temps de relire mes articles afin d'apporter une vision permettant de relever de nouveaux points importants.

Merci à l'ensemble des personnes du Département de Génie Mécanique qui m'ont aidé à réaliser mes missions d'enseignement. Je ne pourrai citer tout le monde, mais M. **Etienne LEFUR**, M. **Jean Marie VIRELY**, M. **Bruce ANGLADE** m'ont permis de m'échauffer pour le métier d'enseignant. Merci également aux deux **Christine** pour leur réactivité. De plus, merci à Mme **Christine ROSE** pour avoir supporté mes multiples retards de remplissage de papiers administratifs.

Ces remerciements seraient plus qu'incomplets sans un éloge à Mme **Danielle MARTIN** sans qui je n'aurai pas pu commencer cette thèse...

Je souhaite également remercier tous les membres du LURPA pour l'enrichissement professionnel et personnel apporté par ces derniers. De plus, je remercie particulièrement quelques collègues ; **Chavanne Champa** qui m'ont supporté dans notre bureau ; **Danancher** et **Camboulives** avec qui nous avons vécu l'ENS ensemble du début à la fin ; **Xavier** pour la motivation qu'il m'a donnée via un faux mépris... J'espère que l'aubergeade restera dans les souvenirs de chacun.

Je ne vais pas conclure ces remerciements sans penser à toute ma famille qui m'a soutenu et encouragé durant ces trois ans de travaux. En effet, cette thèse n'ayant pas commencé dans les meilleures conditions, ma famille et belle-famille a été à nos côtés pour nous maintenir la tête hors de l'eau. De plus ces derniers ont toujours montré la fierté qu'il pouvait avoir concernant mes travaux. Il m'est donc indispensable de remercier mes parents, **Laure** et **Philippe**, ma soeur **Myriam** et mon frère **Florian** pour leur présence à la soutenance et leur apport à rédaction. Comme je l'ai déjà dit, ma belle-famille a également été là pour me supporter... Merci à eux, **Sylvie**, **Pacal**, **Stéphanie**, **Julien** et **Mathieu** pour, en plus, être venu voir ma soutenance avec des remerciements sincères.

C'est avec une bonne dose de tendresse et d'amour que je souhaite également remercier ma femme **Caroline** pour son aide (relecture, pré soutenance) et surtout pour m'avoir supporté durant les 6 derniers mois de ma thèse... Je souhaite finir ma phase de remerciement en pensant à ma fille **Rose** qui est née quatre jours avant le début de ma thèse. Elle et ma femme ont été indispensables à ce début de thèse quelque peu particulier. C'est pour cela qu'un **MERCI** ne serait pas suffisant pour égaler tout ce qu'elles m'ont apporté.

Table des matières

Remerciements

Table des matières **i**

Table des figures **v**

Liste des tableaux **ix**

Introduction **1**

1 Problématique et état de l'art **5**

1 Usinage 5 axes 7

1.1 Contexte 7

1.2 Posage d'un outil pour un usinage en bout 11

1.3 Présentation des différents types de collisions en usinage 5 axes 12

2 Gestion classique des interférences globales en usinage 5 axes 15

2.1 Modélisation de l'outil 16

2.2 Modélisation de l'environnement, test de collisions et correction d'orientation 18

2.2.1 Définition et utilisation du C-Space 18

2.2.2 Méthode des enveloppes convexes 20

2.2.3 Méthode des sphères de Gauss 24

2.2.4 Application d'un décalage par rapport à la surface obstacle . . . 29

2.3 Optimisation de la trajectoire obtenue par interpolation 32

2.4 Conclusions 36

3 Gestion des interférences globales avec différentes approches potentielles 37

3.1 Effort dérivant d'un potentiel, champ de vecteurs 37

3.2 Utilisation de champs et de potentiels dans le domaine de la robotique mobile 38

3.3 Application de l'approche potentielle à l'usinage 43

4 Synthèse et problématique 46

4.1 Synthèse de l'état de l'art 46

4.2 Problématique 48

2 Développement du modèle **51**

1 Introduction . . . 53
2 Description du modèle dynamique basé sur l'analogie mécanique . . . 56
2.1 Modèle général 3D du mouvement d'un solide . . . 56
2.2 Méthode de résolution . . . 60
2.2.1 Présentation élémentaire de RK4 . . . 61
2.2.2 Application de la méthode RK4 à la génération de trajectoires . . . 62
2.3 Modèle à 2 ddl autour des vecteurs **f** et **t** dans le repère local . . . 62
3 Modélisation des actions mécaniques . . . 63
3.1 Modélisation de l'action répulsive . . . 64
3.2 Modélisation de l'action de rappel . . . 68
3.2.1 Expression du moment de rappel . . . 68
3.2.2 Condition de fonctionnement apériodique . . . 70
3.3 Attribution de valeurs fixes aux paramètres étudiés . . . 73
4 Effet des paramètres dynamiques sur la trajectoire obtenue . . . 73
4.1 Étude géométrique de la modification angulaire de l'axe outil . . . 75
4.2 Étude de l'impact de l'orientation de l'axe outil modifiée sur le comportement cinématique et dynamique du système . . . 76
5 Conclusions . . . 78

3 Modélisation d'une configuration d'usinage réaliste **81**

1 Introduction . . . 83
2 Résolution de l'équation du mouvement en régime permanent . . . 84
2.1 Étude des paramètres de simulation dynamique en régime permanent . . . 84
2.2 Résolution mécanique . . . 87
2.3 Analyse de la réponse du système . . . 90
3 Modélisation réaliste de l'environnement . . . 92
3.1 Rappel langage APT . . . 92
3.2 Maillage structuré . . . 93
3.3 Proposition de modifications du modèle d'effort répulsif . . . 96
3.4 Simulation sur pièce industrielle de type *Impeller* . . . 100
4 Modélisation réaliste de l'outil . . . 102
4.1 Maillage de l'outil cylindrique . . . 104
4.2 Modélisation de la peau pour l'ensemble outil/porte-outil . . . 107
5 Étude du temps de calcul de la simulation sur pièce industrielle . . . 110
6 Conclusions et perspectives . . . 112

4 Pré calcul de l'action répulsive **115**
1 Introduction . . . 117
2 Voxelisation et influence sur la trajectoire obtenue . . . 121
2.1 Paramétrage de la voxelisation . . . 121
2.2 Influence de la méthode d'interpolation retenue . . . 124
2.3 Influence de la taille de voxelisation sur la génération de trajectoires . . . 125
2.3.1 Droite alignée dans la veine . . . 126
2.3.2 Droite perpendiculaire aux surfaces obstacles . . . 127
2.4 Augmentation de l'admissibilité de la taille des voxels . . . 129
3 Influence de la voxelisation la simulation d'usinage de l'*impeller* . . . 130
3.1 Analyse de l'écart géométrique obtenue avec la voxelisation . . . 130
3.2 Analyse de la complexité temporelle obtenue avec la voxelisation . . . 132
3.2.1 Complexité réduite du champ d'effort voxelisé . . . 132
3.2.2 Gain de performance . . . 133
4 Conclusions et perspectives . . . 134

Conclusions générales et perspectives **137**

Bibliographie **143**

Annexes **151**
Annexe 1 : Méthode d'interpolation RK4 . . . 153
Annexe 2 : Surfaces de Bézier . . . 161
Annexe 3 : Interpolation linéaire permettant l'obtention des temps de calcul . . . 163
Annexe 4 : Résultats des simulations avec droite représentant les trajectoires avec une interpolation cubique . . . 165

Resume **167**

Table des figures

1 Diagramme d'activité du processus d'élaboration 2

1.1 Exemple d'*impeller* 7
1.2 Usinage en bout et sur le flanc [Takeuchi et al., 1995] 8
1.3 Diminution du nombre de posage utile à l'aide de l'usinage 5 axes 8
1.4 Zone de recouvrement en fraisage 5 axes positionné [Makki et al., 2006] 9
1.5 Modélisation du gain en terme de longueur utile 9
1.6 Influence du positionnement du repère pièce par rapport au repère machine [Castagnetti, 2005] 10
1.7 Géométrie outil 5 axes 11
1.8 Paramétrage du positionnement d'outils en 5 axes [Tournier, 2001] 12
1.9 Diagramme d'exigence partiel de l'usinage 5 axes 13
1.10 Diagramme de contexte des types d'interférences 13
1.11 Arborescence de l'état de l'art concernant la gestion des collisions globales 15
1.12 Diagramme d'activité de la simulation de fabrication 16
1.13 Formalisme APT 16
1.14 Décomposition implicite de la géométrie outil [Ho et al., 2001] 17
1.15 Décomposition de l'outil en 4 zones d'étude [Monies et al., 2004] 17
1.16 Exemple de maillage d'un outil [Schmitz et al., 2007] 18
1.17 Schéma illustrant le problème [Lee and Chang, 1995] 20
1.18 Prise en compte de la géométrie outil [Lee and Chang, 1995] 21
1.19 Différentes méthodes de génération d'enveloppes convexes [Ding et al., 2004] . . . 22
1.20 Utilisation de *Bounding Sphere* englobant le solide considéré [Chang et al., 2009] . 22
1.21 Mise en place de *AABB* [Cohen et al., 1995] 23
1.22 Utilisation de la méthode *OBB* en 3D (a) et en 2D (b) [Ding et al., 2004] 23
1.23 Illustration de la méthode OBBTree [Gottschalk et al., 1996] 24
1.24 Exemple de sphères de Gauss [Woo, 1994] 25
1.25 Positionnement outil 5 axes [Wang and Tang, 2007] 25
1.26 Différence entre visibilité globale et locale [Balasubramaniam et al., 2000] 26
1.27 Cônes de visibilité [Balasubramaniam et al., 2000] 26
1.28 Création de la carte de visibilité [Wang and Tang, 2007] 27
1.29 Modification de la carte de faisabilité [Wang and Tang, 2007] 27
1.30 a-visibilité locale b-visibilité globale c-visibilité réelle [Makki, 2009] 28
1.31 Évitement d'obstacle [Wang and Tang, 2007] 28
1.32 Restriction de la carte de visibilité globale [Hu et al., 2013] 29

1.33 Présentation de la méthode de réorientation [Morishige et al., 1997] 30
1.34 Résultat de la simulation [Morishige et al., 1997] 30
1.35 Méthode de détermination de l'orientation admissible de l'axe outil [Morikawa et al., 2006] 31
1.36 Zone libre d'interférences dues aux surfaces obstacles [Morikawa et al., 2006] . . . 31
1.37 Paramétrage de la problématique [Ho et al., 2003] 32
1.38 Résultat de la simulation [Ho et al., 2003] 33
1.39 Sphères de Gauss obtenues le long de la trajectoire [Zhiwei et al., 2012] 33
1.40 Évolution de l'angle due à l'interpolation linéaire entre Q_1 et Q_2 [Zhiwei et al., 2012] 34
1.41 Choix de la longueur outil [Bi et al., 2010] 34
1.42 DOA et lissage des axes rotatifs [Castagnetti et al., 2008] 35
1.43 Différence entre le positionnement outil initial et l'optimisé [Castagnetti et al., 2008] 35
1.44 Champ potentiel et champ vectoriel correspondant 38
1.45 Objectif des travaux étudiés [Khatib, 1980] 38
1.46 Mise en avant du problème traité [Ge and Cui, 2000] 39
1.47 Effort répulsif obtenu après dérivation de la nouvelle fonction définissant le potentiel [Ge and Cui, 2000] 41
1.48 Résultats du problème traité [Ge and Cui, 2000] 41
1.49 42
1.50 Positionnement du problème [Cho et al., 1997] 43
1.51 Calcul de l'effort répulsif utilisé [Cho et al., 1997] 44
1.52 Solution obtenue dans le cas d'un *impeller* [Cho et al., 1997] 44
1.53 Représentation des deux situations [Asakawa and Kanjo, 2013] 45
1.54 Trajectoire prenant en compte l'obstacle [Asakawa and Kanjo, 2013] 45
1.55 Utilisation d'un système haptique [Chen et al., 2005b] 46
1.56 Représentation des deux situations possibles [Chen and Tang, 2006] 46

2.1 Mise en avant de la problématique 53
2.2 Diagramme d'activité de la simulation de fabrication envisagée 54
2.3 Diagramme d'activité représentant **Geo5XField** 55
2.4 **Mvt**$(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}})$ 57
2.5 **Mvt**$(\text{outil}_{\text{strat}}/R_{fnt})$ 58
2.6 **Mvt**(R_{fnt}/R_p) 59
2.7 Représentation du principe de résolution RK4 61
2.8 Paramétrisation et angles d'orientation 63
2.9 Géométrie du cas test 64
2.10 (a) Potentiel répulsif (b) Effort répulsif 65
2.11 Évolution de l'effort répulsif en fonction de $r/r_{0_{ini}}$ pour différents r_0 66
2.12 Évolution de l'effort répulsif en fonction de $r/r_{0_{ini}}$ pour différents η 67
2.13 Simulation avec moment répulsif 67
2.14 Modélisation du ressort amortisseur angulaire selon θ_f 69
2.15 Simulation avec effort répulsif et de rappel 69
2.16 Représentation des différents régimes 71

2.17 Simulation avec effort répulsif et de rappel 72
2.18 Influence de la raideur, de l'inertie et du voisinage sur l'orientation outil 74
2.19 Différence entre $\theta_{\text{corrigé}}$ et $\theta_{\text{stratégie}}$ 75
2.20 Différence entre $\dot{\theta}_{\text{corrigé}}$ et $\dot{\theta}_{\text{stratégie}}$ 76
2.21 Différence entre $\ddot{\theta}_{\text{corrigé}}$ et $\ddot{\theta}_{\text{stratégie}}$ 77
2.22 Différence entre $\dddot{\theta}_{\text{corrigé}}$ et $\dddot{\theta}_{\text{stratégie}}$ 78

3.1 Usinage d'un *impeller* 83
3.2 Présentation du cas d'étude 85
3.3 Vision pratique : distance entre obstacles = $8mm$ 85
3.4 Évolution de θ_f, $\mathcal{M}_{rep_f}$ et $\mathcal{M}_0.(1 + cos(\omega.t))$ en fonction de s 86
3.5 Évolution du ratio d'amplitude adimensionnée en fonction du rapport pulsation propre de l'excitation sur pulsation propre du système 88
3.6 Diagramme de Bode du système amorti 89
3.7 Vision pratique : distance entre obstacles = $1mm$ 90
3.8 Évolution de θ_f en fonction de s 91
3.9 Environnement et posage outil par la méthode APT [APT, 1983] 92
3.10 Modélisation de la surface considérée comme obstacle lors de cette étude (*Check-Surface*) 93
3.11 Mise en place de la simulation 94
3.12 Vision pratique : distance entre points obstacles = $3mm$ 94
3.13 Vision pratique : distance entre points obstacles = $1mm$ 95
3.14 Exemple de passage d'un maillage structuré à un non structuré 96
3.15 Représentation des trois méthodes de calcul d'efforts répulsifs 98
3.16 Plan d'expérience 99
3.17 Décomposition de l'*impeller* étudié 100
3.18 Trajectoire calculée sans prise en compte des obstacles 101
3.19 Tracé des obstacles et vision des collisions avec la trajectoire programmée 101
3.20 Trajectoire calculée avec prise en compte des obstacles 102
3.21 Illustration de la problématique concernant la représentation de l'outil 103
3.22 Outil hémisphérique 103
3.23 Maillage de la surface représentant l'outil cylindrique utilisé 104
3.24 Problématique sur la définition du moment répulsif complet 105
3.25 Simulation avec la programmation de base 106
3.26 Comparaison des modifications de θ_f obtenues 107
3.27 Modélisation de l'ensemble outil/porte-outil sans la partie de coupe hémisphérique soumise aux collisions locales 108
3.28 Simulation avec outil complet 108
3.29 Représentation de la différence de réorientation 109
3.30 Évolution des angles corrigés en fonction de s 109
3.31 Usinage complet de la surface guide ($v = 0,2 : 0,1 : 0,8$) 111

4.1 Pièce test et détail des surfaces pour la veine étudiée 117

4.2 Création des voxels [Kitney et al., 1989] . . . 118
4.3 *Bounding Box*[Tarbutton et al., 2013] . . . 118
4.4 Méthode de voxelisation de l'environnement étudié [Asakawa and Kanjo, 2013] . . 119
4.5 Calcul approximé de F_{rep} sur le point étudié [Asakawa and Kanjo, 2013] . . . 119
4.6 Principe de l'octree avec codage de la position d'une sous-cellule [Girardeau-Montaut, 2006] . . . 120
4.7 Construction d'octree autour d'une surface maillée [Nvidia, 2014] . . . 120
4.8 Partitionnement de l'espace total . . . 122
4.9 Illustration d'une conservation de voxels utiles ($10mm$) . . . 122
4.10 Voxel limite . . . 123
4.11 Interpolation linéaire $3D$ [Tarbutton et al., 2013] . . . 124
4.12 Comparaison des méthodes d'interpolation . . . 125
4.13 Illustration 2D . . . 126
4.14 Droite de l'espace représentant une trajectoire simulée . . . 126
4.15 Erreur relative de l'effort répulsif obtenu . . . 127
4.16 Droite de l'espace représentant une trajectoire . . . 128
4.17 Erreur relative de l'effort répulsif obtenu . . . 128
4.18 Positionnement de la voxelisation . . . 129
4.19 Comparaison méthode continue et voxelisation . . . 131
4.20 Influence de la voxelisation sur l'orientation outil . . . 131
4.21 Programmation d'usinage 5 axes d'une pièce dite de type veine . . . 141
4.22 Résultats obtenus . . . 156
4.23 Modélisation du problème . . . 157
4.24 Résultats obtenus . . . 159
4.25 Surface de Bézier engendrée par la génératrice $C(u)$ de degrés 3 et les directrices $D_i(v)$ de degrés 3 . . . 161
4.26 Modélisation de la surface considérée comme obstacle lors de cette étude (*Check-Surface*) . . . 162
4.27 Temps de calcul pour l'usinage complet de l'*impeller* . . . 163
4.28 Erreur relative de l'effort répulsif obtenu . . . 165
4.29 Erreur relative de l'effort répulsif obtenu . . . 165

Liste des tableaux

2.1 Paramètres fixés 73
2.2 Niveau des paramètres pour la simulation 74

3.1 Paramètres 91
3.2 Étude du temps de calcul selon la modélisation de l'outil 112

4.1 Comparaison des temps de génération des trajectoires entre calcul exact et calcul par voxels de l'effort répulsif 134
4.2 Paramètres 139
4.3 Partie des solutions 155
4.4 Partie des solutions 156
4.5 Coordonnées des points de contrôles utilisés 162

Introduction

Contexte

Le processus de réalisation des pièces de formes complexes par usinage est un processus essentiel dans les domaines de l'aéronautique, des moules et matrices. À partir d'un modèle CAO issu de travaux de conception fondés sur l'analyse fonctionnelle préalablement établie (donnant les tolérancements et spécifications géométriques adéquats), un modèle FAO est alors décrit avec les trajectoires et orientations outils générées à l'aide d'un postprocesseur. Le passage par une commande numérique permet alors d'adapter la trajectoire calculée à la machine utilisée et de passer à une fabrication du composant concerné. Par la suite, le produit fini est alors représenté comme étant un assemblage de composants fabriqués. La représentation donnée en figure 1 illustre le processus d'élaboration décrit par un diagramme d'activité SysML permettant, au cours de cette étude, de positionner les modifications effectuées.

Il est donc important de rappeler que le processus d'élaboration sert à décrire l'enchaînement d'activités permettant, à partir d'un cahier des charges, d'obtenir un produit assemblé fournissant un besoin réalisé le plus proche possible du besoin attendu par le client (diminution maximale de l'Indice Prévisionnel des Risques). De nombreux travaux traitent des différentes activités présentées sur la figure 1. Les travaux présentés au cours de cette thèse se positionnent au niveau de l'activité notée **Simulation de fabrication**. Cette dernière comporte comme données d'entrée concernant la pièce étudiée le modèle CAO ainsi que le dessin de définition et comme donnée de sortie un modèle FAO. Bien évidemment des données extérieures sont également présentes (données machine et données outil), mais non représentées.

Dans ce contexte, l'objectif de la thèse est de permettre à un utilisateur de logiciel FAO, une génération de trajectoires 5 axes hors collision par l'utilisation de la méthode de champs répulsifs rarement employée dans le domaine de la fabrication. Cela permettrait d'assurer une trajectoire corrigée utilisable dans le domaine actuel de la fabrication.

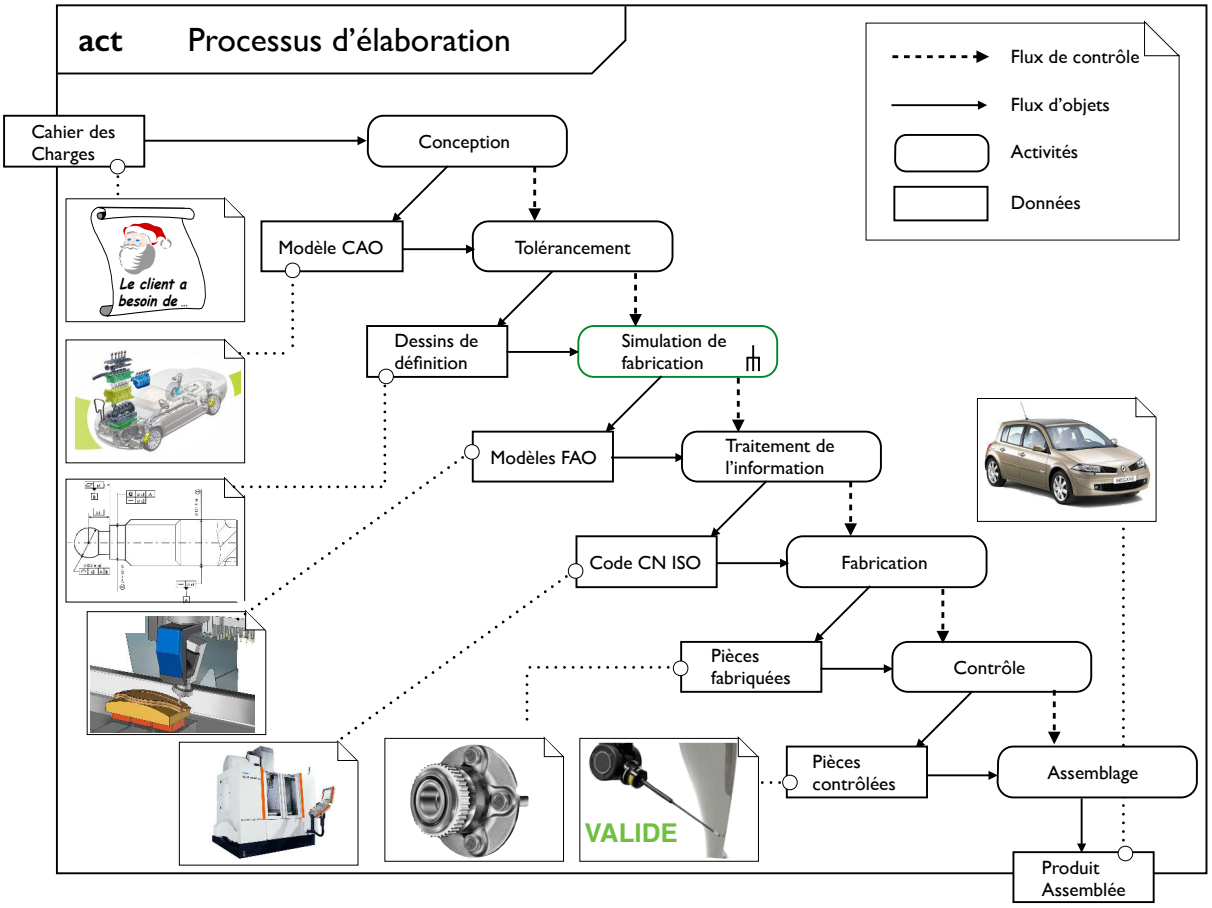


Figure 1 – Diagramme d'activité du processus d'élaboration

Articulation du manuscrit

Le manuscrit est organisé en 4 chapitres. Le premier chapitre permet de réaliser une analyse approfondie de l'existant en ce qui concerne l'évitement de collision. Cette dernière commence, dans un premier temps par la description de l'usinage 5 axes et plus particulièrement en bout d'outil. Trois types d'interférences sont alors décrits afin de comprendre le positionnement des travaux présentés au cours de cette thèse. L'état de l'art concernant la gestion classique des collisions, dites globales, dans le domaine de l'usinage 5 axes est réalisé avec un partitionnement en trois étapes :

— modélisation de l'outil,
— modélisation de l'environnement, test de collisions et correction d'orientation,
— optimisation de la trajectoire corrigée.

Une analyse de l'existant concernant la gestion de ces mêmes interférences avec une approche dite «d'utilisation de fonctions potentielles» est alors réalisée. Cette technique, principalement utilisée dans le domaine de la robotique mobile, permet d'envisager une correction intuitive de l'orientation de l'outil avec une prise en compte anticipée de l'approche de l'obstacle concerné. Enfin ce chapitre met en avant la problématique retenue avec les différents objectifs en découlant.

Le second chapitre permet de poser les bases mathématiques permettant de programmer, d'un point de vue théorique, l'évitement de collision entre un point outil et un point obstacle. Pour ce faire, l'aspect physique est alors mis en corrélation avec l'aspect usinage 5 axes afin de relier ces deux domaines de travail à l'aide de l'utilisation du Principe Fondamental de la Dynamique. Pour permettre un évitement de collision entre le point étudié et le point obstacle, il est alors nécessaire de mettre en place un effort répulsif, permettant l'évitement, et un effort de rappel, permettant le retour du point étudié vers la trajectoire initialement programmée. Ces derniers sont alors intégrés à l'équation différentielle précédemment obtenue. Enfin, l'influence de différents paramètres sur la réponse du système obtenue en régime transitoire est traitée.

Afin de compléter la partie théorique développée durant le chapitre 2, le troisième porte sur une étude de l'équation différentielle du second ordre avec second membre du mouvement en régime permanent. Cela est réalisé dans le but d'utiliser des obstacles beaucoup plus réalistes afin de s'approcher d'une simulation d'usinage 5 axes concrète. Pour ce faire, après avoir étudié l'analyse de la réponse en régime permanent, l'étude porte sur deux nouveaux points :

— modélisation réaliste de l'environnement,

— modélisation réaliste de l'outil.

Ces modélisations apportent quelques modifications sur l'expression théorique du mouvement étudié. Dans le dessein de conclure sur cette généralisation, une simulation sur une pièce industrielle est alors réalisée. Enfin, une nouvelle problématique est ensuite soulevée concernant les temps de calcul obtenus sur cette pièce de type *impeller*.

Le dernier chapitre de cette thèse propose alors une alternative permettant, en échange d'un comportement quelque peu différent, d'obtenir un temps de calcul beaucoup plus faible pour une trajectoire d'usinage complète. Pour atteindre cet objectif, la méthode utilisée est le partitionnement de l'espace. Cela permet d'ajouter un paramètre à l'environnement et supprime une étape de calcul au système précédemment présenté. Ces gains sont illustrés sur la même pièce complexe afin d'obtenir une comparaison cohérente.

Chapitre 1

Problématique et état de l'art

Ce premier chapitre permet de préciser le contexte de l'étude avec une attention portée au posage d'un outil en usinage 5 axes en bout. Puis, les travaux décrivent quelques méthodes existantes permettant un évitement de collision entre l'outil et l'environnement. La classification des interférences décomposée en deux espaces principaux, collisions locales ou collisions globales, est effectuée. Par la suite, une partie des différentes méthodes existantes d'évitement dans le domaine de l'usinage 5 axes sont présentées avec une séparation entre correction et optimisation d'orientation. Ensuite, une méthode de génération de champs et de potentiels répulsifs issue de travaux concernant principalement la robotique mobile sera présentée afin de comprendre comment sont repoussés les solides étudiés. Cette étude bibliographique permet de positionner clairement ces travaux en mettant en exergue l'intérêt du calcul mis en place dans l'obtention d'une trajectoire d'outil finale suffisamment lisse.

Sommaire

1 Usinage 5 axes 7
1.1 Contexte 7
1.2 Posage d'un outil pour un usinage en bout 11
1.3 Présentation des différents types de collisions en usinage 5 axes 12
2 Gestion classique des interférences globales en usinage 5 axes 15
2.1 Modélisation de l'outil 16
2.2 Modélisation de l'environnement, test de collisions et correction d'orientation . . . 18
2.3 Optimisation de la trajectoire obtenue par interpolation 32
2.4 Conclusions 36
3 Gestion des interférences globales avec différentes approches potentielles . . 37
3.1 Effort dérivant d'un potentiel, champ de vecteurs 37
3.2 Utilisation de champs et de potentiels dans le domaine de la robotique mobile . . . 38
3.3 Application de l'approche potentielle à l'usinage 43
4 Synthèse et problématique 46
4.1 Synthèse de l'état de l'art 46
4.2 Problématique 48

1 Usinage 5 axes

1.1 Contexte

Durant cette thèse, les travaux présentés portent principalement sur l'écart entre le modèle FAO et la fabrication réelle de la pièce localisable sur le diagramme d'activité général présentant le processus d'élaboration en figure 1.

Alors que l'usinage à 5 axes est maintenant répandu dans les grands groupes industriels, des machines moins chères et de plus petites tailles laissent entrevoir aux PME des possibilités d'améliorer leur compétitivité par le recours à ce procédé de fabrication. Le fraisage 5 axes est un procédé qui trouve son intérêt dans trois applications distinctes. Tout d'abord, la réalisation de formes dites difficiles (voir impossibles dans le domaine du 3 axes) telles que les pièces d'aéro ou d'hydrodynamique comme les aubes de compresseurs ou de turbines illustrées en figure 1.1.

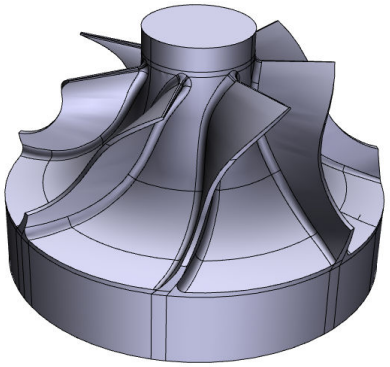


Figure 1.1 – Exemple d'*impeller*

Dans ce domaine, l'usinage 5 axes permet d'accroître les possibilités d'usinage des pièces dites complexes à l'aide d'un nombre de positionnement et d'orientation disponibles beaucoup plus importantes que dans le domaine du 3 axes. Cela permet d'augmenter l'accessibilité.

Afin de réaliser l'usinage de ce type de surface, il est possible de programmer pour les phases de semi-finition et de finition, soit un usinage en bout d'outil, soit un usinage par le flanc. De nombreux travaux mettent en avant les différences obtenues entre ces deux méthodes [Monies, 2001] [Rubio, 1993] [Tönshoff and Rackow, 2000]. Comme illustrée en figure 1.2, la différence majeure entre ces deux méthodes d'usinage est le contact entre l'outil et la surface concernée. En effet, pour l'usinage en bout, le contact entre la pièce et l'outil est ponctuel ce qui génère, après la phase de balayage programmée, une hauteur de crête restante (dépendant de l'orientation outil et de la distance entre passes).

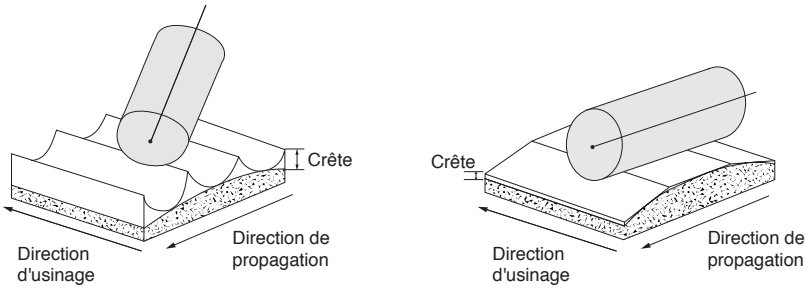


Figure 1.2 – Usinage en bout et sur le flanc [Takeuchi et al., 1995]

Cette figure illustre, pour des surfaces classiques (développable et réglée), que l'usinage sur le flanc reste fortement avantageux par rapport à la méthode d'usinage en bout avec une assurance de contact linéique permettant un taux d'enlèvement de matière plus important (donc un temps d'usinage plus faible), avec, pour finir, un meilleur état de surface. Cependant, le passage à l'usinage 5 axes permet, en partie, la programmation d'usinage sur des surfaces complexes qui ne sont pas forcément développables et de ce fait, non usinables parfaitement en flanc d'outil. En effet, de nombreux travaux ont démontré l'existence de cette problématique concernant l'utilisation d'outil en flanc [Rehsteiner and Renker, 1993]. Il parait alors utile de travailler sur le positionnement de l'outil par rapport à la surface à usiner (théoriquement) [Chaves-Jacob, 2009] [Pechard, 2009] [Senatore et al., 2005]. L'usinage en bout, au contraire, permet d'envisager l'usinage de tous types de surface complexe avec une meilleure gestion des écarts géométriques.

Dans un second temps, il faut souligner que l'usinage 5 axes est également un moyen de minimiser le nombre de posages et donc les erreurs systématiques de remise en position de pièces prismatiques. En effet, l'exemple donné en figure 1.3 illustre ce gain de nombre de mises en position. Cependant, bien que le nombre de remises en position de la pièce usinée diminue, la machine 5 axes ajoute deux mobilités en terme de rotation afin de pouvoir travailler les sous-phases obtenues. Ces dernières impliquent un défaut supplémentaire en comparaison au système 3 axes classiquement utilisé [Prevost, 2011].

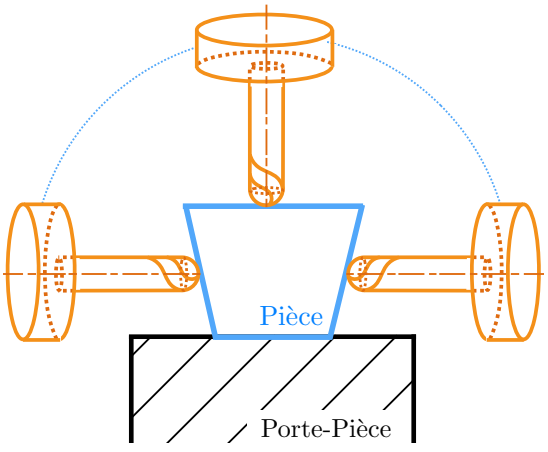


Figure 1.3 – Diminution du nombre de posage utile à l'aide de l'usinage 5 axes

Il peut aussi être important de prendre en compte les erreurs d'identification du modèle géométrique utilisé. En effet, des essais ont permis de mettre en avant que l'utilisation de deux configurations articulaires différentes peut, si l'axe de rotation utilisé est loin de l'axe de rotation théorique, générer deux usinages différents. Cette étude expérimentale est représentée en figure 1.4 [Makki et al., 2006].

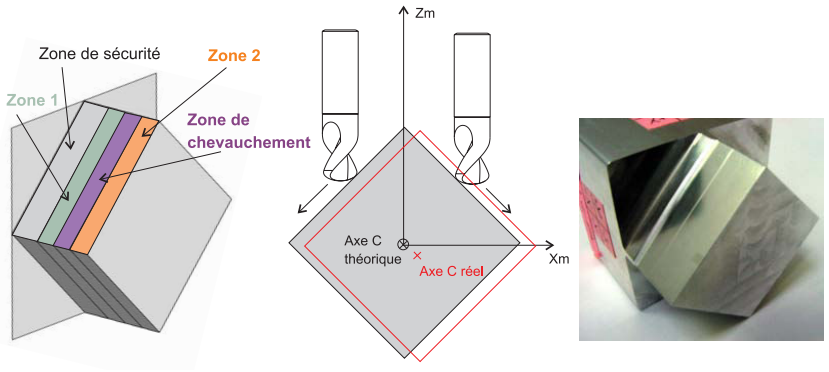


Figure 1.4 – Zone de recouvrement en fraisage 5 axes positionné [Makki et al., 2006]

Enfin, le fraisage 5 axes permet d'améliorer la productivité à qualité constante pour les pièces de types moules et matrices où le problème de nombre de posages ne se pose pas étant donné qu'il n'y a pas de contre-dépouille sur ce type de pièce, en vue de permettre le démoulage qui suit. Les méthodes 5 axes restent également avantageuses au niveau de l'usinage de ces pièces de formes complexes étant donné qu'un autre de ces points forts est la possibilité de réduire la dimension utile des outils. En effet, la réorientation réalisable, comme illustrée en figure 1.5, permet de diminuer la longueur de l'outil, tout en conservant le même point de contact. Il est donc possible d'obtenir la dimension optimale de l'outil dans le cadre de l'usinage de moule et de matrice [Cho et al., 2012]. À l'aide de cette réorientation, l'effet «canne à pêche» est atténué et cela permet de minimiser les erreurs dues à la flexion et aux vibrations du corps outil.

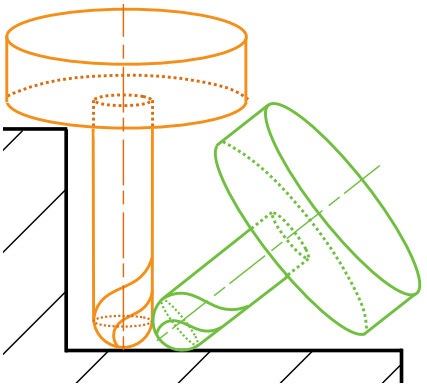
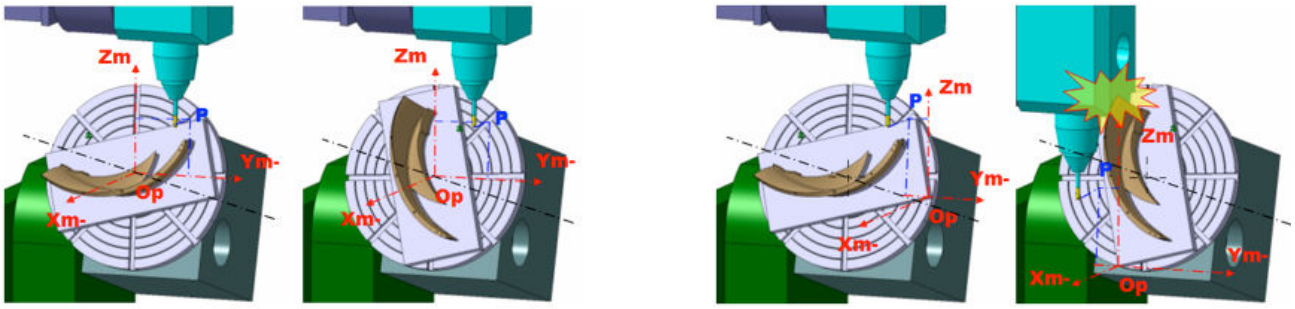


Figure 1.5 – Modélisation du gain en terme de longueur utile

La nouvelle représentation de l'environnement complet est à prendre en compte afin de repositionner les obstacles susceptibles de générer des collisions. Cela n'est pas forcément aisé pour

l'utilisateur, car plusieurs repères de travail peuvent être utilisés selon la machine et le logiciel utilisés.

Il s'avère donc important, pour une génération de trajectoires, de connaitre l'influence du positionnement des différents repères utilisés au cours de la simulation ainsi que les orientations initialement programmées. L'exemple donné en figure 1.6 illustre la collision obtenue pour un seul des deux cas alors que la trajectoire programmée est identique d'un point de vue machine [Castagnetti, 2005].



(a) Origine O_p au centre (b) Origine O_p décalée

Figure 1.6 – Influence du positionnement du repère pièce par rapport au repère machine [Castagnetti, 2005]

Il est alors possible de synthétiser les différents avantages de l'usinage 5 axes en cinq points principaux :

- usiner des surfaces présentant des contre-dépouilles,
- imposer et maintenir des conditions de coupes en modifiant l'orientation outil,
- usiner sur le flanc de l'outil permettant d'enlever le plus de matière à chaque passe,
- minimiser le nombre de posages sur la machine,
- réduire l'effet «canne à pêche».

Il faut également rappeler les principaux inconvénients qui génèrent un intérêt au travail de recherche dans le domaine de l'usinage 5 axes. En effet, comme cela a été précédemment rappelé, le passage au 5 axes apporte de la complexité de programmation, avec une nécessité de simulation pré usinage, l'utilisation du postprocesseur parfois complexe, la gestion des collisions et les vitesses obtenues réellement. De plus le volume utile (volume de travail disponible) est fortement diminué à cause de l'architecture imposée via l'ajout des deux possibilités de rotation supplémentaires [Castagnetti, 2008] qui vont également apporter une nouvelle problématique concernant l'impact de la cinématique de ces dernières sur l'usinage final de la pièce considérée. Le fait d'avoir des trajectoires plus lisses permet, si besoin est, de combler la différence de performance entre les axes de rotation et de translation sur certaines MOCN 5 axes.

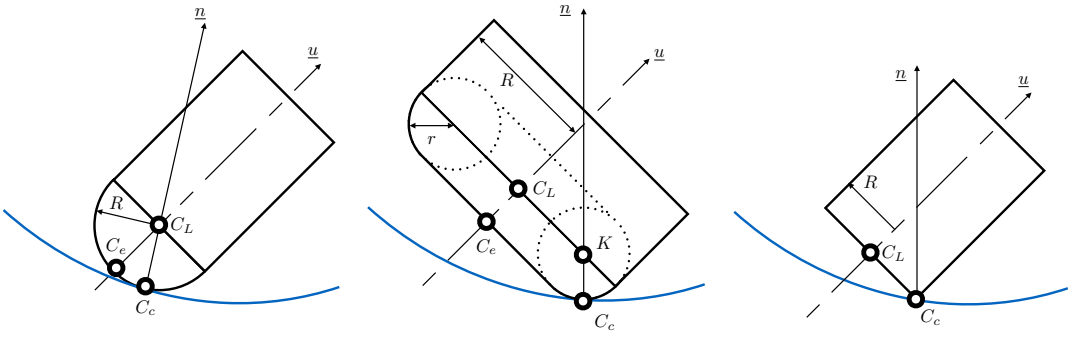
Par la suite, les notions utiles du positionnement outil et de la génération de trajectoires en usinage 5 axes en bout d'outil sont présentées. Il est alors mis en évidence qu'il existe plusieurs types d'interférences possibles qui sont toutes traitées à l'aide de diverses méthodes décrites ultérieurement.

1.2 Posage d'un outil pour un usinage en bout

Ce type d'usinage permet la réalisation de surfaces par la création de passes d'usinage ordonnées en fonction de la stratégie d'usinage retenue et la possibilité de modifier l'orientation de l'axe de l'outil. Ceci permet d'éviter, par exemple, la coupe au centre dans le cas de l'utilisation d'outils hémisphériques. L'usinage en bout est défini en partant du fait qu'il existe un seul point de contact entre la surface théorique et la modélisation de l'outil.

Les figures 1.7 illustrent le positionnement d'un outil lors d'une programmation en 5 axes pour les trois modèles d'outils classiquement utilisés [Duc, 1998]. Afin d'obtenir le positionnement complet d'un outil en 5 axes, il est nécessaire de connaitre l'orientation de l'outil par rapport au repère associé. Le positionnement global d'un outil en 5 axes peut alors être défini par la connaissance du couple (C_L,**u**) avec C_L point centre outil et **u** la direction de l'axe de l'outil [Duc, 1998] comme illustré en figure 1.8 ; ou par la connaissance du quintuplé (X_c,Y_c,Z_c,α,β) [Jun et al., 2003]. Les trois points représentant la géométrie de l'outil illustré en figure 1.7 sont notés comme suit :

— C_L *Cutter Location* : point centre outil,
— C_c *Cutter Contact* : point de contact,
— C_e *Cutter Extremity* : point extrême de l'axe de rotation.



(a) Outil hémisphérique (b) Outil torique (c) Outil cylindrique

Figure 1.7 – Géométrie outil 5 axes

Le paramétrage de positionnement outil retenu lors de ces travaux est le repère de Frenet défini par les trois vecteurs (**f**,**n**,**t**) [Tournier, 2001]. Les vecteurs utilisés sont définis comme suit :

— **f** est le vecteur tangent à la courbe suivie par le point piloté,
— **n** est le vecteur normal à la surface,
— **t** est le vecteur transverse à la trajectoire et tangent à la surface tel que : $\mathbf{t} = \mathbf{f} \wedge \mathbf{n}$.

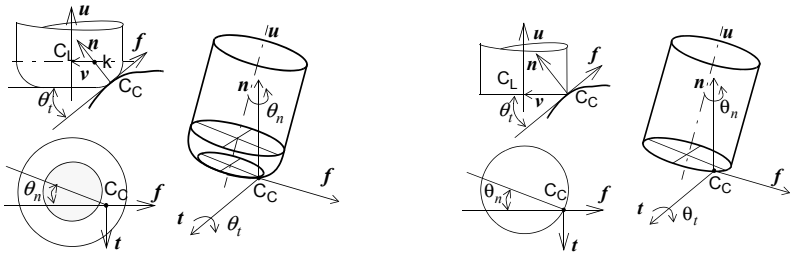
De cette manière, le positionnement outil est complètement paramétré par rapport à la surface usinée. Initialement, l'axe outil **u** est confondu avec **n** avant une application de deux rotations permettant d'obtenir l'orientation désirée. En effet, la longueur outil étant fixée, le choix de deux angles de travail sur les trois définis ci-dessous permettra de gérer l'orientation de l'axe outil.

— $(\theta_f, \mathbf{f})$: le roulis (*roll*),

— $(\theta_n, \mathbf{n})$: le lacet (*yaw*),

— $(\theta_t, \mathbf{t})$: le tangage (*pitch*).

Comme il n'y a pas de choix de traitement d'orientation des axes imposé, il est possible de choisir la zone de travail. Par exemple, certains travaux [Choi et al., 1993] [Lee, 1997] utilisent tout d'abord le tangage puis travaillent avec le lacet (figure 1.8), cette vision représente la configuration classiquement utilisée. D'autres méthodes utilisent une modification orientation différente où la première rotation est le tangage suivi du roulis [Redonnet et al., 2000].



(a) Outil torique (b) Outil cylindrique

Figure 1.8 – Paramétrage du positionnement d'outils en 5 axes [Tournier, 2001]

Par la suite, les différents types de collisions possibles sont présentés afin de comprendre les diverses méthodes d'évitement existantes.

1.3 Présentation des différents types de collisions en usinage 5 axes

Une fois que le positionnement outil est défini, la gestion des collisions en bout entre ce dernier et l'environnement défini dans le domaine de l'usinage 5 axes est un point de travail important et peut être relié à la fabrication via le diagramme d'exigence fourni en figure 1.9.

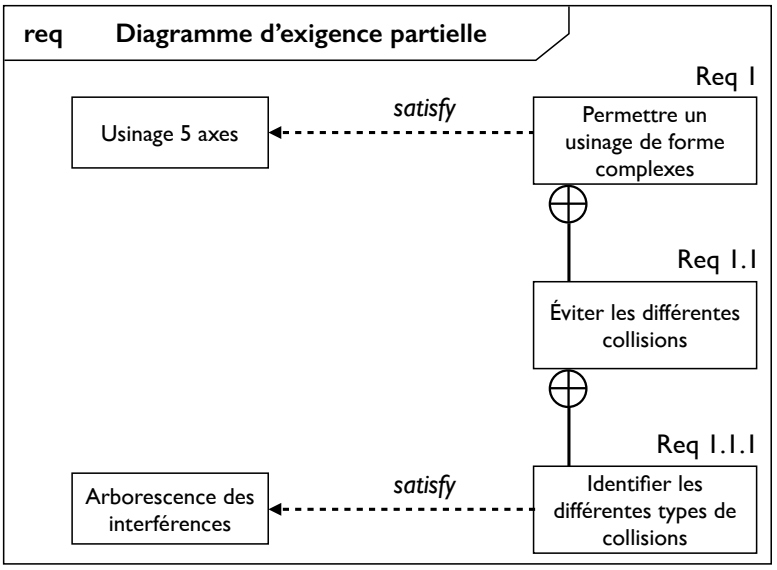


Figure 1.9 – Diagramme d'exigence partiel de l'usinage 5 axes

C'est pour cela que la description du positionnement outil est alors essentielle pour comprendre comment les éléments considérés comme obstacles peuvent être modélisés, détectés et évités. Les collisions dans le domaine de l'usinage sont classées en trois catégories. Elles sont, par exemple, définies par le diagramme de contexte 1.10 :

— interférences locales : décalage des courbures entre l'outil et la surface balayée (élimination d'un excès de matière),
— interférences de talonnage : collision entre l'arrière de la fraise utilisée en bout et la surface usinée (dégradation de cette dernière),
— interférences globales : collision entre l'ensemble outil/porte-outil/broche et l'environnement machine restant (dégradation du centre d'usinage).

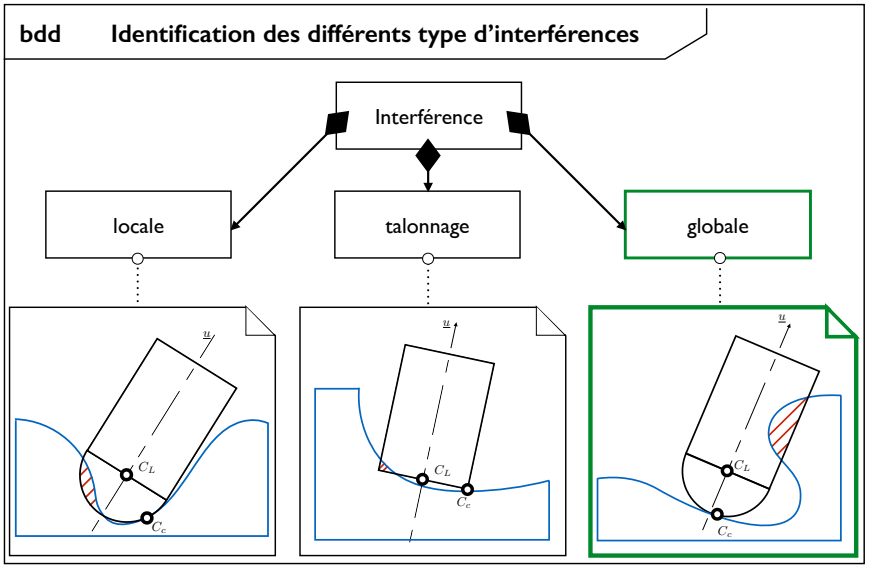


Figure 1.10 – Diagramme de contexte des types d'interférences

Ces trois types d'interférences ont un impact différent sur la pièce usinée. Il peut s'agir de génération de défauts géométriques parfois plus grands que les tolérances spécifiées (au niveau des collisions locales), mais également d'une dégradation de la machine au pire des cas (au niveau des collisions globales). De ce fait, ils sont traités de manières différentes.

Après avoir décrit le contexte de travail, la modélisation du positionnement outil en 5 axes et l'existence de différents types d'interférences entre outil et environnement, la suite des travaux porte sur la gestion des collisions globales avec différentes manières de traiter le problème. La méthode d'obtention d'une trajectoire fluide après correction lors d'un évitement d'obstacles existants dans le domaine de l'usinage 5 axes (avec des critères chiffrés) est également un point développé dans la suite de cet état de l'art. Il sera mis en avant la difficulté de calcul à ce niveau-là et notamment le nombre d'itérations nécessaires pour un évitement assuré. Le positionnement local est indépendant des collisions globales, c'est pour cela qu'il est envisageable de travailler sur l'évitement de collisions globales, sans avoir de contraintes concernant la partie coupante de l'outil. Dans le but de ne pas surcharger cette étude, la description des méthodes de gestion des collisions locales ne sera pas donnée dans ce rapport. Il est tout de même important de souligner que, de nombreuses méthodes ont été développées dans ce domaine, telles que l'utilisation de surface *Offset* pour outils toriques [Rubio et al., 1998], l'utilisation de cônes d'acceptabilité [Wang and Tang, 2008], le développement de méthode *Rolling ball* [Gray et al., 2003] [Lee et al., 2000], la méthode itérative de repositionnement [Lauwers et al., 2001] [Lin et al., 2014] ou encore [Chen et al., 2005a] [Chen et al., 2009] [Hosseinkhania et al., 2007] [Jun et al., 2002] [Lee and Lee, 2007] [Li and Jerard, 1994].

La partie suivante traite d'interférences globales qui mettent en jeu l'outil, le porte-outil, et l'ensemble du voisinage. Au cours de cette thèse, seule la gestion de ce type d'interférences est traitée, ce qui permet d'obtenir une description plus détaillée de l'état de l'art.

2 Gestion classique des interférences globales en usinage 5 axes

L'étude de l'existant concernant l'évitement de collisions globales dans le domaine de l'usinage 5 axes est indispensable afin de comprendre les différentes méthodes proposées jusque là. En effet, ces interférences interviennent lors d'une collision entre l'outil et l'environnement extérieur (les surfaces de la pièce non usinées lors de la phase considérée seront assimilées à l'environnement) et peuvent avoir un impact sur la pièce usinée, mais également sur l'intégralité de la machine. Il est nécessaire de les éviter afin d'assurer la sécurité de l'utilisateur, et ne pas endommager la pièce usinée.

Dans le cadre de l'évitement de collisions globales, plusieurs méthodes existantes seront présentées permettant de résoudre ce problème d'un point de vue géométrique. Dans l'intention d'ordonner les travaux réalisés dans le domaine de l'usinage, l'arborescence 1.11 permet de classer les différentes démarches en fonction des étapes, amenant à l'évitement de collision idéal. Les modélisations existantes de l'ensemble outil/porte-outil, ainsi que de l'environnement sont abordées. Ensuite, les travaux portent sur les différentes méthodes de test de collisions mises en place dans la littérature. Enfin le dernier point traité est la compréhension des corrections et optimisations d'orientation dans le but de générer une trajectoire hors collision la plus lisse possible.

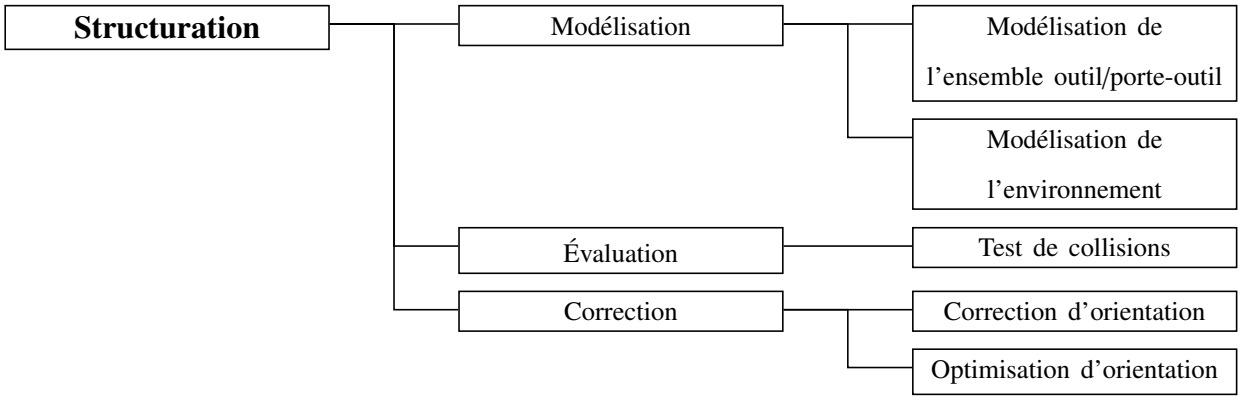


Figure 1.11 – Arborescence de l'état de l'art concernant la gestion des collisions globales

Les liens existants entre les différentes étapes de l'arborescence présentée précédemment sont donnés par le diagramme d'activité 1.12 venant détailler celui notifié en introduction.

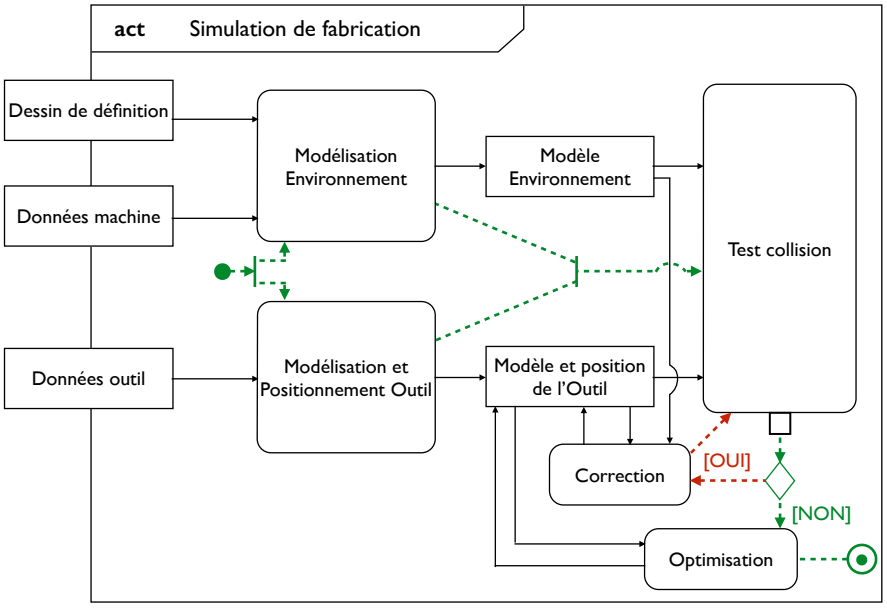


Figure 1.12 – Diagramme d'activité de la simulation de fabrication

Les flux principaux sur ce diagramme sont ceux de contrôles (en pointillés), la partie représentée en rouge représente une des problématiques liées à l'existant dans le domaine de l'usinage 5 axes. Il s'agit d'un des inconvénients de l'aspect itératif dans la littérature.

2.1 Modélisation de l'outil

En ce qui concerne la représentation de la géométrie outil, la plupart des modélisations obtenues sont issues du formalisme général APT [APT, 1983] utilisé dans de nombreux d'articles [Bohez et al., 2003] [Yau and Tsou, 2009]. Ce dernier normalise la géométrie d'un outil à l'aide de sept paramètres comme illustré en figure 1.13.

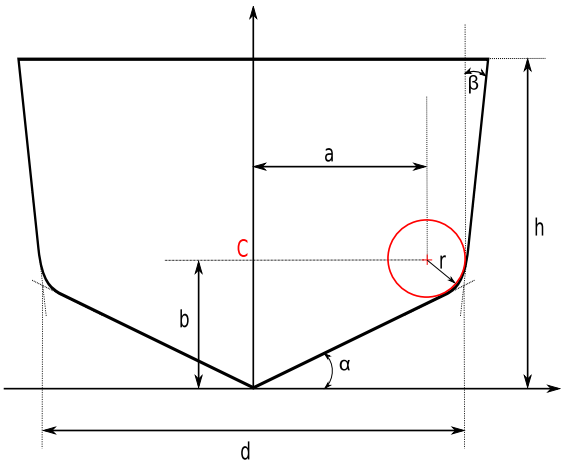


Figure 1.13 – Formalisme APT

L'avantage est que n'importe quelle forme d'outil normalisé peut être obtenue à l'aide de ce formalisme. Après avoir choisi la forme générale de l'outil employé, il est nécessaire de générer une

modélisation de ce dernier. Plusieurs types de méthodes peuvent être utilisées dans le cadre d'une étude de génération de trajectoire d'usinage. La première approche décrite est la décomposition de la modélisation de l'outil en surface implicite [Bi et al., 2010] [Ho et al., 2001].

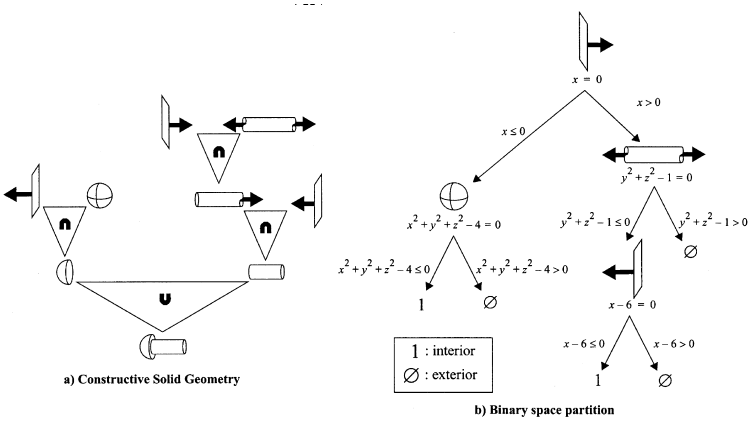


Figure 1.14 – Décomposition implicite de la géométrie outil [Ho et al., 2001]

Cette méthode représentée en figure 1.14 décompose le corps outil utilisé en un sous-ensemble défini de manière mathématique permettant de séparer chaque outil (non complexe), en éléments géométriques simples. Cette décomposition permet d'avoir des méthodes de calcul simplifiées à l'aide de ce fractionnement en surfaces implicites.

Afin d'assurer un meilleur traitement du modèle de l'obstacle dans le cadre d'un évitement de collisions, il est possible de générer une séparation de l'outil en quatre zones permettant de définir par avance diverses méthodes du traitement de la problématique de ces zones : frontale, arrière et latérale [Monies et al., 2004]. La figure 1.15 illustre cette décomposition.

— Zone 1 : $0 \leqslant X_i \leqslant R$, $-R \leqslant Y_i \leqslant 0$
— Zone 2 : $-R \leqslant X_i \leqslant 0$, $-R \leqslant Y_i \leqslant 0$
— Zone 3 : $0 \leqslant X_i \leqslant R$, $0 \leqslant Y_i \leqslant R$
— Zone 4 : $-R \leqslant X_i \leqslant 0$, $0 \leqslant Y_i \leqslant R$

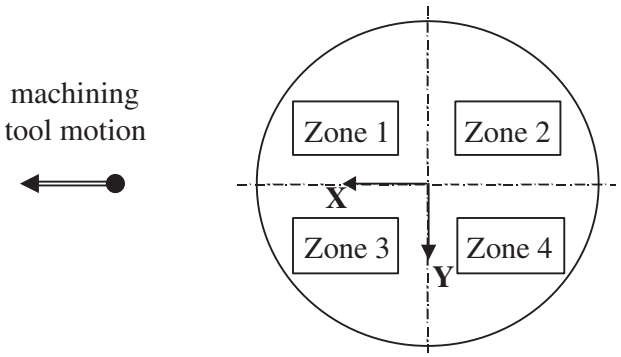


Figure 1.15 – Décomposition de l'outil en 4 zones d'étude [Monies et al., 2004]

Cette décomposition en zone d'action permet d'étoffer le nombre d'informations présentes dans la représentation de l'outil considéré afin de modifier la correction apportée selon la partie outil considérée. D'autres ajouts d'informations peuvent être établis sur la modélisation de l'outil utilisé en fonction des besoins ressentis. Dans le cas où l'outil utilisé est généré à l'aide d'une forme complexe (elle n'est pas décomposable en surfaces implicites), le maillage de ce dernier permet d'obtenir une géométrie discrète approchée [Schmitz et al., 2007]. Ce cas est illustré en figure 1.16.

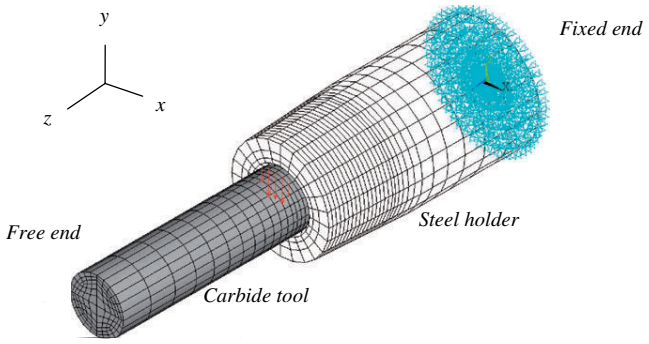


Figure 1.16 – Exemple de maillage d'un outil [Schmitz et al., 2007]

Il existe, bien évidemment, des modélisations d'outil à une échelle inférieure. En effet, les représentations des enveloppes, du nombre de dents, de l'usure de ces dernières peuvent être modélisées, mais cela n'a pas un fort intérêt dans le cadre de cette thèse étant donné que les collisions locales ne sont pas prises en comptes.

Après avoir décrit une partie des méthodes existantes afin de modéliser l'outil utilisé lors de la simulation, la modélisation de l'environnement ainsi qu'une partie des différents tests de collisions existants sont présentés.

2.2 Modélisation de l'environnement, test de collisions et correction d'orientation

Dans un premier temps, différentes méthodes de modélisation d'environnement sont présentées afin de répertorier une partie de l'existant. Pour être plus clair sur le partitionnement de l'environnement utilisé, la méthode C-Space est décrite.

2.2.1 Définition et utilisation du C-Space

Une méthode de désignation de volumes appelée le C-Space est régulièrement utilisée lors des travaux étudiés dans le domaine de l'usinage 5 axes [Choi et al., 1997].

Un ensemble Q est appelé un espace de configuration (C-Space) pour un système, si chaque élément de Q correspond à une configuration valide du système et si chaque configuration du système peut être identifiée par un unique élément de Q. Cela permet d'identifier le volume à la portée de l'outil de coupe dans un espace cartésien. La première exigence est que le C-Space doit contenir la C_L Surface qui représente la zone d'usinage accessible.

Le volume nommé C-Space a pour but de définir les zones accessibles et non accessibles pour l'outil utilisé. Cette méthode permet, dans un volume machine, de séparer les zones libres des zones occupées. Ainsi, il est possible de trouver un chemin sécurisé pour la trajectoire de l'outil à l'intérieur du C-Space comme noté en équation (1.1) :

$$(A \cap B) = \varnothing \tag{1.1}$$

avec :

— A : solide en déplacement,

— B : ensemble des obstacles présents.

La définition des différents éléments composant le C-Space est donnée après l'étude du volume total en équations (1.2) (1.3).

$$V_{total} = V_F \cup V_M \cup V_G \tag{1.2}$$

$$V_F \cap V_M \cap V_G = \varnothing \tag{1.3}$$

Avec :

— V_F : volume de C-Space désignant l'espace libre de collision appelé le Free Space,

— V_M : volume du C-Space désignant l'espace d'usinage appelé Machining Space,

— V_G : volume de gouging,

— S_D : surface design à usiner,

— S_S : surface représentant le brut à usiner.

Le C-Space est donc défini comme suit (équation (1.4)) :

$$C_{Space} \text{ ELEMENTS} = (S_S, S_D, V_F, V_M, V_G) \tag{1.4}$$

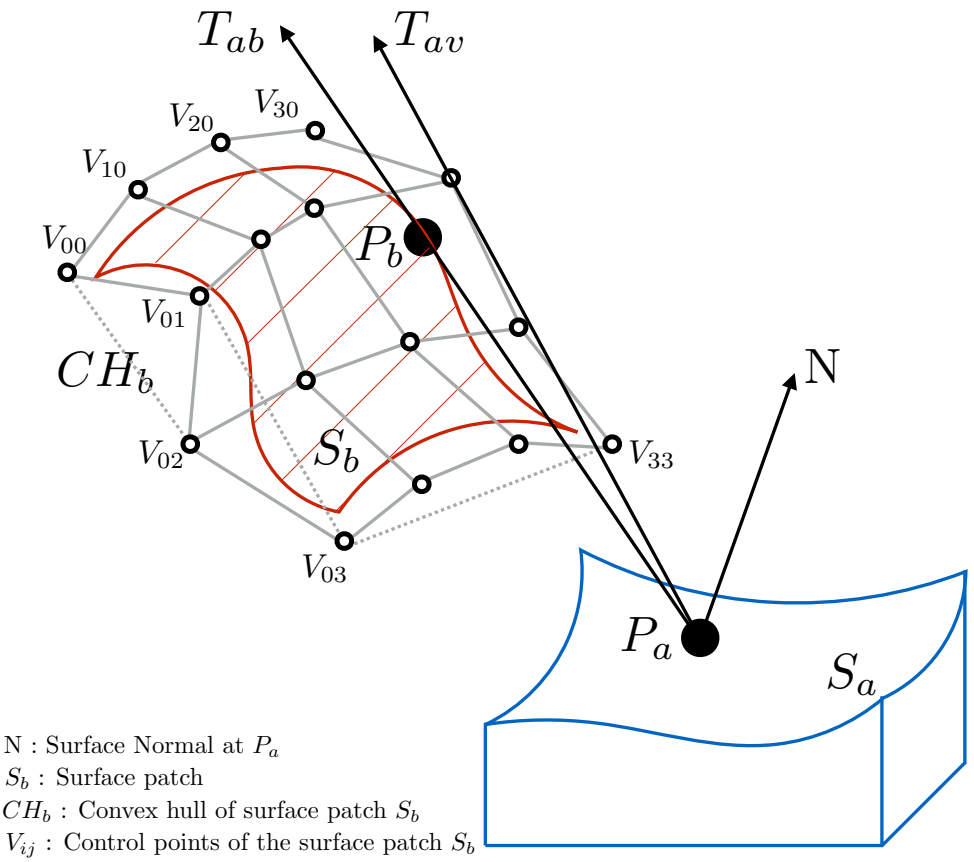
Dans le domaine d'évitement de collisions, de nombreux auteurs ont eu recours à la modélisation C-Space [Balasubramaniam et al., 2003] [Jun et al., 2003] [Lin et al., 2014] [Morishige et al., 1997].

Plusieurs travaux présentent différentes méthodes permettant d'assurer, d'un point de vue géométrique, un évitement de collision entre la modélisation de l'outil et l'environnement. Cette étude bibliographique porte sur les trois modèles les plus régulièrement utilisés : la génération d'enveloppes convexes, la prise en compte de sphères de Gauss et la notion de décalage.

2.2.2 Méthode des enveloppes convexes

La première méthode présentée est celle des enveloppes convexes [Lee and Chang, 1995]. Cette dernière permet de déterminer rapidement les positions outils possibles pour un point de la trajectoire. Pour ce développement, le postulat de départ est qu'une courbe paramétrique à pôles C_u se trouve toujours à l'intérieure de son polygone caractéristique [Leon, 1991]. Cette propriété s'applique aussi aux surfaces avec des enveloppes convexes qui dans ce cas sont modélisées comme des polyèdres. Afin d'illustrer cette méthode, reprenons l'exemple schématisé en figure 1.17 [Lee and Chang, 1995].

La surface notée S_a représente la surface à usiner et la surface notée S_b représentant l'obstacle. L'enveloppe convexe associée CH_b est créée et il est admis (d'après les caractéristiques des surfaces à pôles) qu'un axe outil passant par $P_a \in S_a$ en inclinaison progressive vient d'abord en contact avec CH_b avant de toucher S_b. Suite à cette constatation, il est alors possible de définir l'ensemble des orientations outil utilisables pour un point de la surface à usiner P_a.



N : Surface Normal at P_a
S_b : Surface patch
CH_b : Convex hull of surface patch S_b
V_{ij} : Control points of the surface patch S_b
T_{av} : Tool interfering orientation between P_a and convexe hull CH_b at V_{31}
T_{ab} : Tool interfering orientation between P_a and surface point at P_b
P_a and P_b : Surface points on surface patch S_a and S_b respectively

Figure 1.17 – Schéma illustrant le problème [Lee and Chang, 1995]

Cependant, cette méthode nécessite de tester l'ensemble des orientations outil possibles ce qui peut vite devenir laborieux. Une fois l'axe outil choisi, un problème subsiste, car l'ensemble des orientations possibles ne prend pas en compte la géométrie complète de l'outil. La seconde partie de la méthode est illustrée sur la figure 1.18.

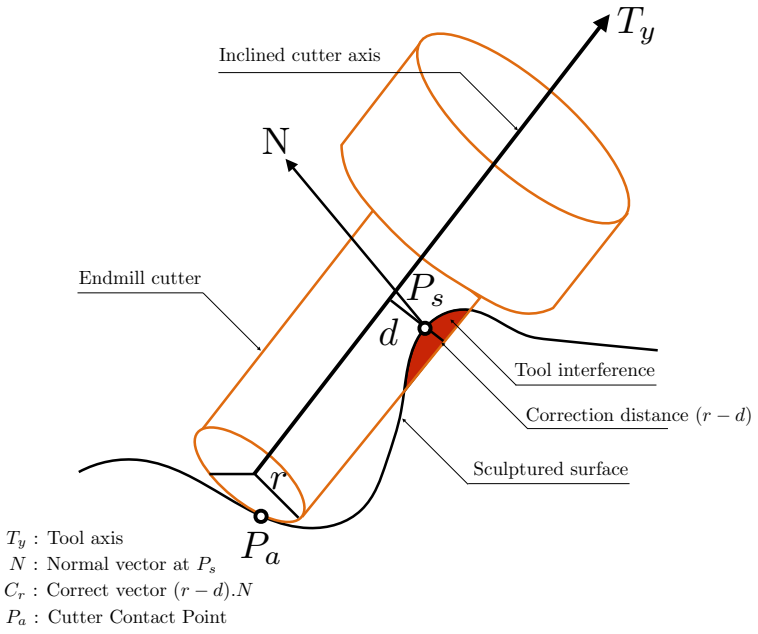
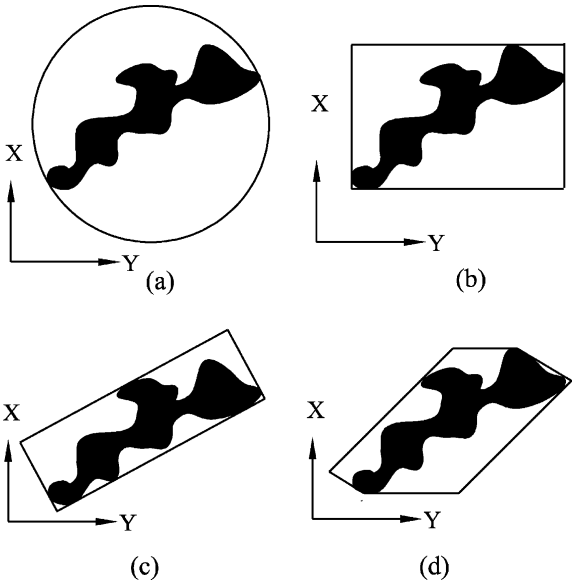


Figure 1.18 – Prise en compte de la géométrie outil [Lee and Chang, 1995]

Afin d'intégrer le rayon d'outil, un repère est associé à l'outil avec T_y l'axe de l'outil. Les coordonnées des points des surfaces avoisinantes sont explicitées et leur distance à l'axe outil est calculée (notée d). Il suffit par la suite de comparer cette distance d au rayon de l'outil r. La condition de modification détaillée est donnée en équation (1.5).

$$\begin{cases} d > r & \text{Pas de modification} \\ d < r & \text{Pivotement autour de } P_a \text{ pour déplacer } P_s \text{ d'une valeur de } (r-d) \text{ selon } \mathbf{N} \end{cases} \tag{1.5}$$

Cette méthode des enveloppes convexes permet un évitement des collisions en usinage d'un point de vue géométrique, à l'aide d'un recalage hors collision, si besoin est. Par contre, selon le degré de la surface, l'enveloppe convexe peut se retrouver très éloignée de la surface réelle. Les différentes méthodes d'obtention des enveloppes convexes ont chacune leur intérêt, ainsi l'utilisation d'*Oriented Bounding Box* [Ding et al., 2004] est comparée avec d'autres méthodes telles que *Bounding Sphere (BS)*, *Axis Aligned Bounding Box (AABB)* [Bergen, 1998] ou *Discrete Bounding Box (k-dop)*. La figure 1.19 illustre ces différentes enveloppes convexes englobant, dans ce cas, le même solide étudié.



(a) Bounding Sphere. (b) Axis aligned bounding box (AABB).
(c) Oriented bounding box (OBB). (d) Discrete bounding box (k-dop).

Figure 1.19 – Différentes méthodes de génération d'enveloppes convexes [Ding et al., 2004]

La première méthode présentée portant sur les *Bounding Sphere (BS)* illustrée en figure 1.19(a) génère une zone active sphérique englobant le solide concerné. Son avantage est la facilité de test pour le chevauchement entre deux sphères comme l'illustre la figure 1.20 [Chang et al., 2009]. Dans ce cas, l'utilisation de sphère est particulière, car elle sert à englober une *OBB* qui contient déjà le solide concerné.

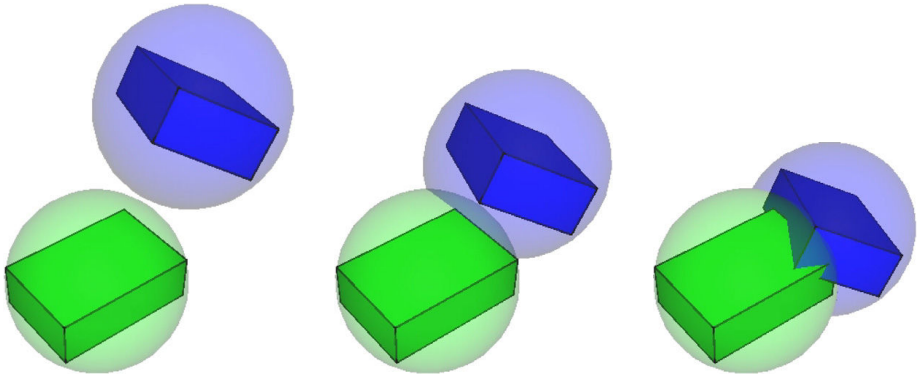


Figure 1.20 – Utilisation de *Bounding Sphere* englobant le solide considéré [Chang et al., 2009]

La deuxième méthode décrite notée *Axis Aligned Bounding Box* $(AABB)$, représentée en figure 1.19(b), utilise une boîte englobante de type pavé dont les trois bords sont parallèles aux axes x,y,z du système de coordonnées utilisé. Cela permet d'être rapide, car il est considéré, comme décrit en figure 1.21 [Cohen et al., 1995], des enveloppes toujours orientées de la même manière (seul son encombrement est modifié). Cependant, ici aussi, l'espace perdu peut être important (notamment lorsque le solide étudié se trouve à 45 °).

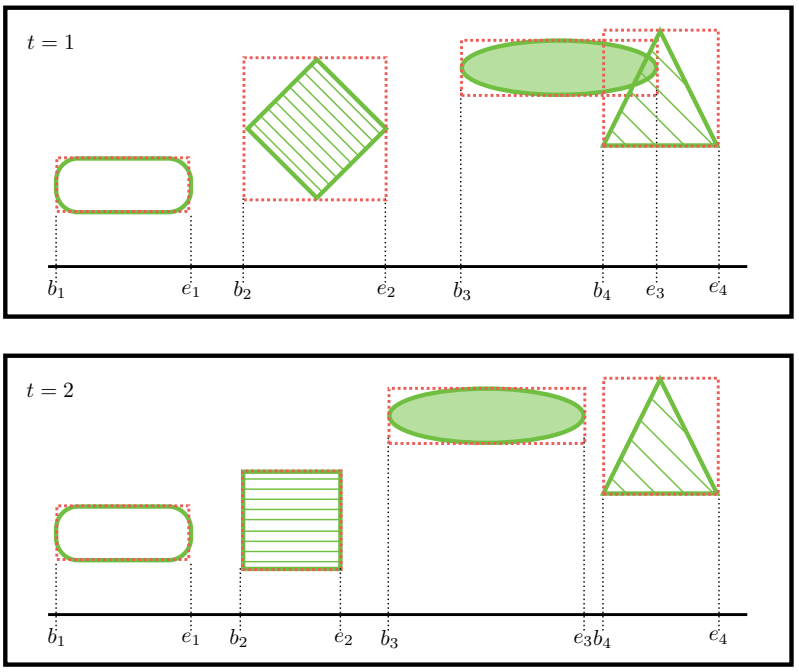


Figure 1.21 – Mise en place de *AABB* [Cohen et al., 1995]

Pour ce qui est de la méthode *Oriented Bounding Box* notée *OBB*, elle utilise une technique similaire à la stratégie *AABB* mais avec une orientation qui peut être variable au cours du temps par rapport au repère de travail offrant une assurance d'englobage complet du solide concerné [Barequet and Peled, 2001] [Ding et al., 2004]. Pour le test de collision, la méthode de séparation d'axe est utilisée afin de pouvoir tester de manière numérique la présence ou non de collision. La figure 1.22 illustre la méthode de détection de collision avec utilisation de la méthode *OBB*.

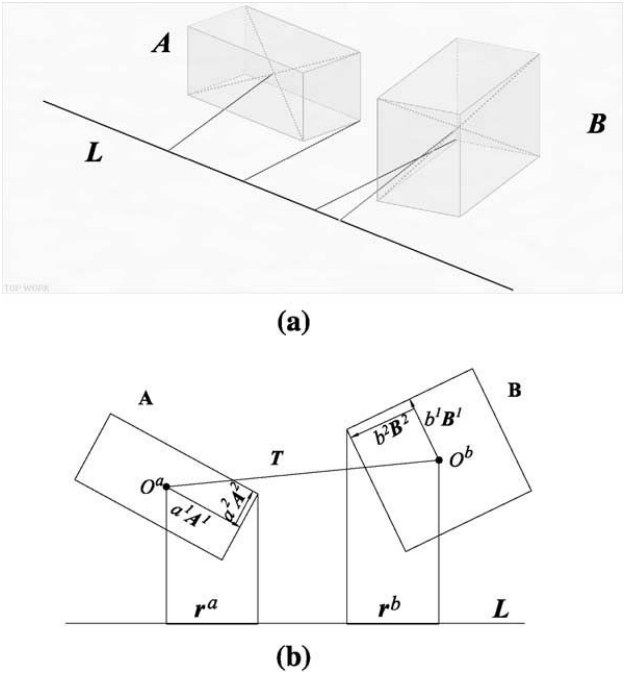


Figure 1.22 – Utilisation de la méthode *OBB* en 3D (a) et en 2D (b) [Ding et al., 2004]

Afin d'assurer la présence de non-collision, il faut que pour la projection considérée, donnée en équation (1.6), soit respectée.

$$|T.L| > r_a + r_b \tag{1.6}$$

La dernière méthode citée notée $k - dop$ (*Discrete bounding box*) qui est une intersection de plusieurs *OBB* permet de réduire encore les fautes d'association d'un espace libre à une zone obstacle tout en augmentant la difficulté de représentation. Un approfondissement de cette méthode [Gottschalk et al., 1996] consiste en une décomposition du solide concerné en plusieurs parties pour diminuer le volume considéré comme inaccessible. La figure 1.23 illustre, l'utilisation de la méthode *OBB* avec décomposition du solide concerné.

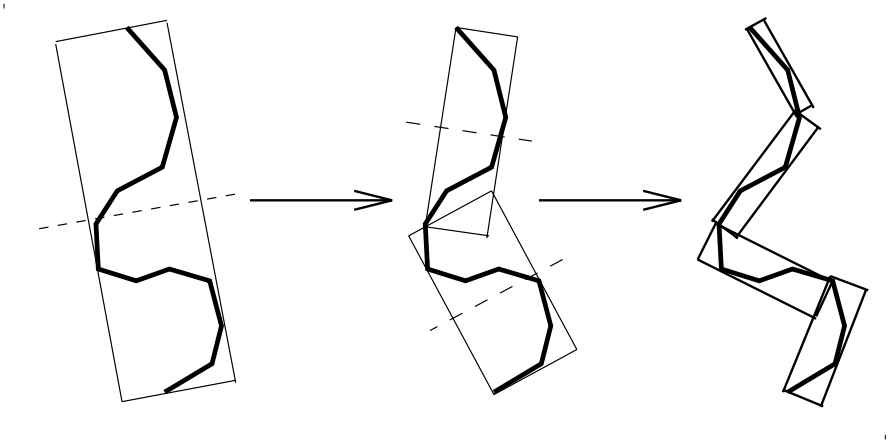


Figure 1.23 – Illustration de la méthode OBBTree [Gottschalk et al., 1996]

Ces différentes techniques assurent une connaissance de la présence ou non de collisions, mais peuvent s'avérer coûteuses, soit avec une solution enveloppante proposée pas suffisamment affinée soit, pour un cas contraire, un temps de calcul très important lors d'une complication de l'enveloppe convexe utilisée.

Une fois les différentes méthodes d'utilisation d'enveloppes convexes dans le cadre d'évitement de collision globale décrites, la suite concerne un autre moyen de résoudre ce problème qui est l'utilisation des sphères de Gauss (aussi appelée analyse de la visibilité).

2.2.3 Méthode des sphères de Gauss

La méthode des sphères de Gauss [Woo, 1994] permet de définir, selon la surface concernée, les zones d'accessibilité comme illustrées en figure 1.24. Il est alors possible de remarquer que pour chaque type de surface est décrite la sphère de Gauss correspondante.

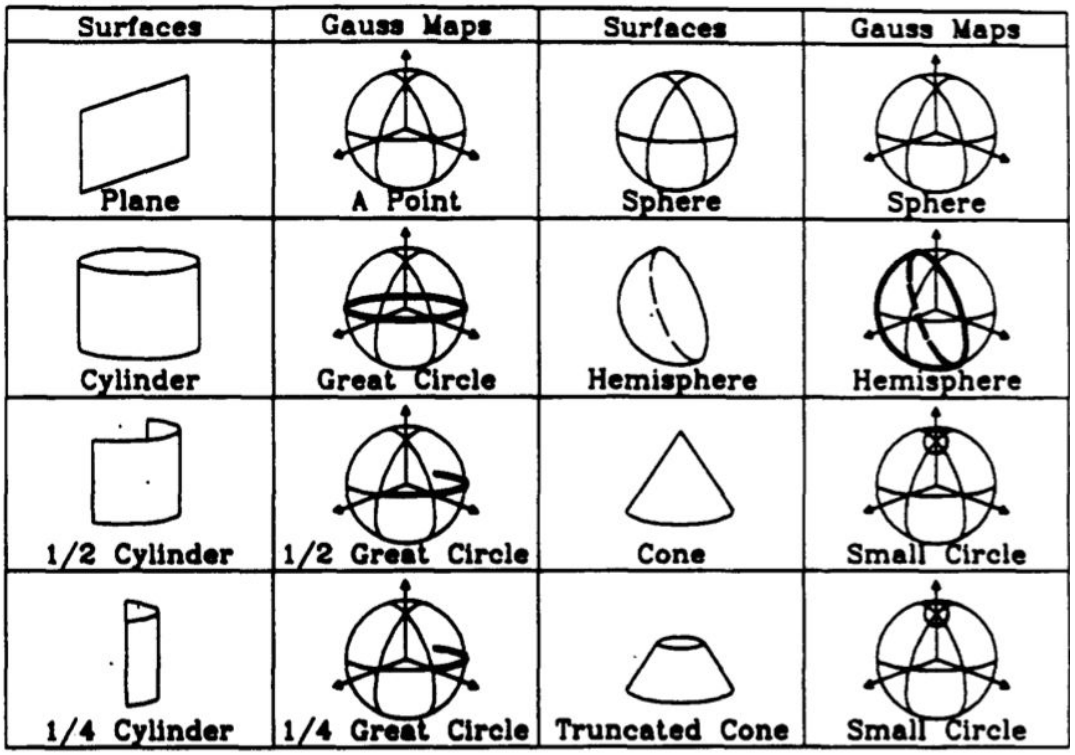


Figure 1.24 – Exemple de sphères de Gauss [Woo, 1994]

Les démarches utilisant des cônes de visibilité, ou des cartes de visibilité, sont directement rattachées aux sphères de Gauss obtenues, car il ne s'agit que de descriptions différentes d'un même traitement du problème sur le point étudié C_c et l'axe de l'outil **T** comme décrit en figure 1.25.

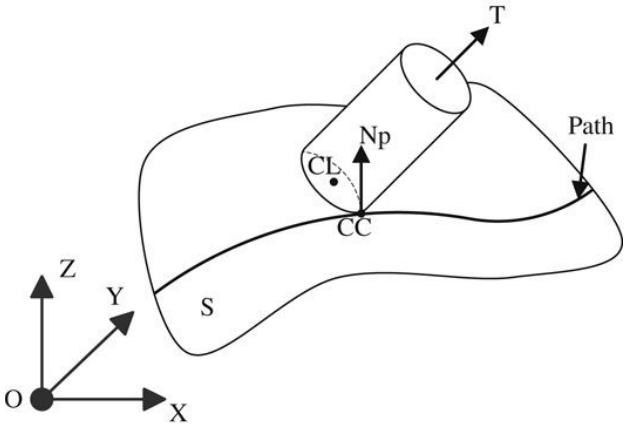


Figure 1.25 – Positionnement outil 5 axes [Wang and Tang, 2007]

L'utilisation de sphères de Gauss permet d'obtenir la visibilité locale attribuée au point étudié et fournit l'espace des orientations outil possible. Il faut bien distinguer la visibilité globale de la visibilité locale, la différence est illustrée sur la figure 1.26.

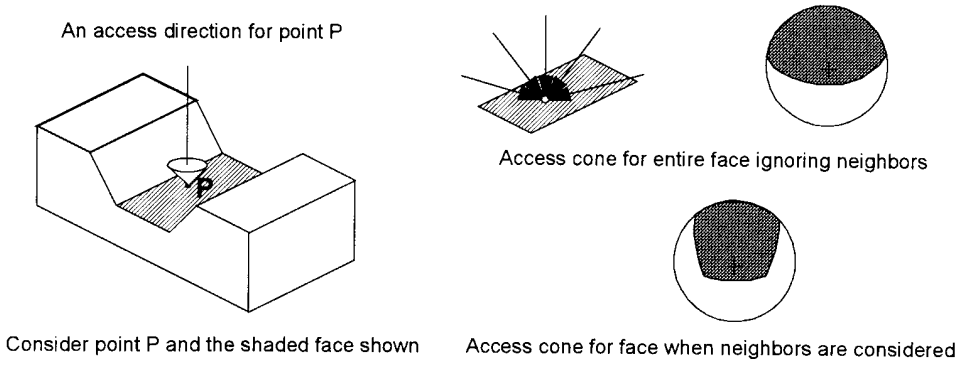


Figure 1.26 – Différence entre visibilité globale et locale [Balasubramaniam et al., 2000]

Sur la figure 1.26, la surface étudiée est un plan, de ce fait, la sphère de Gauss attribuée à la visibilité locale est une demi-sphère. Par la suite est tracé, sur une sphère, l'ensemble des orientations atteignables [Roberts and Rawat, 2007] en fonction des surfaces considérées comme obstacles. En effet, lors de la création de la visibilité locale, les surfaces voisines à la surface étudiée, à laquelle appartient le point P ne sont pas prises en compte. La visibilité globale restreint donc la première sphère de Gauss à un sous-ensemble qui prend en compte les surfaces avoisinantes (voir figure 1.26). La sphère de Gauss obtenue permet d'obtenir l'ensemble des orientations possibles pour l'axe outil au point étudié P.

La première partie de cette méthode est développée pour la mise en place des cônes de visibilité sur une surface [Balasubramaniam et al., 2000] [Woo, 1994]. La notion de cône de visibilité prenant en compte les collisions globales telle que cela est illustrée en figure 1.27.

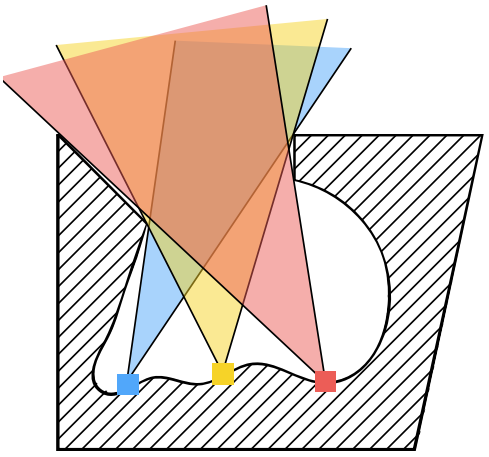


Figure 1.27 – Cônes de visibilité [Balasubramaniam et al., 2000]

Une autre modélisation similaire est basée sur l'utilisation de cartes de visibilité [Hu et al., 2013] [Makki, 2009] [Wang and Tang, 2007]. La position et l'orientation de l'outil sont alors connues par la connaissance du couple (C_L,$\mathbf{T}$) (figure 1.25). Il est alors possible de créer la carte de visibilité (VMAP) correspondante qui s'obtient par projection de l'axe outil possible sur

le plan normal à la surface tel qu'illustré en figure 1.28 à l'aide de la condition (1.7).

$$\mathbf{T}(\alpha,\beta).\mathbf{N}_p \begin{cases} \geqslant 0 & \text{Valide} \\ < 0 & \text{Invalide} \end{cases} \tag{1.7}$$

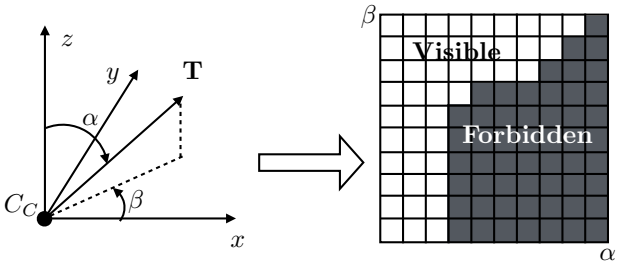


Figure 1.28 – Création de la carte de visibilité [Wang and Tang, 2007]

Cette carte de visibilité représente une discrétisation de l'espace afin de projeter le vecteur **T** selon les angles α et β. Après avoir généré cette carte de visibilité, il est alors possible de s'attaquer au problème de prise en compte de la géométrie réelle de l'outil, dans le but de construire la carte de visibilité réelle. Pour ce faire, une intersection entre la carte de faisabilité initiale et la carte de visibilité correspondante est effectuée pour définir la variation angulaire programmable en fonction des itérations avec un évitement d'obstacle. La figure 1.29 montre l'étape de création de la carte de faisabilité prenant en compte l'environnement machine et permettant de choisir la variation de l'angle étudié.

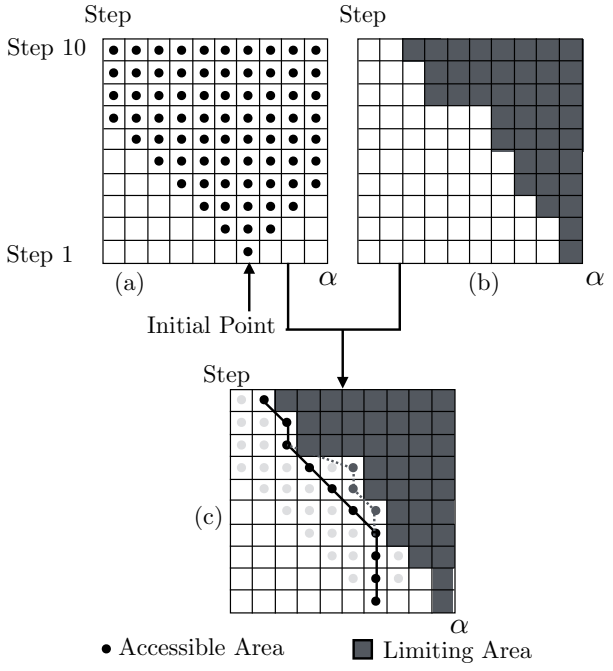


Figure 1.29 – Modification de la carte de faisabilité [Wang and Tang, 2007]

Sur le schéma 1.29, le tableau (a) représente la carte de faisabilité initiale, le tableau (b) la carte de visibilité (VMAP) et la les différentes possibilités de quantification de l'angle étudié sont

données en figure (c). Cette réduction d'espace disponible présentée à l'aide d'une carte de visibilité peut également être représentée avec une schématisation des outils utilisés et l'utilisation de cônes de visibilités. La figure 1.30 illustre la démarche adoptée avec la prise en compte de la machine utilisée venant réduire le cône de visibilité disponible (pour un seul pas de calcul représenté) [Makki, 2009] .

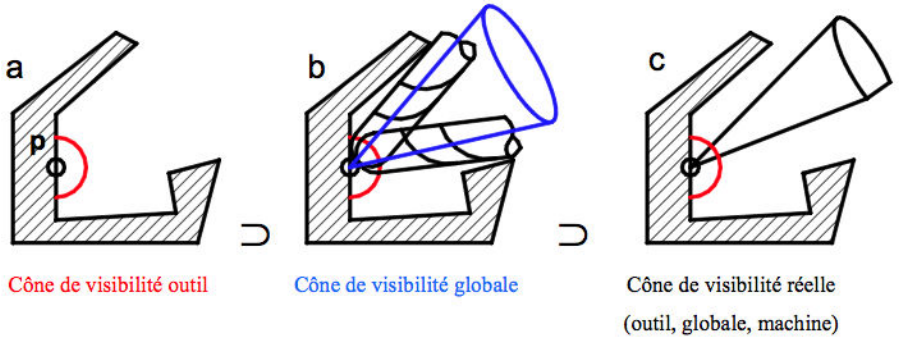


Figure 1.30 – a-visibilité locale b-visibilité globale c-visibilité réelle [Makki, 2009]

Les résultats obtenus sont très satisfaisants d'un point de vue de l'évitement des collisions comme l'illustre la figure 1.31. Cependant ces différentes méthodes génèrent une explosion combinatoire nécessitant un temps de calcul très important pour déterminer toutes les possibilités d'orientations restantes.

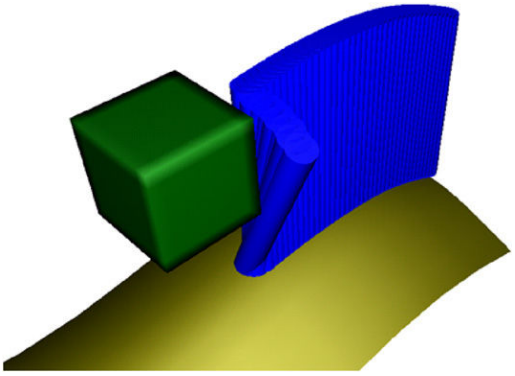


Figure 1.31 – Évitement d'obstacle [Wang and Tang, 2007]

Afin de finaliser l'état de l'art de la partie usinage 5 axes, une modélisation complète de l'environnement machine en distinguant plusieurs types d'obstacles et donc plusieurs cartes de faisabilité est présentée [Hu et al., 2013]. La figure 1.32 illustre la réduction supplémentaire de la zone de visibilité avec l'environnement complet.

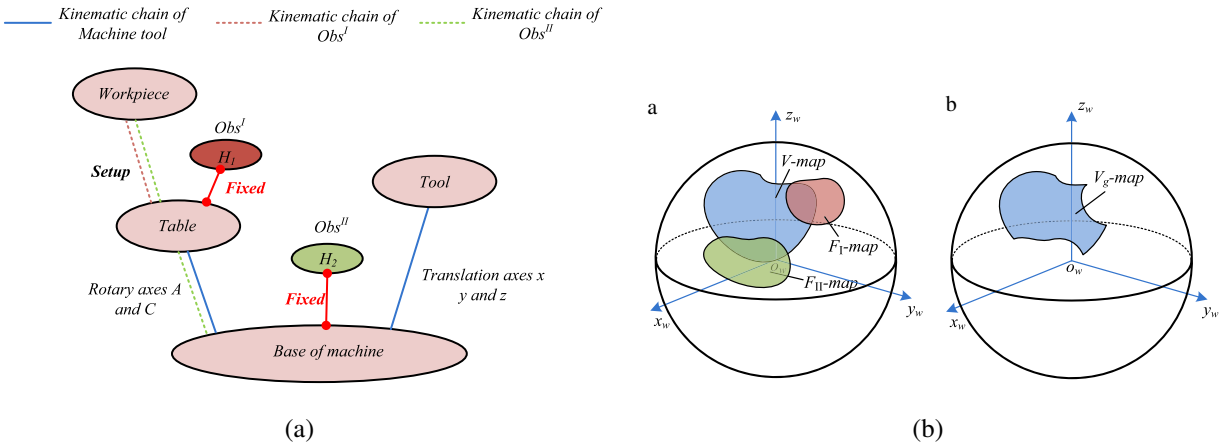


Figure 1.32 – Restriction de la carte de visibilité globale [Hu et al., 2013]

Ces travaux prennent en compte un grand nombre de paramètres permettant d'assurer une non-collision plus complète. Cependant cela entraîne une multiplication des temps de calcul en fonction de la discrétisation de la carte de visibilité utilisée. Plus cette dernière sera fine, plus la prise en compte d'une collision sera assurée, mais plus l'explosion combinatoire des différentes solutions à étudier sera importante.

Les travaux concernant l'utilisation de sphères de Gauss et de cartes de visibilité ont été approfondis par une utilisation de polygones convexes [Hsueh et al., 2007], une utilisation du C-Space et d'une approche dynamique [Balasubramaniam et al., 2002] ou l'utilisation d'une *RGB map* [Wang et al., 2006].

2.2.4 Application d'un décalage par rapport à la surface obstacle

Une technique d'évitement de collisions impliquant la réalisation d'un décalage (d'une valeur et d'une direction à déterminer) par rapport à la surface considérée comme obstacle [Morishige et al., 1997] est également utilisée. Leur travail consiste (en utilisant la méthode C-Space présentée précédemment) à définir une zone d'orientation possible pour chaque itération en connaissant l'espace de collision et en utilisant la création d'une zone *Offset* permettant d'assurer une présence en zone libre. Cette démarche est illustrée en figure 1.33.

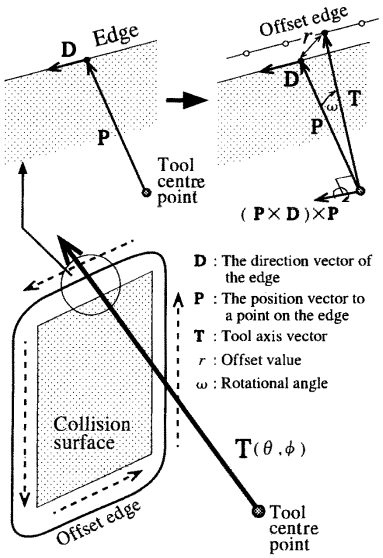


Figure 1.33 – Présentation de la méthode de réorientation [Morishige et al., 1997]

La nouvelle orientation limite de l'axe outil **T** est alors obtenue comme décrit en équation (1.8).

$$\mathbf{T} = \mathbf{P}.\mathbb{R}^{v}(\omega) \tag{1.8}$$

Avec $\mathbb{R}^{v}(\omega)$: La matrice de rotation de l'angle ω (donné en équation (1.9)) autour du vecteur **v** décrit en équation (1.10) et **P** le vecteur entre le point centre outil et le bord de l'environnement étudié.

$$\omega = sin^{-1}(\frac{r}{||\mathbf{P}||}) \tag{1.9}$$

$$\mathbf{v} = (\mathbf{P}x\mathbf{D})x\mathbf{P} \tag{1.10}$$

Le résultat obtenu à la suite de ces travaux est illustré en figure 1.34 où il est possible de constater une prise en compte du couple d'angles $\{\theta,\phi\}$ limites permettant d'assurer une non-collision puis une vision finale de la nouvelle trajectoire de l'ensemble de l'outil obtenue (sans modification de la partie usinée).

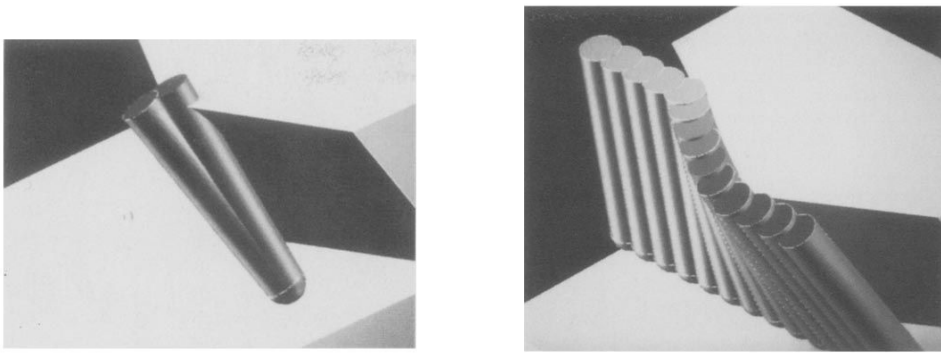


Figure 1.34 – Résultat de la simulation [Morishige et al., 1997]

D'autres travaux utilisent également la notion de décalage par rapport aux surfaces obstacles permettant d'obtenir l'ensemble des orientations outils disponibles dans la configuration étudiée [Morikawa et al., 2006] [Umehara et al., 2006]. La figure 1.35 (sur la gauche) illustre la méthode utilisée avec la génération, en fonction de la trajectoire programmée et de la longueur outil utilisée, de la courbe paramétrique sur la surface obstacle considérée comme limite. Par la suite, la prise en compte du rayon de l'outil utilisé et de la sécurité estimée nécessaire est réalisé à l'aide d'une rotation de l'axe outil autour de la direction d'avance (notée V).

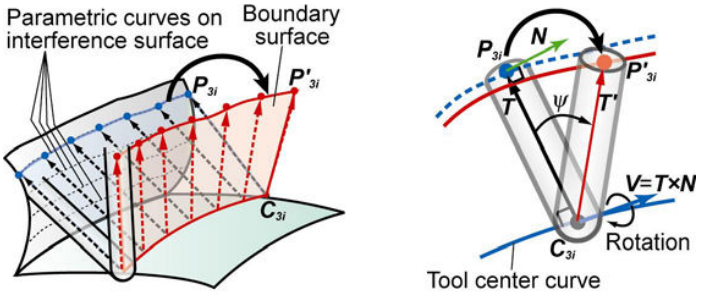


Figure 1.35 – Méthode de détermination de l'orientation admissible de l'axe outil [Morikawa et al., 2006]

Une zone indiquant la non-présence d'interférence avec un décalage est alors obtenue et permet de prendre en compte la géométrie de l'outil utilisé avec une possibilité de génération de trajectoire lisse de manière continue. La figure 1.36 donne une représentation de ce dit domaine sans interférence (*Interference-free area*).

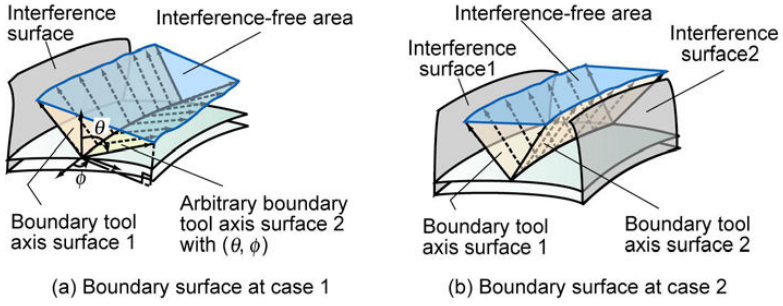


Figure 1.36 – Zone libre d'interférences dues aux surfaces obstacles [Morikawa et al., 2006]

D'autres méthodes permettent également d'effectuer des tests de collisions et des corrections d'orientation (ou de position) de diverses manières telles que recherche d'*Open Region* [Gian et al., 2003], la recherche de la *Lower envelop* représentant l'enveloppe de trajectoires comportant le moins de collisions [Ilushin et al., 2005] et l'utilisation de la méthode *Snakes/Ladders* [Ahmad et al., 2011]. Ces dernières ne seront pas plus développées afin de ne pas surcharger ce document bien que ces différents cas soient tout à fait utilisables dans le cadre de cette problématique.

2.3 Optimisation de la trajectoire obtenue par interpolation

L'étude d'optimisation de la trajectoire obtenue permet de relier directement la solution de réorientation obtenue, avec les méthodes précédentes, à l'interpolation nécessaire afin d'obtenir une modification d'orientation outil lisse.

Afin d'apporter une correction efficace aux détections de collisions entre outil et obstacles, plusieurs méthodes ont été développées à la suite des travaux présentés précédemment sur la détection de collision. Tout d'abord l'utilisation d'une interpolation des orientations de l'axe outil dans le dessein de déterminer l'orientation adaptée de l'outil au franchissement de l'obstacle est décrite [Ho et al., 2003]. Pour ce faire, la modélisation et le paramétrage utilisés sont représentés en figure 1.37.

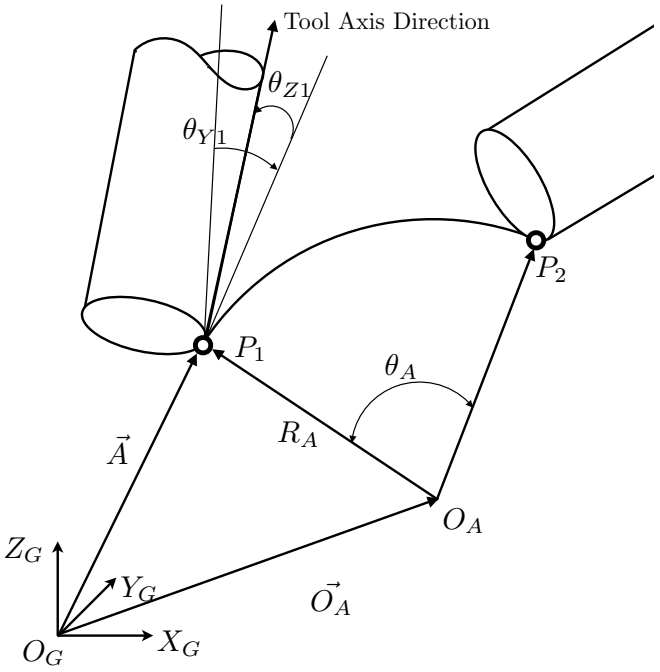


Figure 1.37 – Paramétrage de la problématique [Ho et al., 2003]

La matrice de rotation exprimée en équation (1.11) permet d'avoir une vision de la réorientation.

$$\mathbb{R}_y(\theta_y).\mathbb{R}_z(\theta_z) = \begin{bmatrix} cos(\theta_y) & 0 & sin(\theta_y) \\ 0 & 1 & 0 \\ -sin(\theta_y) & 0 & cos(\theta_y) \end{bmatrix} . \begin{bmatrix} cos(\theta_z) & -sin(\theta_z) & 0 \\ sin(\theta_z) & cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.11}$$

Après application du lissage linéaire décrit en équation (1.12) et (1.13) :

$$\boldsymbol{\mathcal{A}}(\lambda) = (1-\lambda).\boldsymbol{\mathcal{A}}_1 + \lambda.\boldsymbol{\mathcal{A}}_2 \tag{1.12}$$

$$\boldsymbol{O}_A(\lambda) = \mathbb{R}_y(\theta_{y1} + \lambda.\Delta\theta_y).\mathbb{R}_z(\theta_{z1} + \lambda.\Delta\theta_z).\boldsymbol{O}_A \tag{1.13}$$

le résultat est illustré en figure 1.38.

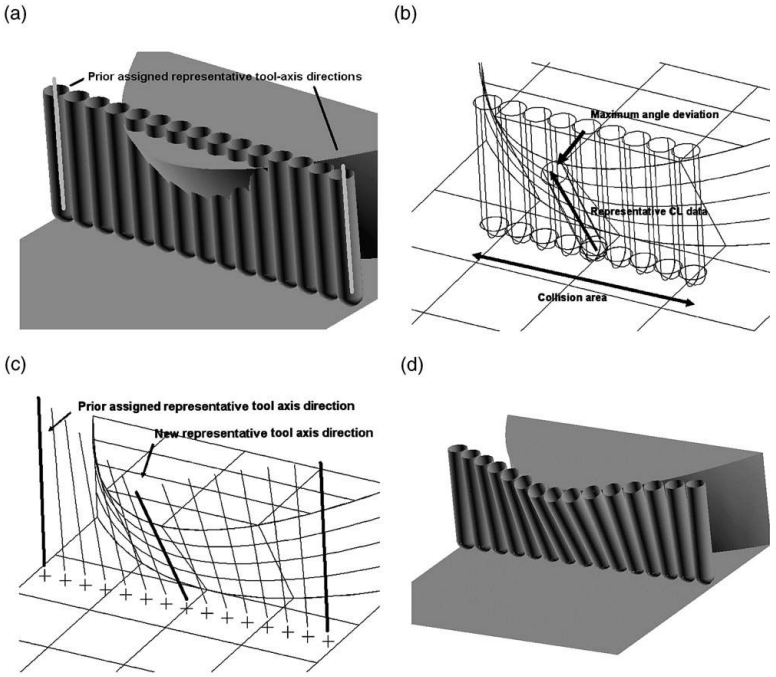


Figure 1.38 – Résultat de la simulation [Ho et al., 2003]

Dans ce cas, la correction apporte effectivement le relissage de la trajectoire attendu, passant par les trois positionnements d'axes définis. L'optimisation est alors réalisée de point en point, c'est-à-dire que les orientations fixées initialement restent les mêmes. Une amélioration serait de travailler de zone en zone. Cette liberté supplémentaire permettrait d'obtenir une trajectoire plus lisse après correction.

L'une de ces applications est dite de *morphing* et apporte une amélioration à la visibilité présentée précédemment en prenant en compte les itérations représentant la trajectoire à suivre [Zhiwei et al., 2012]. Cela permet d'être plus continu en terme de variation d'orientation. La figure 1.39 illustre la problématique pour réaliser le lien entre les différents points. Sur chaque position de calcul, la zone grisée représente la partie où la position de l'outil étudié, en fonction de la position de son point C_C et de son angle de rotation β autour de **f**, est dans la zone admissible.

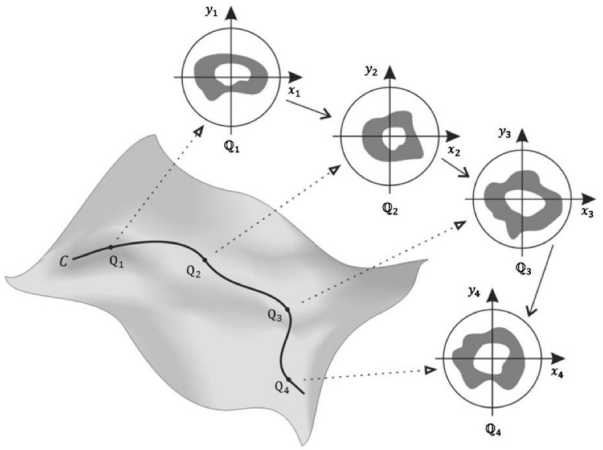


Figure 1.39 – Sphères de Gauss obtenues le long de la trajectoire [Zhiwei et al., 2012]

Ainsi la méthode décrite en figure 1.40 permet d'aboutir à un déplacement plus lisse selon les variations angulaires obtenues lors de l'évitement de l'obstacle (avec, comme précédemment, l'utilisation d'un lissage linéaire [noté $Q(\lambda)$]).

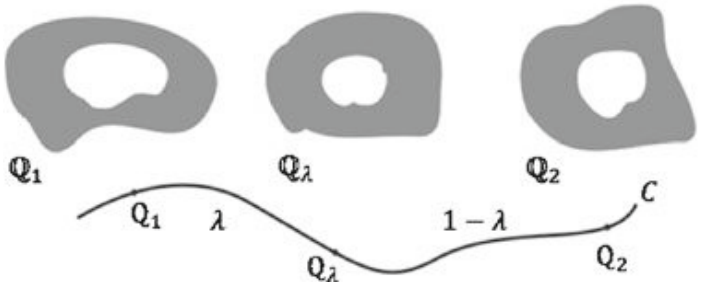


Figure 1.40 – Évolution de l'angle due à l'interpolation linéaire entre Q_1 et Q_2 [Zhiwei et al., 2012]

De plus, le fait de créer une fonctionnelle permettant de passer d'une zone à l'autre apporte un autre avantage qui est la minimisation de la longueur outil utilisée [Bi et al., 2010] [Cho et al., 2012]. En effet, il est nécessaire, entre deux points d'étude, de définir un critère permettant de fixer la valeur maximale de modification d'orientation. Ce dernier est décrit en équation (1.14) (avec une évolution de deux angles représentant la méthodologie 5 axes utilisée) [Bi et al., 2010].

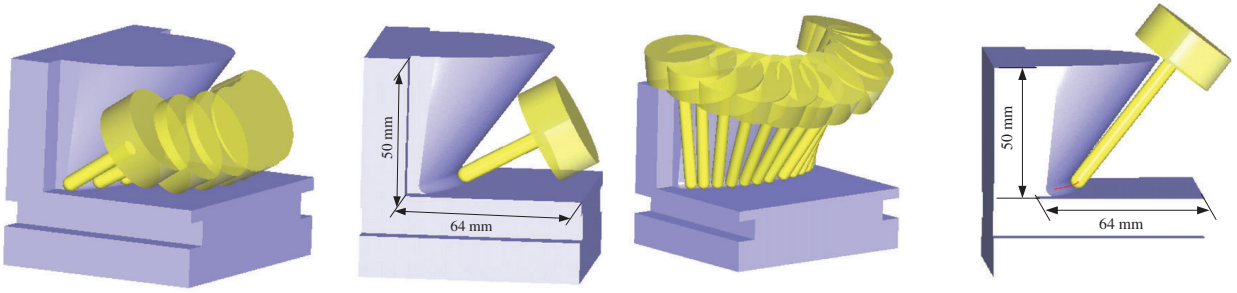
$$\text{Critère retenu} = \max_i\left(\sum_i\left(\sqrt{(\alpha_{i+1}-\alpha_i)^2+(\beta_{i+1}-\beta_i)^2}\right)\right) < \theta_t(d) \tag{1.14}$$

Avec :

— d : la distance entre deux points étudiés,

— $\theta_t(d)$: la variation maximale d'orientation acceptable en fonction de la distance parcourue.

La figure 1.41 illustre les résultats obtenus avec une mise en évidence de l'impact de la hauteur outil utilisée sur l'inclinaison nécessaire à l'évitement de collision.



(a) Fort angle d'inclinaison (b) Longueur outil importante

Figure 1.41 – Choix de la longueur outil [Bi et al., 2010]

La dernière solution de minimisation de changement d'orientation présentée au cours de ces travaux est la création d'un Domaine d'Orientation Admissible (DOA) défini dans l'espace pièce

[Castagnetti et al., 2008]. La création de ce dernier permet d'exprimer les degrés de liberté autorisés afin de réaliser le lissage de l'orientation. Par la suite, cette solution est traduite dans le repère machine comme illustré en figure 1.42.

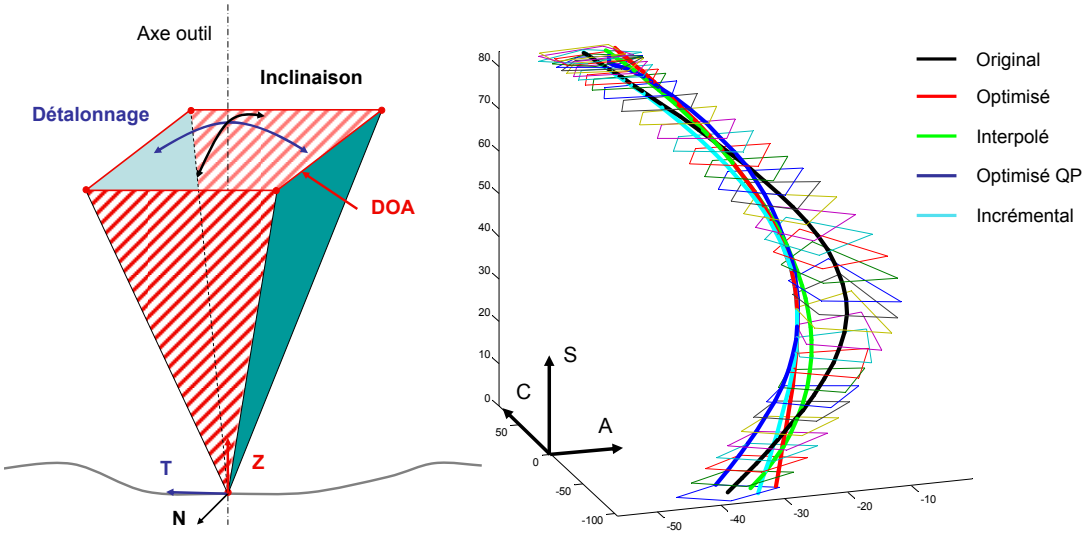


Figure 1.42 – DOA et lissage des axes rotatifs [Castagnetti et al., 2008]

L'optimisation est alors obtenue avec une fonction objectif, qui permet de minimiser (en accélération) la brutalité du changement d'orientation dans le repère machine, donnée en équation (1.15).

$$\text{Fonction Objectif} = \min_i[\sum_i((\frac{\partial A_i}{\partial S_i})^2 + (\frac{\partial C_i}{\partial S_i})^2), \in (\text{DOA})] \tag{1.15}$$

Ce travail sur la fluidité de la trajectoire obtenue dans l'espace machine permet de réaliser le lissage d'une trajectoire programmée comme illustré en figure 1.43.

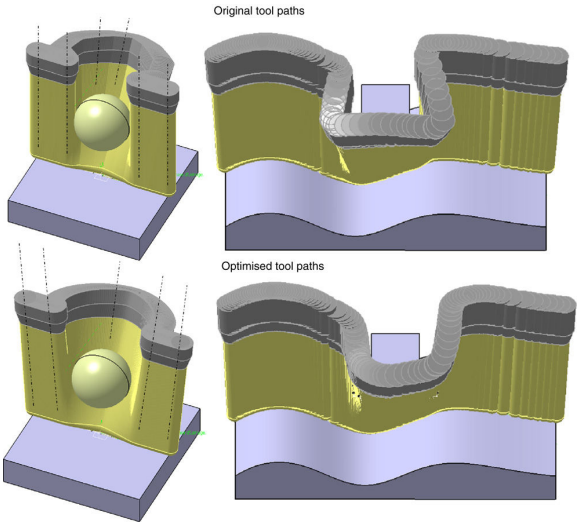


Figure 1.43 – Différence entre le positionnement outil initial et l'optimisé [Castagnetti et al., 2008]

Une fois le DOA généré, il est possible d'optimiser le changement d'orientation de l'axe outil en jouant sur la dérivée seconde de la position (notion d'accélération). Pour y parvenir, un axe de

travail intéressant est l'utilisation d'un critère de fluidité qui est maximisé [Pechard, 2009] où le raisonnement est basé sur la minimisation de l'énergie de déformation permettant l'obtention d'une trajectoire plus lisse. Ce dernier critère de fluidité peut être réalisé de nombreuses façons selon le besoin ressenti, il est alors possible de convenir, comme décrit précédemment dans l'espace machine, de travailler sur les dérivées secondes, mais il est également possible, afin d'accroître la sensibilité, d'utiliser une limitation de jerk. Un indicateur de fluidité de l'orientation basé sur la norme des dérivées troisième de l'orientation par rapport à l'abscisse curviligne donné en équation (1.16) peut être utilisé [Beudaert, 2013].

$$\text{indicateur de fluidité de l'orientation} = \sqrt{(\frac{d^3 i}{ds^3})^2 + (\frac{d^3 j}{ds^3})^2 + (\frac{d^3 k}{ds^3})^2} \tag{1.16}$$

Ce critère permet d'effectuer des lissages de trajets 5 axes notés dans le repère local au niveau des changements de direction et assurant une fluidité de la trajectoire obtenue.

Les différentes méthodes d'optimisation de trajectoires par interpolation présentées dans cette sous-partie montrent l'intérêt porté à l'obtention de trajectoires lisse afin d'augmenter le degré de continuité obtenu par rapport à la correction basique. Il est alors clair que plusieurs compromis (notamment au niveau de la longueur outil utilisée) sont à faire afin d'obtenir le lissage souhaité. Par contre ces différentes méthodes n'assurent pas forcément une convergence d'optimisation, car certaines trajectoires ne permettraient pas d'obtenir un minimum global. L'utilisateur peut donc se retrouver forcé de travailler avec un minimum local.

2.4 Conclusions

Un grand nombre de méthodes de modélisation de l'outil, de l'environnement, des tests de collisions, des corrections et optimisation des orientations a été présenté dans cette partie. Afin d'être le plus clair possible, la description fut décomposée en trois sous parties dans le but de présenter un fragment des méthodes existantes permettant, dans un premier temps, de modéliser l'outil utilisé. Par la suite, trois méthodes principales de détection de collisions et de correction de positionnement ont permis de comprendre les programmations actuellement mises en place dans la littérature. Le dernier point abordé au cours de cet état de l'art est la présentation de techniques permettant une optimisation des trajectoires obtenues, la plus couramment effectuée par une interpolation.

Bien que cette dernière ait été développée de différentes manières permettant effectivement un lissage très utile pour les machines utilisées ; cette étape présente tout de même un caractère à relever, elle est itérative et peut provoquer des explosions combinatoires comme cela est souligné en rouge sur le diagramme d'activité 1.12 représenté en introduction de cette étude. Par conséquent, les temps de calcul afin de trouver la nouvelle orientation de l'axe outil étudié deviennent très importants. Pour passer à un traitement du problème plus continu, il paraît utile d'étudier des travaux principalement développés dans le domaine de la robotique mobile, répondant à la même

problématique d'évitement de collisions en utilisant des champs répulsifs, permettant aux surfaces considérées comme obstacles de repousser le robot manipulé avant collision.

Ce modèle bâti sur des critères physiques du mouvement permet de fixer la solution sans avoir recours à une méthode itérative.

3 Gestion des interférences globales avec différentes approches potentielles

Cette partie permet de décrire les méthodes d'évitement de collisions classiquement utilisées dans le domaine de la robotique mobile. Bien que les problématiques de précision en terme de positionnement ne soient pas du même ordre de grandeur dans les domaines de la robotique mobile et de l'usinage, l'assurance d'un évitement de collision représente une aussi grande importance. Il paraît donc intéressant d'étudier les approches proposées dans ce domaine pour juger la possibilité et l'intérêt d'adaptation à l'usinage 5 axes.

En ce qui concerne la robotique mobile, la méthode la plus couramment employée dans le but d'éviter des obstacles présents sur la trajectoire programmée du robot est la prise en compte de points répulsifs (représentant des obstacles) et de points attractifs (représentant les points objectifs des trajectoires concernées) générant des potentiels amenant à la formulation d'efforts.

3.1 Effort dérivant d'un potentiel, champ de vecteurs

En notant le champ vectoriel $\mathcal{F}$ et $\mathcal{U}$ le potentiel scalaire associé, le champ $\mathcal{F}$ est alors défini comme dérivant du potentiel $\mathcal{U}$:

$$\mathcal{F} = -\mathbf{grad}(\mathcal{U}) \tag{1.17}$$

Ce champ vectoriel peut être alors représenté par des vecteurs normaux, en tout point, aux différentes surfaces équipotentielles. Pour un champ vectoriel donné, le potentiel correspondant n'est pas unique, il est défini à une constante près. Cela n'est pas une contrainte dans le cadre de cette étude, car le champ de vecteurs est issu du potentiel scalaire. Le signe utilisé est choisi par convention. Une représentation graphique de ce potentiel accompagné de l'effort correspondant est donnée en figure 1.44. L'effort y est alors représenté par les vecteurs, et les potentiels par les demi-sphères.

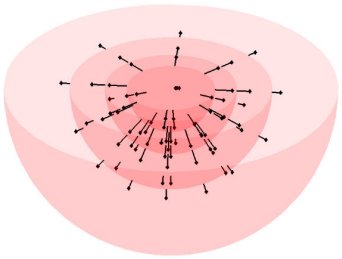


Figure 1.44 – Champ potentiel et champ vectoriel correspondant

Après avoir décrit le type d'effort utilisé classiquement, la partie suivante concerne l'étude des travaux utilisant cette méthode dans le cadre de la robotique mobile.

3.2 Utilisation de champs et de potentiels dans le domaine de la robotique mobile

Pour la gestion des collisions en robotique mobile par utilisation d'efforts à distance, plusieurs articles sont indispensables dans le cadre d'une analyse de l'existant. Tout d'abord, cette méthode a été initialement proposée dans les travaux d'Oussama Khatib [Khatib, 1980] [Khatib, 1986]. Il est expliqué comment sont gérées les collisions entre un robot mobile et divers obstacles, avec une utilisation de potentiels répulsifs. La méthode consiste à permettre à un robot mobile d'éviter des obstacles se trouvant sur le chemin reliant son point de départ à son point objectif.

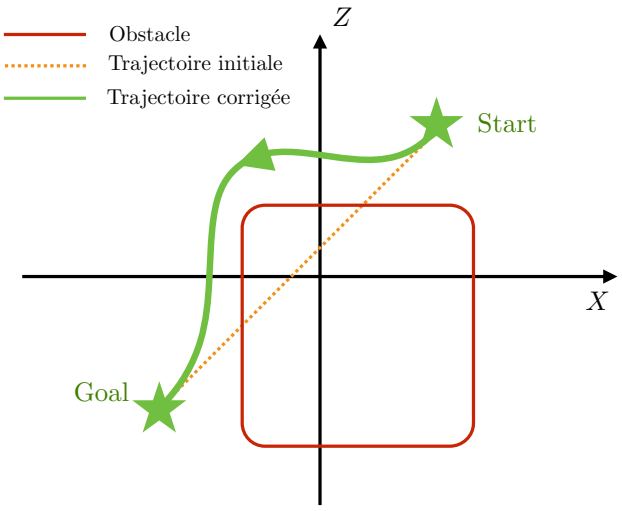


Figure 1.45 – Objectif des travaux étudiés [Khatib, 1980]

La trajectoire désirée, représentée en figure 1.45, est obtenue par l'application au robot d'un effort attractif à partir du point nommé point GOAL. Cependant, si le voisinage configuré initialement de l'obstacle inclut une partie de la trajectoire programmée entre le point initial et le point

GOAL, l'effort permettant de repousser le robot, décrit comme étant une fonction notée en équation (1.18) sur le point du robot étudié $N_k(x_k,y_k,z_k)$, est activé.

$$\mathcal{F}_{iR}(x_k,y_k,z_k) \triangleq [\frac{\partial V_i}{\partial x_k}\frac{\partial V_i}{\partial y_k}\frac{\partial V_i}{\partial z_k}]^T \tag{1.18}$$

Avec $V_i(x,y,z)$ une fonction potentielle définie en équations (1.19) et (1.20) et respectant les 3 propriétés P_1, P_2 et P_3.

$$V_i(x,y,z) = \sum_{j=1}^{n_i} V_i^j(x,y,z) \tag{1.19}$$

$$V_i^j(x,y,z) = \begin{cases} (\frac{1}{\rho_{ij}^2} - \frac{1}{\rho_{ij0}^2})^2 & \text{pour } |\rho_{ij}| < |\rho_{ij0}| \\ 0 & \text{pour } |\rho_{ij}| > |\rho_{ij0}| \end{cases} \tag{1.20}$$

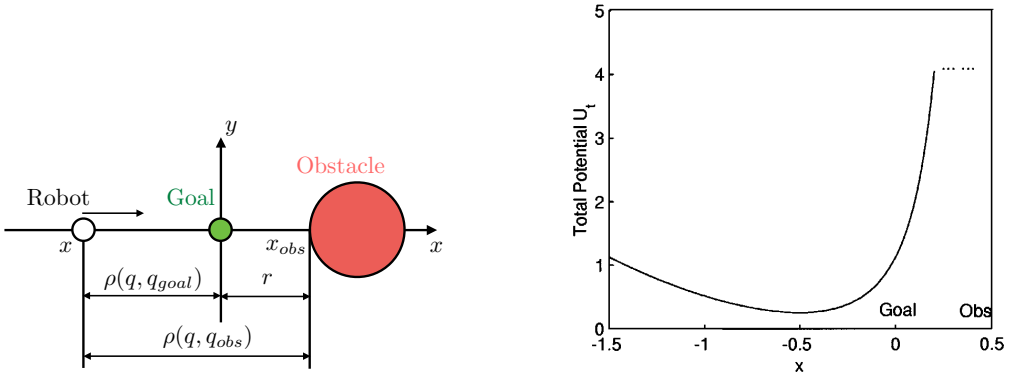
Avec ρ_{ij} la distance entre les 2 points étudiés et ρ_{ij0} le voisinage programmé. Une particularité du potentiel répulsif est alors mise en avant : la notion de voisinage. En effet, le potentiel répulsif est nul (et donc la trajectoire n'est pas déviée) tant que le robot n'est pas rentré dans le voisinage de l'obstacle. Cela permet de modifier la trajectoire du robot uniquement dans la zone considérée comme utile.

— P_1 : $V_i^j(x,y,z)$ est une fonction à gradient continue sur l'espace considéré,

— P_2 : $V_i^j(x,y,z)$ est définie non-négative,

— P_3 : lorsque ρ_{ij} tend vers zéro le potentiel $V_i^j(x,y,z)$ tend vers l'infini.

Cette méthode a été largement utilisée et donne un évitement d'obstacles assuré via la limite de l'effort répulsif calculée. Il faut tout de même noter que selon la première modélisation il n'est pas aisé de rejoindre un point GOAL dans le voisinage. Cette problématique notée GNRON (*Goals NonReachable with Obstacles Nearby*) est illustrée par un cas simple en une dimension représentée en figure 1.46 (où $\mathbf{q} = [x\ y]^T$) [Ge and Cui, 2000].



(a) Positionnement de l'étude

(b) Fonction potentielle obtenue

Figure 1.46 – Mise en avant du problème traité [Ge and Cui, 2000]

Dans ce cas particulier, la localisation du point GOAL dans le voisinage de l'obstacle considéré empêche d'atteindre l'objectif initialement programmé. Pour ce faire, la formulation de l'effort répulsif généré par l'obstacle considéré est modifiée en utilisant l'effort de rappel le plus courant [Koren and Borenstein, 1991]. L'équation générale (1.21) donne la valeur du potentiel total exercé sur le point étudié.

$$U_{total}(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}) \tag{1.21}$$

Avec l'expression du potentiel de rappel donné en équation (1.22) et du potentiel répulsif explicité en équation (1.23).

$$U_{att}(\mathbf{q}) = \frac{1}{2}.\xi.\rho^m(\mathbf{q},\mathbf{q}_{goal}) \tag{1.22}$$

$$U_{rep}(\mathbf{q}) = \begin{cases} \frac{1}{2}.\eta.(\frac{1}{\rho(\mathbf{q},\mathbf{q}_{obs})} - \frac{1}{\rho_0})^2.\rho^n(\mathbf{q},\mathbf{q}_{goal}) & \text{si } \rho(\mathbf{q},\mathbf{q}_{obs}) \leqslant \rho_0 \\ 0 & \text{sinon} \end{cases} \tag{1.23}$$

Avec :

— ξ, m, η et n : des constantes à choisir,

— ρ : la distance étudiée,

— ρ_0 : la valeur du voisinage utilisée.

Cela permet de déterminer les efforts émanants de ces potentiels attractif et répulsif explicités en équations (1.24) et (1.25).

$$\mathcal{F}_{att}(\mathbf{q}) = -\nabla U_{att}(\mathbf{q}) = \frac{1}{2}.\xi.m.\rho^{m-1}(\mathbf{q},\mathbf{q}_{goal}) \tag{1.24}$$

$$\mathcal{F}_{rep}(\mathbf{q}) = -\nabla U_{rep}(\mathbf{q}) = \begin{cases} \mathcal{F}_{\boldsymbol{rep1}}.\mathbf{n}_{OR} + \mathcal{F}_{\boldsymbol{rep2}}.\mathbf{n}_{RG} & \text{si } \rho(\mathbf{q},\mathbf{q}_obs) \leqslant \rho_0 \\ 0 & \text{sinon} \end{cases} \tag{1.25}$$

La décomposition de l'effort répulsif détaillée ci-dessous est illustrée en figure 1.47 :

— $\mathbf{n}_{OR} = \nabla\rho(\mathbf{q},\mathbf{q}_{obs})$,

— $\mathbf{n}_{RG} = \nabla\rho(\mathbf{q},\mathbf{q}_{goal})$,

— $\mathcal{F}_{\boldsymbol{rep1}} = \eta.(\frac{1}{\rho(\mathbf{q},\mathbf{q}_{obs})} - \frac{1}{\rho_0}).\frac{\rho^n(\mathbf{q},\mathbf{q}_{goal})}{\rho^2(\mathbf{q},\mathbf{q}_{obs})}$,

— $\mathcal{F}_{\boldsymbol{rep2}} = \frac{n}{2}\eta.(\frac{1}{\rho(\mathbf{q},\mathbf{q}_{obs})} - \frac{1}{\rho_0})^2.\rho^{n-1}(\mathbf{q},\mathbf{q}_{goal})$.

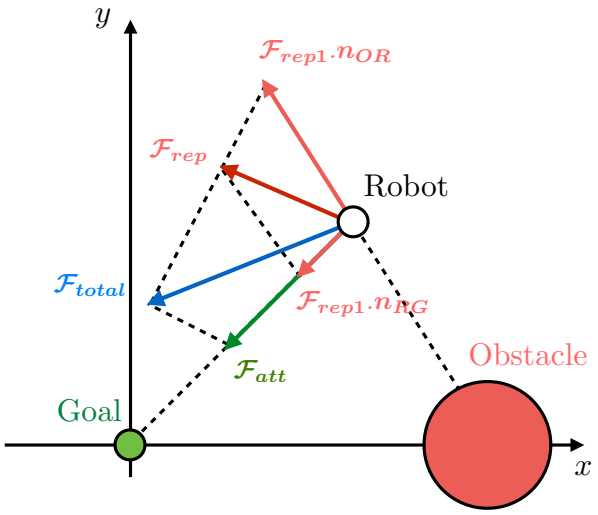
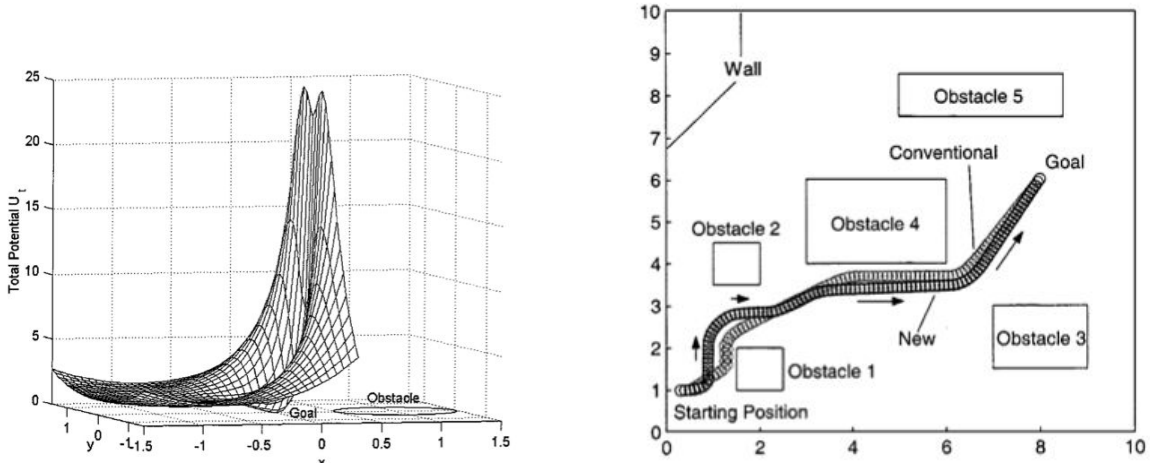


Figure 1.47 – Effort répulsif obtenu après dérivation de la nouvelle fonction définissant le potentiel [Ge and Cui, 2000]

Le résultat final obtenu sur la problématique illustrée en figure 1.47 est représenté en figure 1.48 [Ge and Cui, 2000]. Le potentiel répulsif est nul au point GOAL et tend vers l'infini à l'approche de l'obstacle. Une seconde simulation sur une trajectoire complexe avec évitement d'obstacles est également présentée en figure 1.48.



(a) Représentation du potentiel obtenu

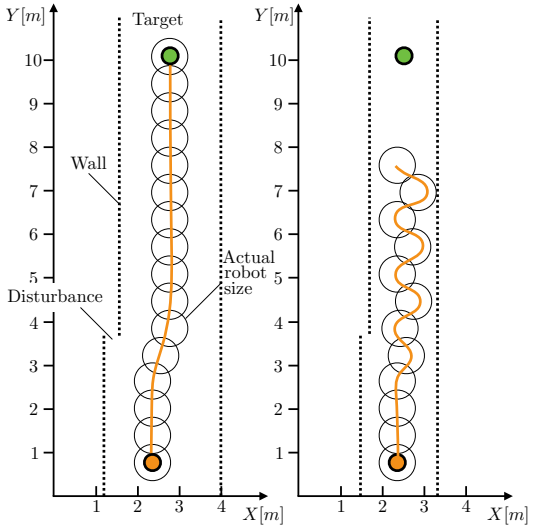
(b) Application à une trajectoire complète

Figure 1.48 – Résultats du problème traité [Ge and Cui, 2000]

Dans le domaine de la robotique mobile, certains auteurs s'intéressent aux positions critiques [Borenstein and Koren, 1989] [Koren and Borenstein, 1991]. En effet, bien que le problème d'oscillation après franchissement d'obstacles ait peu d'influence dans le domaine de la robotique mobile, il était important pour eux de s'assurer de la capabilité de leur programmation 2D ; les résultats de leurs travaux sont donnés en figure 1.49(a) avec diverses affectations de valeurs aux constantes utilisées donnant différentes réponses du mouvement.

D'autres méthodes de programmation utilisant des potentiels existent tels que l'assemblage de régions obstacles [Hwang and Ahuja, 1992], un travail avec utilisation de multi robots [Song and Kumar, 2002], une vérification de la trajectoire obtenue [Ulrich and Borenstein, 2000] ou encore [Arrasa et al., 2002] [Mbede et al., 2000] [Volpe and Khosla, 1990] [Warren, 1989].

Le dernier point développé de l'état de l'art concernant la robotique mobile est la comparaison entre les différentes méthodes d'utilisation dans le domaine de la robotique mobile et la correction par utilisation de potentiels répulsifs. Cette comparaison est réalisée dans le cas d'un déplacement plan [Abiyev et al., 2010]. Cela permet de démontrer que la méthode d'Oussama Khatib [Khatib, 1986] est la plus efficace et vient l'améliorer dans le but d'avoir une trajectoire encore plus continue, afin d'éviter les phases de changement d'accélérations trop brutales. Les résultats de ces travaux sont illustrés en figure 1.49(b).



(a) Gestion de collisions en position critique [Koren and Borenstein, 1991]



(b) Comparaison des méthodes d'évitement de collisions [Abiyev et al., 2010]

Figure 1.49

D'autres auteurs utilisent les potentiels afin de générer un évitement d'obstacles et apportent une amélioration à l'existant en proposant, par exemple, un réglage des constantes selon la trajectoire souhaitée [Bentes and Saotome, 2012], ou l'intégration d'un aspect dynamique du mouvement appliqué pour, notamment, une prise en compte d'obstacles mobiles [Brock and Khatib, 1998] [Ge and Cui, 2002] [Khatib and Maitre, 1978].

D'autres travaux portant sur la robotique mobile vont dans une autre direction et utilisent le calcul géométrique (habituellement utilisé dans le domaine de l'usinage). Ces travaux ne seront pas développés dans cette thèse, car ils s'éloignent du sujet étudié, mais il semble important d'en souligner l'existence dans le but de montrer que les roboticiens ont également essayé la méthode géométrique pour pouvoir la comparer avec l'existant dans leur commu-

nauté [Faverjon and Tournassoud, 1987] [Lozano-Perez, 1980]. Un cas pratique a également été développé pour illustrer cet aspect théorique [Rubrecht, 2011].

3.3 Application de l'approche potentielle à l'usinage

La notion de potentiels et de champs répulsifs a déjà été proposée dans le domaine de l'usinage 5 axes en cas statique pour un évitement de collisions globales avec des configurations spéciales. En effet, une version simplifiée, illustrée en figure 1.50, de la formulation d'efforts répulsifs développée par [Khatib, 1986] a été utilisée dans le domaine de l'usinage tel que décrite en équation (1.26) [Cho et al., 1997].

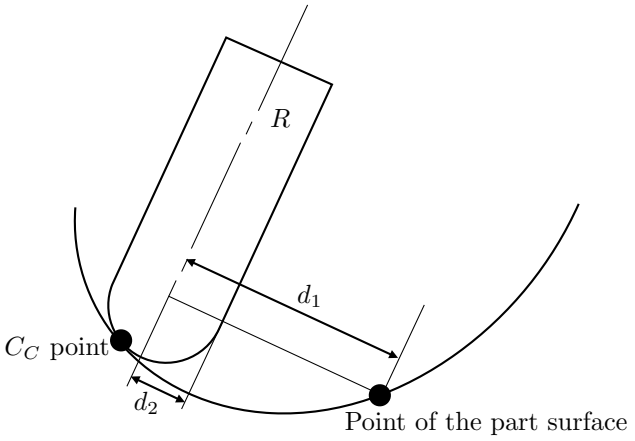


Figure 1.50 – Positionnement du problème [Cho et al., 1997]

$$U = \sum_{i=1}^{m} \int_{S_i} \sum_{j=1}^{n} \int S_j \frac{r_i.r_j}{k_i}.f(d_1).dS_i.dS_j \tag{1.26}$$

Avec la fonction représentant l'effort donné en équations (1.27) et (1.28) :

$$f(d_1) = \begin{cases} \frac{1}{d_1} \text{ si } d_1 < d_2 \\ 0 \text{ si } d_1 > d_2 \end{cases} \tag{1.27}$$

et :

$$d_2 = \begin{cases} R - r + \sqrt{2.r.y - y^2} & \text{si } 0 < y < r \\ R & \text{si } y \geqslant r \end{cases} \tag{1.28}$$

Dans le cadre de cette étude, la notion de voisinage n'est pas utilisée. En effet, l'effort répulsif exprimé en $\frac{1}{d}$ est initialement présente dans l'espace de travail complet. Cette méthode permet de supprimer un paramètre, mais impose un déplacement (aussi petit soit-il) même lorsque l'outil est éloigné de l'obstacle étudié. C'est pour cela que dans cet article, l'effort répulsif est activé lorsque le corps outil pénètre dans la zone de collision et tend vers l'infini lors de l'approche de l'axe outil. Cette méthode impose une discontinuité (qui reste un paramètre acceptable ou non selon

l'utilisation et le niveau de discontinuité obtenu) au niveau de d_2 non présente avec la modélisation classique [Khatib, 1986]. La figure 1.51 illustre la génération de l'effort répulsif.

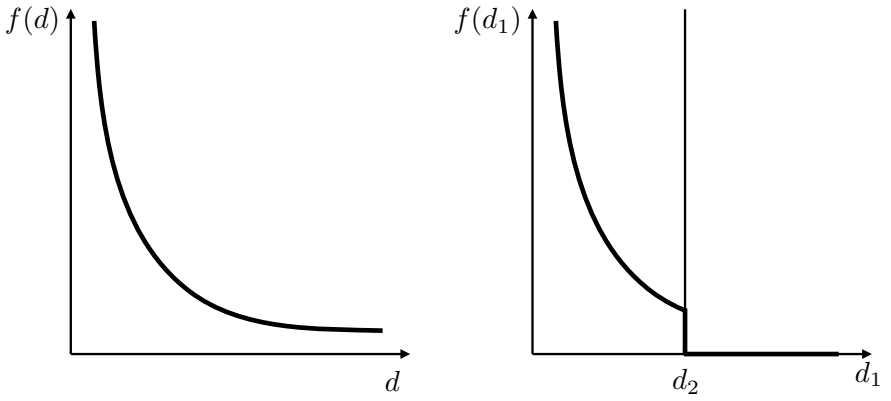
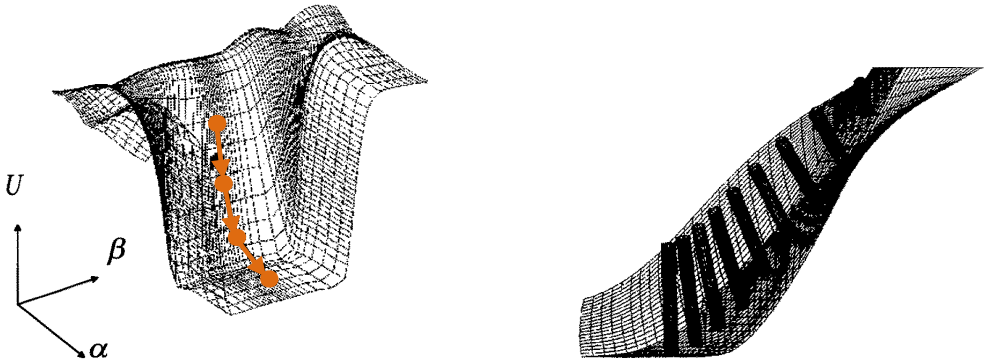


Figure 1.51 – Calcul de l'effort répulsif utilisé [Cho et al., 1997]

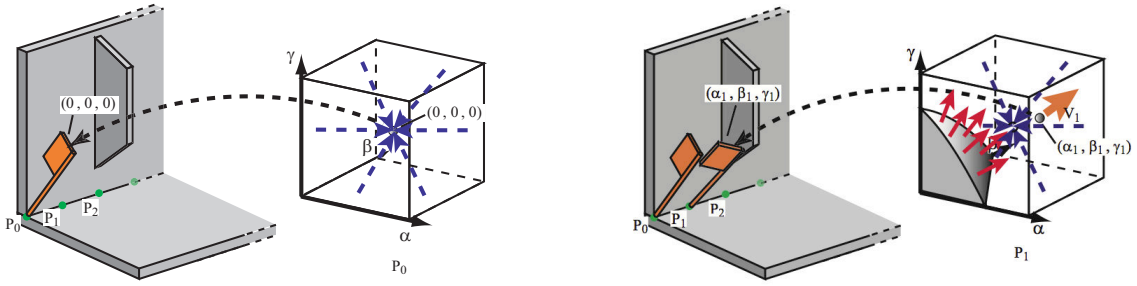
L'outil complet est alors considéré comme étant le voisinage activant l'effort répulsif permettant une réorientation de l'axe outil. Quand la collision est détectée, après plusieurs itérations illustrées en figure 1.52, l'axe outil est réorienté afin d'atteindre une zone de non-collision et la solution obtenue est illustrée en figure 1.52.



(a) (b)

Figure 1.52 – Solution obtenue dans le cas d'un *impeller* [Cho et al., 1997]

D'autres travaux modélisent un ensemble d'efforts répulsifs émanant d'une surface considérée comme obstacle afin d'assurer une trajectoire hors collision du bras anthropomorphe concerné [Asakawa and Kanjo, 2013]. De ce fait, un ensemble d'efforts répulsifs et de rappels (non décrits formellement dans l'article étudié) vont permettre au bras robotisé de modifier la trajectoire de sa partie non fixée et ainsi d'éviter une collision. Tant que le robot étudié est suffisamment éloigné, il n'est soumis qu'à l'effort de rappel, et dès qu'il rentre dans la zone de voisinage, il est alors soumis à un ensemble d'efforts répulsifs (voir figure 1.53).



(a) Zone hors voisinage (b) Zone dans le voisinage

Figure 1.53 – Représentation des deux situations [Asakawa and Kanjo, 2013]

Après application sur la trajectoire complète, la trajectoire du corps du bras anthropomorphe concernée est représentée en figure 1.54.

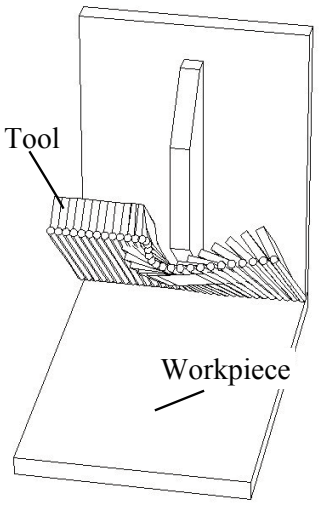


Figure 1.54 – Trajectoire prenant en compte l'obstacle [Asakawa and Kanjo, 2013]

La dernière méthode permettant d'effectuer des changements d'orientation en temps réel est l'utilisation de systèmes haptiques dans le domaine de l'usinage 5 axes [Balasubramaniam et al., 2002] [Chen et al., 2005b] [Liua et al., 2004] [Zhu and Lee, 2004b]. Cette méthode de gestion de collision dans le domaine de l'usinage est illustrée en figure 1.55 et permet, à l'aide d'un retour d'effort de modifier en temps réel la position et l'orientation de l'axe outil étudié.

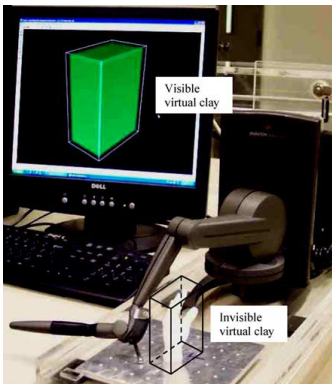
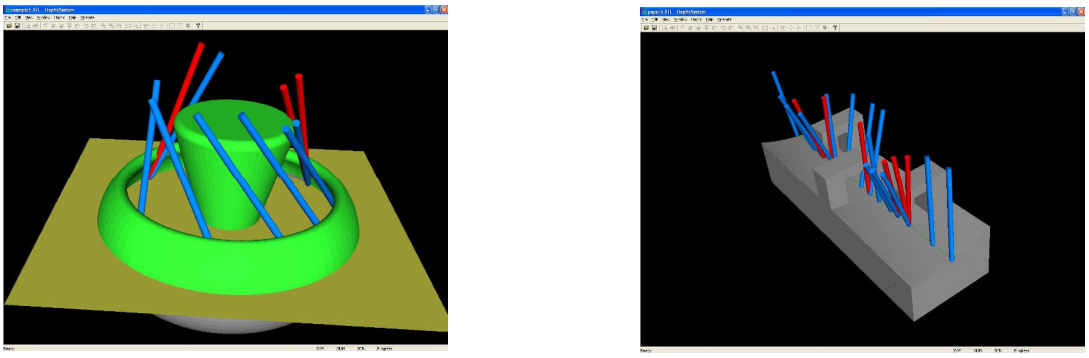


Figure 1.55 – Utilisation d'un système haptique [Chen et al., 2005b]

De plus, une automatisation afin d'éviter les réorientations manuelles comportant plus de chances d'introduire de fortes discontinuités ou bien amenant dans une nouvelle zone de collision peut être utilisée [Chen and Tang, 2006]. La figure 1.56 donne deux simulations réalisées avec la méthode manuelle et la méthode semi-automatique.



(a) Manipulation manuelle

(b) Manipulation semi-automatique

Figure 1.56 – Représentation des deux situations possibles [Chen and Tang, 2006]

4 Synthèse et problématique

4.1 Synthèse de l'état de l'art

Les méthodes dites «géométriques» issues de la littérature portant sur la gestion des interférences globales ont été présentées en partitionnant la démarche en trois phases : la modélisation de l'environnement, l'évaluation des collisions et la correction de l'orientation de l'axe de l'outil.

Plusieurs méthodes ont été abordées parmi lesquelles la méthode des enveloppes convexes qui met l'accent sur la modélisation de l'environnement pour faciliter la détection ultérieure des collisions entre solides. La méthode consiste à générer des polyèdres qui forment une enveloppe

convexe autour des surfaces concernées. Différentes techniques sont disponibles, chacune permettant d'envelopper la surface plus ou moins précisément. Un compromis doit alors être effectué entre précision et complexité des enveloppes générées et donc des calculs de détections de collisions ultérieurs. En effet, une enveloppe trop simpliste, c'est à dire trop éloignée des frontières réelles de la surface conduirait à ne pas considérer nombre de situations hors collisions. Dans le cadre de l'usinage de surfaces complexes, cette approche générerait probablement des polyèdres avec de très nombreuses facettes pour envelopper au mieux la scène d'usinage.

La seconde méthode permettant de résoudre la problématique de détection de collision est l'utilisation de sphères de Gauss ou analyse de la visibilité. L'analyse se décompose en plusieurs étapes en fonction de la richesse de l'information intégrée au calcul. Cela permet de choisir jusqu'où la prise en compte de collision est réalisée. L'utilisateur peut sélectionner, uniquement une vision locale du problème, la prise en compte de la pièce ou bien l'environnement machine au complet. Si d'un point de vue théorique, l'approche «sphère de Gauss» est élégante, sa mise en œuvre passe par un calcul numérique pour explorer tout le domaine «C-Space» attaché à chaque positon d'usinage. Cela génère une explosion combinatoire et donc des temps de calcul potentiellement longs si l'utilisateur travaille dans le cadre d'un environnement très complexe avec une discrétisation très fine des orientations.

La dernière méthode étudiée est l'exploitation de décalages ou surfaces *Offset* des surfaces obstacles ou de leurs courbes frontière en fonction du rayon de l'outil utilisé. Les surfaces *Offsets* sont alors utilisées pour déterminer les orientations maximales de l'axe de l'outil. Cette approche se prête bien à la problématique du passage de l'usinage 3 axes à l'usinage 5 axes pour laquelle l'utilisation d'*Offsets* des bords supérieurs (surfaces tuyaux) permet de réorienter l'outil pour minimiser sa longueur par exemple. Cet aspect purement géométrique simplifie fortement les calculs en ce qui concerne la surface étudiée, par contre, elle peut présenter plusieurs risques. Tout d'abord, il est possible de relever des problèmes de raccordement entre les ces dites surfaces, il n'existe pas forcément de surface *Offset* de même format de description que la surface initiale et enfin, une courbure trop importante peur générer la présence de boucle d'auto intersection.

Toutes ces méthodes d'évitement de collision ne permettent pas toujours de prendre en compte la trajectoire dans sa globalité pour assurer la correction. Chaque étape de détection/correction est itérative afin d'explorer toutes les orientations possibles et trouver les limites. En se référant au diagramme d'activité de la simulation de fabrication fournie en figure 1.12, il est alors aisé de comprendre la problématique de présence de cette phase de correction pouvant générer des temps de calcul importants.

De plus, bien que les collisions soient effectivement évitées il s'avère primordial, d'un point de vue UGV, d'optimiser globalement la trajectoire corrigée obtenue. Pour ce faire, une étude de différentes méthodes d'optimisation après évitement d'obstacles permet de comprendre comment rendre la trajectoire obtenue post-correction plus fluide. Les méthodes étudiées permettent de lisser les trajectoires en interpolant des orientations imposées le long des trajectoires ou des

domaines d'orientation admissibles ou «C-Space». Les méthodes par zone laissent davantage de degrés de liberté, mais nécessitent la mise en œuvre de méthodes d'optimisation et la définition de fonctions objectifs. Ces fonctionnelles sont définies par des critères de «fluidité» géométriques ou cinématiques, dans l'espace pièce ou dans l'espace machine. L'inconvénient de ces approches réside principalement dans la phase d'optimisation à l'issue incertaine (convergence, minimum local, etc.) dans un contexte d'usinage 5 axes.

Toutes ces méthodes nécessitent tout de même une forte expertise de l'utilisateur du logiciel de CFAO afin de faire le choix d'optimisation le plus adapté à la situation du problème étudié. Cela apporte donc une nouvelle complexité de programmation en usinage 5 axes.

Dans un second temps, des travaux de gestion des collisions globales à l'aide de potentiels, principalement utilisés dans le domaine de la robotique mobile sont présentés. La particularité de ces méthodes est l'utilisation d'un effort répulsif issu d'une zone considérée comme obstacle. Cela permet de ne plus avoir à travailler sur la partie «test de collision» étant donné que cet effort tend vers l'infini à l'approche de l'obstacle considéré. De ce fait la boucle représentée en rouge au niveau du diagramme d'activité 1.12 n'est plus prise en compte étant donné que le test de collision n'existe pas.

Dans ce type d'approche, le positionnement du robot au cours de l'évitement est indifférent à l'opérateur tant que ce dernier ne rencontre pas un nouvel obstacle. Ainsi l'aspect dynamique de l'étude n'a pas été pris en compte dans la plupart de ces travaux. Cependant, les exigences requises par l'usinage 5 axes UGV sont plus sévères que celle de la robotique mobile. La réorientation de l'axe outil influe sur la vitesse de coupe effective dans le cas de l'utilisation d'outils hémisphériques, sollicite les axes machines et crée potentiellement des marques sur la surface usinée.

4.2 Problématique

Les besoins extraits de l'analyse partielle de l'existant concernant l'évitement d'obstacles dans le cadre de l'usinage 5 axes peuvent se décomposer en trois exigences principales :

— garantir l'absence de collisions,

— générer des trajectoires fluides d'un point de vue UGV,

— proposer une programmation simple des trajectoires.

Pour cela une approche basée sur les méthodes de simulation numérique de type pseudo-mécanique est proposée. Pour ce faire, les travaux développés au cours de cette thèse se basent sur ce qui est appelé le *Physically Based Modeling* classiquement utilisé, notamment dans la création de films d'animation et de jeux vidéo [Pixar, 2001]. L'outil est considéré comme un solide rigide se déplaçant dans un espace $3D$ sur lequel agissent des actions répulsives et attractives. La résolution des équations du mouvement issues de la mécanique entre l'outil et la pièce permet d'assurer la continuité de la correction tout au long de la trajectoire sans phase d'optimisation. Afin de garantir

l'absence de collisions, des efforts générés par des potentiels répulsifs à l'approche des obstacles seront intégrés à la simulation. Une attention particulière sera apportée à la réduction du nombre de paramètres de simulation afin de simplifier la programmation dans une perspective d'industrialisation du logiciel.

La problématique peut donc se formuler comme suit :

Comment modéliser d'un point de vue mécanique le mouvement virtuel de l'outil par rapport à la pièce s'appuyant sur des potentiels répulsifs afin d'assurer l'évitement de collisions le long d'une trajectoire en fraisage 5 axes UGV ?

Une fois, cette problématique énoncée, il est important de comprendre comment définir l'équation du mouvement, une équation différentielle du second ordre avec second membre, afin de répondre au problème énoncé précédemment. Dans le but de modéliser le second membre de l'équation différentielle qui sera utilisée par la suite, l'étude de la modélisation des efforts répulsifs permettant d'éviter des obstacles élémentaires est alors effectuée. Afin de respecter au mieux l'orientation programmée souhaitée par l'utilisateur, un effort de rappel de l'outil sera également intégré. Bien entendu, cette modélisation nécessite le réglage de plusieurs constantes permettant un retour en orientation programmée après franchissement d'obstacles rapide, mais non oscillant.

La seconde partie de ces travaux sera alors basée sur la mise en place d'une programmation théorique pour des cas simples avec une mise en parallèle de l'aspect physique et de l'usinage. L'outil sera modélisé par son axe passant par deux points limites. L'obstacle considéré sera ponctuel émettant l'effort répulsif programmé. La trajectoire à suivre sera linéaire dans le but de ne pas superposer les problèmes pour les traiter le plus indépendamment possible.

Après avoir fixé les constantes à l'aide de l'étude théorique présentée dans le second chapitre, les travaux porteront sur une multiplication des points obstacles afin de représenter une surface obstacle complexe à l'aide d'un maillage. Là encore, cette partie va permettre de fixer plusieurs paramètres selon le type de comportement souhaité pour simplifier le réglage par l'utilisateur. Enfin, une simulation sur pièces réelles de type *impeller* ou coin «5 axes» sera réalisée pour valider l'intérêt de la méthode et des performances.

Il sera alors important de souligner l'évolution du temps de calcul en fonction des modélisations choisies pour l'outil et la pièce. Effectivement, lors de l'analyse la plus complète d'une trajectoire donnée, il peut s'avérer indispensable d'utiliser une solution alternative dans le but de minimiser les temps de calcul tout en estimant les approximations par rapport à la méthode exacte. Le principe de partitionnement de l'espace par voxelisation sera alors développé afin de répondre à ce besoin.

Chapitre 2

Développement du modèle

Ce second chapitre permet de montrer la mise en œuvre détaillée du *Physically Based Modeling* permettant de traduire l'application du PFD au système considéré. Cette étude débute par une vision théorique et une explication de la prise en compte ou non de certains éléments représentant le mouvement du solide étudié. La méthode Runge-Kutta RK4 de résolution d'équation différentielle du second ordre avec second membre est alors présentée. Cela permet de générer, via la définition des actions répulsives et de rappel d'obtenir une trajectoire hors collisions pour l'usinage 5 axes. Cette étude théorique est alors appliquée sur des simulations simples pour fixer un certain nombre de paramètres afin de garantir un évitement d'obstacle tout en assurant à l'outil étudié un retour vers son positionnement programmé.

Sommaire

1 Introduction 53

2 Description du modèle dynamique basé sur l'analogie mécanique 56

2.1 Modèle général 3D du mouvement d'un solide 56

2.2 Méthode de résolution 60

2.3 Modèle à 2 ddl autour des vecteurs **f** et **t** dans le repère local 62

3 Modélisation des actions mécaniques 63

3.1 Modélisation de l'action répulsive 64

3.2 Modélisation de l'action de rappel 68

3.3 Attribution de valeurs fixes aux paramètres étudiés 73

4 Effet des paramètres dynamiques sur la trajectoire obtenue 73

4.1 Étude géométrique de la modification angulaire de l'axe outil 75

4.2 Étude de l'impact de l'orientation de l'axe outil modifiée sur le comportement cinématique et dynamique du système 76

5 Conclusions 78

1 Introduction

Après avoir réalisé l'analyse de l'existant au sujet de l'évitement de collisions globales autant dans le domaine de l'usinage que dans le domaine de la robotique mobile, la seconde partie des travaux présentés dans cette thèse concerne la mise en place d'un évitement de collision en usinage 5 axes. Cette approche est basée sur des critères physiques amenant à la réalisation d'un modèle pseudo mécanique. Le but est alors de permettre au corps de l'outil utilisé d'assurer un évitement des interférences globales, potentiellement présentes à l'approche d'un obstacle. Dans la suite des travaux, la trajectoire du point piloté (le centre outil C_L) est considérée comme connue et fixée. Les travaux ne concernent que la réorientation de l'axe outil.

Cette problématique est illustrée avec un exemple donné en figure 2.1 où il est possible de voir un outil s'inclinant latéralement sous l'effet d'un champ répulsif associé ici à un obstacle ponctuel. L'orientation du vecteur **u** représentant l'axe outil est alors modifiée par rapport au repère (f,n,t) attaché à la trajectoire programmée. Par la suite, une action de rappel devrait permettre à l'outil de regagner son orientation programmée le long de la trajectoire. La position du point centre outil n'est pas modifiée afin d'assurer un usinage cohérent avec la programmation initiale.

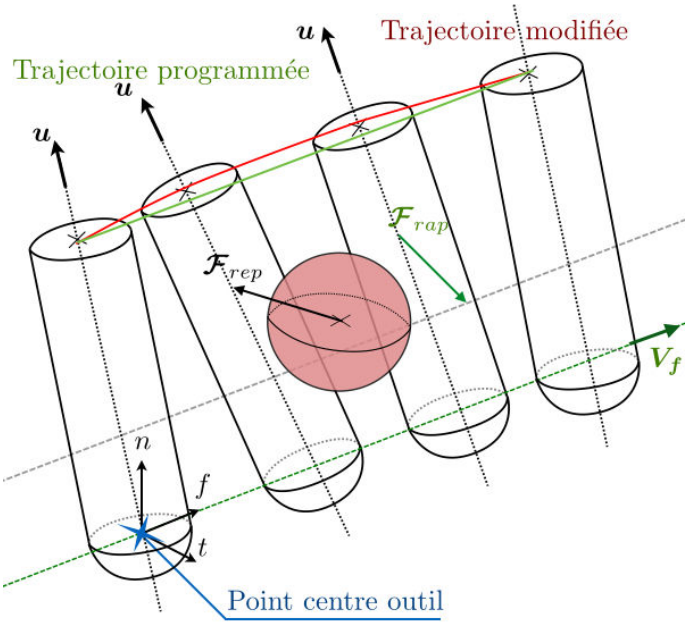


Figure 2.1 – Mise en avant de la problématique

Comme présentée au cours du chapitre précédent, la programmation type d'un logiciel de FAO pour l'évitement de collision est traitée d'un point de vue géométrique. Classiquement l'outil est, dans un premier temps positionné tangent à la surface, puis la construction de la courbe dans le sens de l'usinage permet d'obtenir la trajectoire d'usinage avec un contrôle et si besoin, une correction de l'axe outil obtenu. La méthode de génération directe proposée au cours de cette thèse consiste

quant à elle à modéliser le problème comme celui d'un solide évoluant dans un univers virtuel composé d'éléments de guidage et d'obstacles. La trajectoire de l'outil définie par sa position et son orientation est calculée en résolvant les équations du mouvement issues du Principe Fondamental de la Dynamique. Pour la suite, cette méthode de calcul est nommée **Geo5XField**. Il est essentiel de noter que cette dernière permet d'obtenir une simulation sans itération.

De ce fait le diagramme d'activité représentant la simulation de fabrication donnée en figure 1.12 avec un flux de contrôle concernant la correction deviendrait celui fourni en figure 2.2 sans boucle de contrôle de correction de l'orientation de l'axe outil.

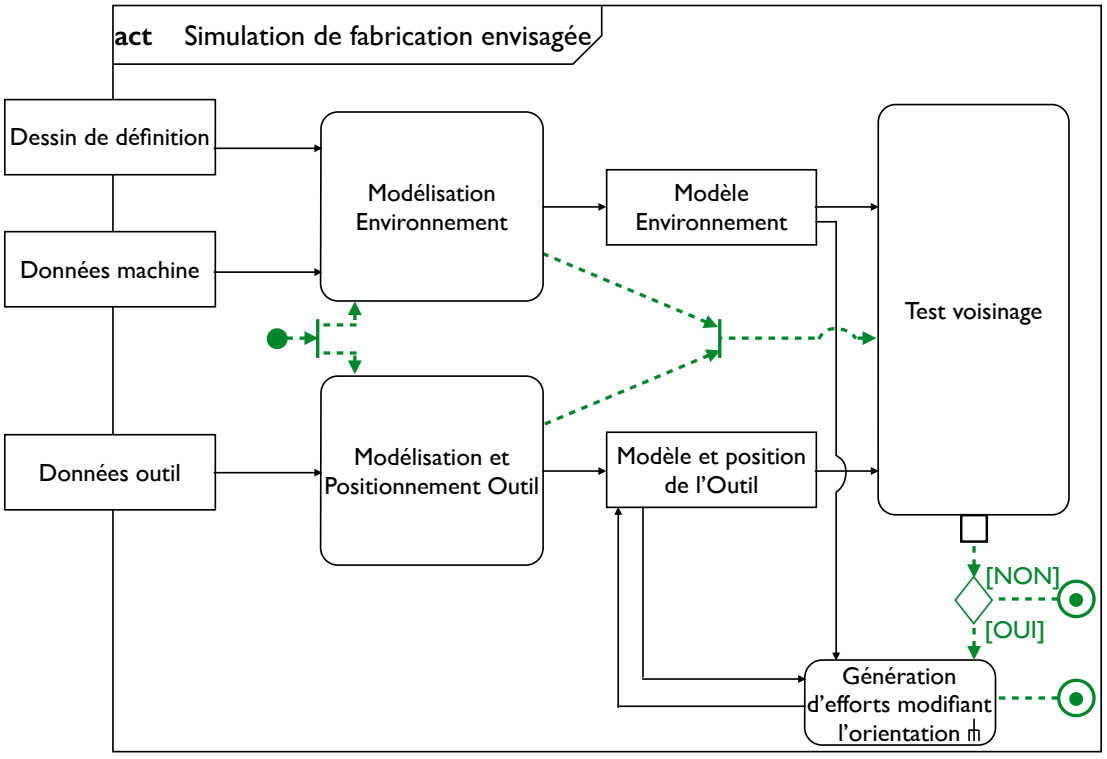


Figure 2.2 – Diagramme d'activité de la simulation de fabrication envisagée

Une des étapes à réaliser est la définition des actions répulsives générées par les obstacles. Ces derniers sont considérés comme des sources émettrices d'efforts répulsifs sur le solide en mouvement et permettent d'assurer l'absence de collision entre l'outil étudié et les surfaces obstacles lors du parcours de la trajectoire programmée. La difficulté principale repose dans la modélisation des obstacles et de l'effort répulsif qui s'exerce sur l'outil. En effet, bien qu'un effort répulsif quelconque, dans le domaine de la dynamique, génère une nouvelle trajectoire continue, ce dernier n'assure pas pour autant l'obtention d'une réorientation fluide d'un point de vue cinématique. En effet, la vitesse de réorientation doit respecter les capacités cinématiques machine disponibles afin de ne pas inclure de marques d'usinage. L'utilisation de fonctions potentielles est exploitée pour obtenir un modèle satisfaisant d'effort répulsif. Celui-ci doit assurer une fluidité de la trajectoire corrigée à l'aide d'une diminution de la rapidité de réorientation ainsi qu'une anticipation de l'arrivée d'un obstacle.

Par la suite, un effort de rappel permettant de respecter au mieux l'orientation initialement programmée est alors configuré. En effet, si aucun effort ne s'oppose à ceux exercés par les obstacles, l'orientation de l'axe outil n'est pas définie. Cependant, la mise en place d'efforts de rappel risque d'apporter de nouvelles problématiques concernant des réorientations incompatibles avec les exigences de l'usinage telles que des oscillations de l'axe outil.

Les exigences que doivent remplir ces efforts répulsifs et de rappel inhérent à l'usinage 5 axes sont données ci-dessous et représentées par le diagramme d'activité 2.3 :

— absence de collision : $\lim_{r \to 0}(||\boldsymbol{\mathcal{F}_{rep}}||) = +\infty$ avec $\boldsymbol{\mathcal{F}_{rep}}$ l'effort répulsif et r la distance entre l'outil et l'obstacle,

— absence d'oscillation : toute réponse oscillante est à éviter afin d'assurer une modification d'orientation lisse,

— réactivité du mouvement de correction de l'axe outil : il faut que le comportement de l'axe outil soit réactif au franchissement de la zone de voisinage de l'obstacle tout en respectant les caractéristiques cinématiques et dynamiques limites.

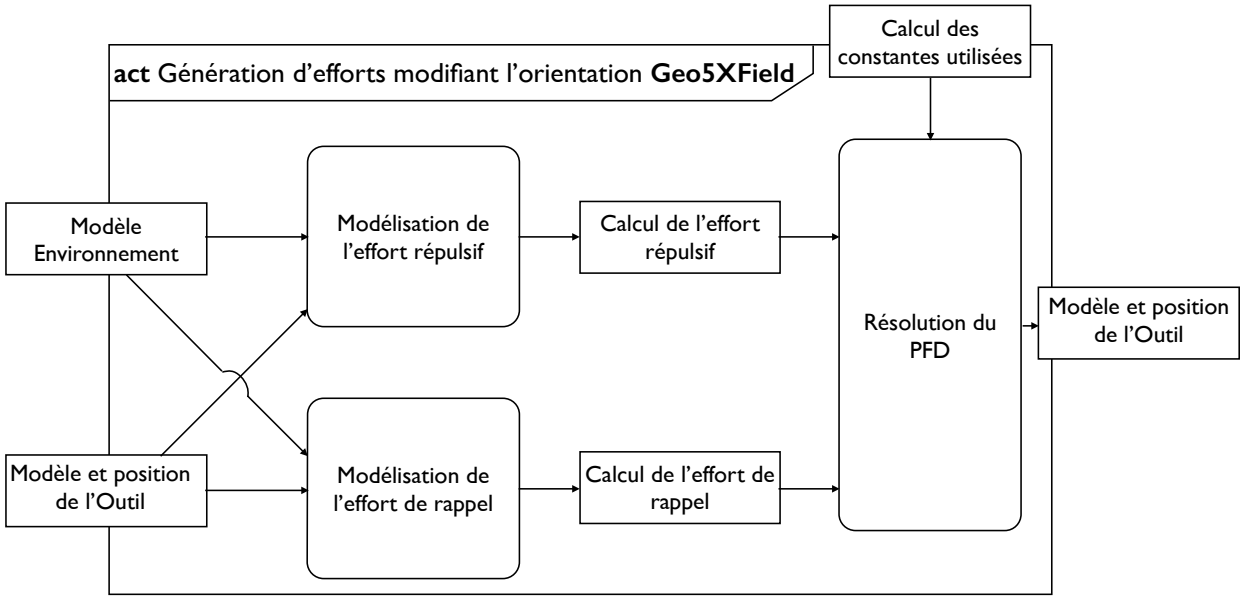


Figure 2.3 – Diagramme d'activité représentant **Geo5XField**

Afin de présenter le choix des différentes conditions permettant de répondre aux exigences citées précédemment, la construction du modèle se décompose en quatre parties :

1) Tout d'abord, un parallélisme entre le point de vue physique de la modélisation utilisée et le point de vue usinage sera mis en place. Pour ce faire, la description des différents repères nécessaires au calcul d'une trajectoire en fraisage 5 axes ainsi que la prise en compte d'une vitesse d'avance constante le long de la trajectoire programmée seront effectuées afin de justifier les hypothèses retenues pour la suite des travaux.

2) Puis la méthode numérique de résolution de l'équation différentielle du second ordre avec second membre est présentée (Runge-Kutta RK4). Cette étape permettra de comprendre l'implémentation effectuée sous Matlab® avec la fonction *ODE*45.

3) Par la suite, l'étude porte sur la modélisation des actions répulsives et des actions de rappel afin de permettre à l'outil d'éviter l'obstacle considéré. Il sera alors important de comprendre leurs effets et les conditions de réalisation du besoin exprimé concernant l'usinage 5 axes.

4) Enfin cette partie se termine par la détermination des paramètres dynamiques permettant d'obtenir, en régime transitoire, la réponse apériodique souhaitée dans le cadre de l'évitement de collision dans le domaine de l'usinage 5 axes.

Pour simplifier l'analyse à ce stade, l'outil est modélisé par deux points C_L et G fixes. Dans ce cas, le point C_L, représentant le point centre outil, suit la trajectoire initialement programmée et donc seul le point G, représentant le centre de gravité de l'outil, est soumis aux actions répulsives et de rappel.

2 Description du modèle dynamique basé sur l'analogie mécanique

Au cours de cette étude, une modélisation dynamique est utilisée dans le but de décrire le déplacement d'un outil. À cette fin, l'écriture théorique utilisée ainsi que les hypothèses considérées dans ces travaux sont énoncées afin d'appliquer l'approche physique pour générer la trajectoire d'usinage.

2.1 Modèle général 3D du mouvement d'un solide

Le Principe Fondamental de la Dynamique conduit à écrire deux équations vectorielles ou six équations scalaires. Ces équations scalaires sont des équations différentielles du second ordre, en général non linéaires, de la forme notée en équation (2.1) où les paramètres q_i sont les i paramètres du mouvement.

$$f(q_i(t),\dot{q}_i(t),\ddot{q}_i(t)) = 0 \tag{2.1}$$

Outre les q_i, ces équations peuvent également contenir des composantes inconnues d'actions mécaniques. Par définition, une équation du mouvement est une équation différentielle du second ordre des paramètres du mouvement ne comportant aucune composante inconnue d'action mécanique. Cette modélisation s'applique donc au traitement de la problématique ; il existe alors un référentiel considéré comme galiléen, noté R dans lequel, pour le solide S au point A quelconque (équation (2.2)) :

$$\left\{\mathcal{D}_{S/R}\right\}_A = \left\{\mathcal{F}_{\bar{S}/S}\right\}_A \tag{2.2}$$

avec $\left\{\mathcal{D}_{S/R}\right\}_A$ le torseur dynamique exprimé au point A et $\left\{\mathcal{F}_{\bar{S}/S}\right\}_A$ le torseur des actions mécaniques sollicitant le système S dans son mouvement par rapport à R exprimé au point A.

Afin de comprendre, dans le cadre de la génération de trajectoires, les repères utilisés, une composition des mouvements entre l'outil et la pièce est développée ci-dessous dans le but de dissocier les aspects étudiés. La composition des mouvements peut être formulée avec deux points de vue différents, d'un point de vue FAO (donné en équation (2.3)) ou d'un point de vue MOCN associé à l'architecture machine (donné en équation (2.4)).

$$\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/R_p) = \mathbf{Mvt}(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}}) + \mathbf{Mvt}(\text{outil}_{\text{strat}}/R_{fnt}) + \mathbf{Mvt}(R_{fnt}/R_p) \tag{2.3}$$

$$\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/R_p) = \mathbf{Mvt}(\text{outil}_{\text{corrigé}}/R_M) + \mathbf{Mvt}(R_M/R_p) \tag{2.4}$$

Avec :

— R_{fnt} : le repère associé à la trajectoire programmée,

— R_p : le repère associé à la pièce usinée,

— R_M : le repère associé à la machine utilisée.

Le passage par le repère machine impose un choix de cette dernière. C'est pour cela que, dans le but de rester le plus généraliste possible, les travaux présentés au cours de cette thèse portent essentiellement sur l'aspect FAO de l'espace pièce.

Par la suite, la composition correspondante est détaillée mouvement par mouvement afin de comprendre les définitions utilisées. Le premier mouvement, noté $\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}})$, représente une modification d'orientation de l'axe outil par rapport à la stratégie programmée. Cette modification traitée d'un point de vue dynamique dans le but d'obtenir une trajectoire fluide permet de déterminer l'orientation de l'axe outil corrigé noté $\mathbf{u}_{\text{corrigé}}$ visible en figure 2.4.

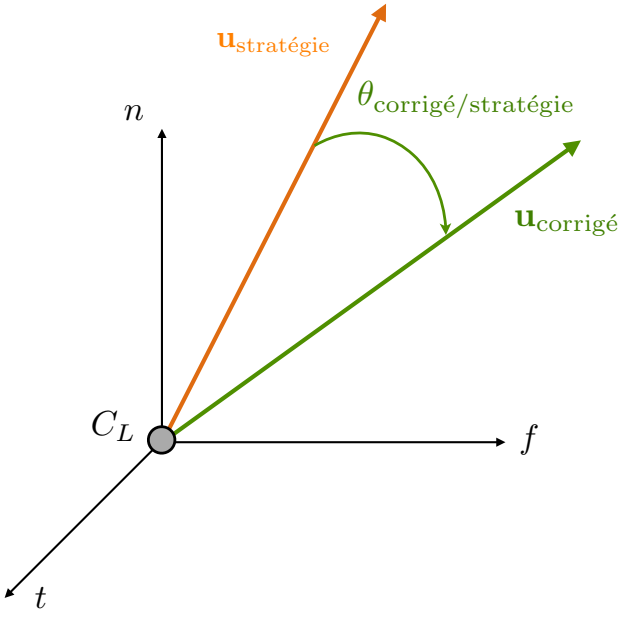


Figure 2.4 – $\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}})$

En ce qui concerne le mouvement de l'outil, dit de stratégie, noté **Mvt**(outil$_{\text{strat}}/R_{fnt}$), il représente l'orientation programmée de l'axe outil par rapport à la base $\mathcal{B}_{fnt}$. Le référentiel permettant de situer la partie active de l'outil le long de la trajectoire programmée est celui de Frenet noté R_{fnt} avec les vecteurs le caractérisant rappelés ci-dessous :

— **f** est le vecteur tangent à la courbe suivie par le point piloté,

— **n** est le vecteur normal à la surface,

— **t** est le vecteur transverse à la trajectoire et tangent à la surface tel que : $\mathbf{t} = \mathbf{f} \wedge \mathbf{n}$.

Le mouvement étudié peut être variable, en fonction de la stratégie d'usinage retenue, comme dans le cas illustré en figure 2.5 où une anticipation d'inclinaison d'outil est programmée. Dans ces travaux, l'angle permettant d'orienter le vecteur $\mathbf{u}_{\text{stratégie}}$ reste constant tout le long de la trajectoire programmée, mais pourrait tout aussi bien être variable.

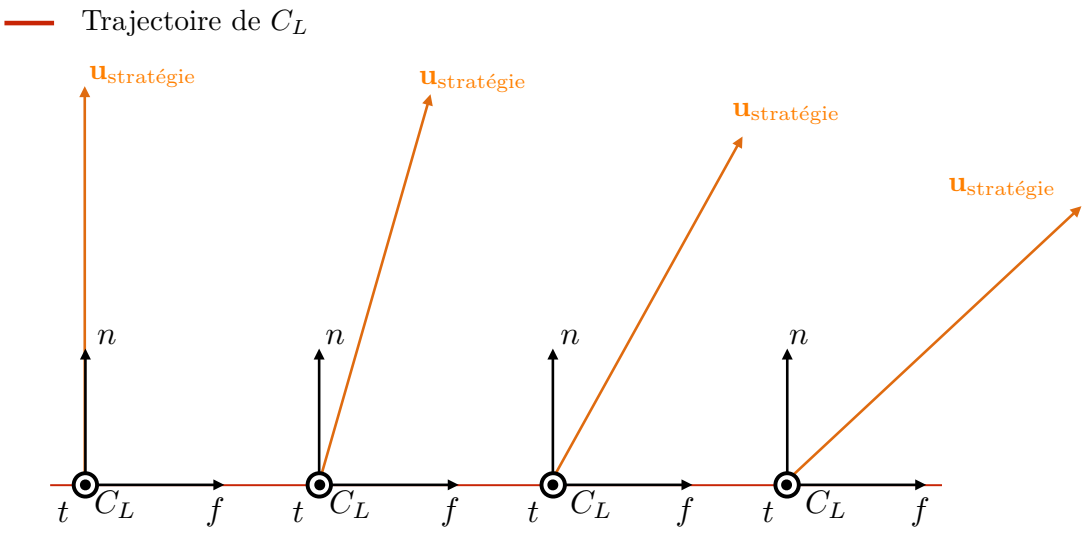


Figure 2.5 – **Mvt**(outil$_{\text{strat}}/R_{fnt}$)

La nouvelle hypothèse utilisée au cours de cette thèse est de considérer que l'orientation de stratégie de l'axe outil $\mathbf{u}_{\text{stratégie}}$ est confondue avec le vecteur **n** associé au repère de Frenet (ce qui imposera par la suite $\theta_{fprog} = \theta_{tprog} = 0°$).

Enfin, pour ce qui est de l'étude entre le mouvement du repère local et le repère pièce, noté **Mvt**(R_{fnt}/R_p), il peut être modélisé comme illustré en figure 2.6. Celui-ci n'est défini que via la géométrie de la surface, la trajectoire choisie et la vitesse de parcours de l'outil le long de cette dernière.

Il est important de rappeler que dans le cadre de cette étude, la modélisation retenue est réalisée au niveau du point C_L de l'outil. En effet, en usinage 5 axes, l'orientation de l'axe outil peut être exprimé dans le système de coordonnées locales suivant (**f**,**n**,**t**) présenté ci-dessus.

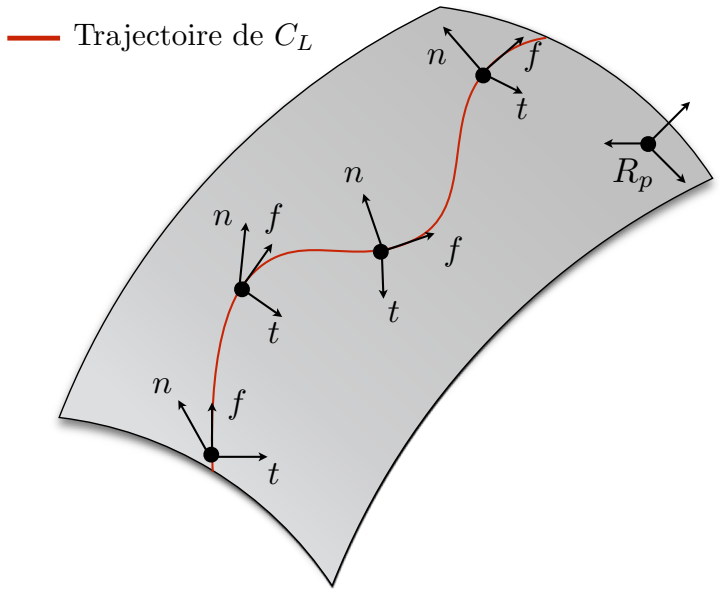


Figure 2.6 – **Mvt**(R_{fnt}/R_p)

Il faut souligner que, lors des simulations présentées par la suite, ces dernières sont programmées avec une valeur de vitesse d'avance du point C_L constante dans le repère de travail R_p et toujours colinéaire à **f**, comme noté en équation (2.5) où le paramètre s représentant l'abscisse curviligne est indépendant de tout repère d'expression, car intrinsèque à la trajectoire (en notant s_0 la valeur de s en $t = 0$).

$$||\dot{\boldsymbol{x}}_{(C_L \in S/R_p)}|| = v_0 \rightarrow s(t) = v_0.t + s_0 \tag{2.5}$$

Un dernier point à soulever concerne la relation entre le temps de simulation et la position programmée. L'équation (2.6) résume cette description en rapport avec la vitesse d'avance énoncée préalablement. Par la suite, une étude par rapport au temps ou une étude par rapport à l'abscisse curviligne sont alors analogues.

$$\frac{d}{dt} = \frac{d}{ds}.\frac{ds}{dt} = \frac{d}{ds}.v_0 \tag{2.6}$$

Le choix effectué concernant la simulation réalisée est de ne prendre en compte l'aspect dynamique uniquement sur le mouvement entre $\mathbf{u}_{\text{corrigé}}$ et $\mathbf{u}_{\text{strat}}$. Tous les autres mouvements de passage entre repères présentés ci-dessus ne seront pas traités de cette manière lors de ces travaux. Il s'agit d'un choix cohérent avec la programmation en usinage 5 axes, car cela signifie que la trajectoire initialement programmée n'est pas touchée par l'aspect dynamique du problème et représente bel et bien la trajectoire initiale. De ce fait, seul l'aspect correction d'orientation est traité dans le cadre de cette étude physique.

L'écriture du Principe Fondamentale de la Dynamique dans le repère pièce s'écrirait alors classiquement comme mentionné en équation (2.7).

$$\begin{cases} M.\ddot{\boldsymbol{x}}_{(C_L \in S/R_p)} = \mathcal{R}\left\{F_{\bar{S}/S}\right\} \\ \mathbb{J}_{C_L}.\ddot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_p)} = \mathcal{M}_{C_L}\left\{F_{\bar{S}/S}\right\} \end{cases} \tag{2.7}$$

Avec :

— M : masse du solide étudié,

— $\boldsymbol{x}_{C_L}$: vecteur représentant la position du point C_L,

— $\boldsymbol{\mathcal{R}}\left\{F_{\bar{S}/S}\right\}$: résultante des actions de l'environnement extérieur $\bar{S}$ sur le solide étudié S,

— $\mathbb{J}_{C_L}$: matrice d'inertie au point C_L,

— $\boldsymbol{\theta}$: vecteur de rotation du solide étudié,

— $\boldsymbol{\mathcal{M}}_{C_L}\left\{F_{\bar{S}/S}\right\}$: moment des actions de l'environnement extérieur $\bar{S}$ sur le solide étudié S au niveau du point C_L.

À l'aide de toutes les hypothèses énoncées précédemment, il paraît alors plus cohérent d'exprimer le PFD dans le repère R_{fnt}, où uniquement la modification d'orientation de l'axe outil est prise en compte et donc seule l'équation des moments (2.8) est utilisée.

$$\mathbb{J}_{C_L}.\ddot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} = \boldsymbol{\mathcal{M}}_{C_L}\left\{F_{\bar{S}/S}\right\} \tag{2.8}$$

Afin de montrer la correspondance de cet aspect dynamique appliqué sur le mouvement **Mvt**(outil$_{\text{corrigé}}$/outil$_{\text{strat}}$), le vecteur d'état noté **X(t)** écrit au point piloté C_L est donné en équation (2.9). La problématique est alors de montrer que générer la trajectoire revient à déterminer la dérivée du vecteur d'état pour une position sur la trajectoire connue ((2.5)) et donc un **X** connu. Afin de formaliser le problème, la dérivée correspondante à ce vecteur d'état est fournie en équation (2.9).

$$\mathbf{X}(t) = \begin{pmatrix} \boldsymbol{\theta}_{(R_{\text{corrigé}}/R_{fnt})} \\ \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} \end{pmatrix} \qquad \frac{d\mathbf{X}(t)}{dt} = \begin{pmatrix} \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} \\ \ddot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} \end{pmatrix} \tag{2.9}$$

L'application du PFD au système étudié permet alors de relier les effets du milieu extérieur à la dérivée du vecteur d'état présentée ci-dessus à l'aide de l'équation (2.10) traduisant la formulation physique décrite initialement en équation (2.8).

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{\theta}_{(R_{\text{corrigé}}/R_{fnt})} \\ \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})} \\ \mathbb{J}_{C_L}^{-1}.\boldsymbol{\mathcal{M}}_{C_L}\left\{F_{\bar{S}/S}\right\} \end{pmatrix} \tag{2.10}$$

Cette méthode décompose alors l'équation différentielle du second ordre en deux équations dépendantes du premier ordre dont la prochaine sous partie présente le principe de résolution, et l'implémentation Matlab® associée.

2.2 Méthode de résolution

Afin d'effectuer cette simulation, un solveur est utilisé dans le but de résoudre le système d'équations différentielles issues du PFD à l'aide de la méthode de Runge-Kutta notée RK4. La description détaillée venant compléter la formulation partielle présentée par la suite est proposée en annexe 1.

2.2.1 Présentation élémentaire de RK4

Soit une équation différentielle du premier ordre donnée en équation (2.11) :

$$\frac{dy}{dx} = f(x,y) \tag{2.11}$$

La méthode RK4 utilise plusieurs points intermédiaires pour calculer la valeur de y_{i+1} à partir de la valeur y_i. L'ordonnée définitive y_{i+1} du point d'abscisse $x_i + h$ est donnée par les relations (2.12) (2.13) et illustrée en figure 2.7 :

$$y_{i+1} = y_i + \frac{1}{6}.((\frac{dy}{dx})_i + 2.(\frac{dy}{dx})_{iA} + 2.(\frac{dy}{dx})_{iB} + (\frac{dy}{dx})_{iC})).h \tag{2.12}$$

$$y_{i+1} = y_i + \frac{1}{6}.(m_1 + 2.m_2 + 2.m_3 + m_4) \tag{2.13}$$

avec une détermination des constantes m_i fournie ci-dessous

— $m_1 = h.f(x_i,y_i)$,

— $m_2 = h.f(x_i + h/2,y_i + m_1/2)$,

— $m_3 = h.f(x_i + h/2,y_i + m_2/2)$,

— $m_4 = h.f(x_i + h/2,y_i + m_3)$.

La méthode RK4 est une méthode d'ordre 4, ce qui veut dire que l'erreur commise à chaque étape est de l'ordre de h^5, alors que l'erreur totale accumulée est de l'ordre de h^4. Un exemple d'application sur un cas simple est fourni en annexe 1 afin de mieux comprendre l'application numérique de ce type de résolution.

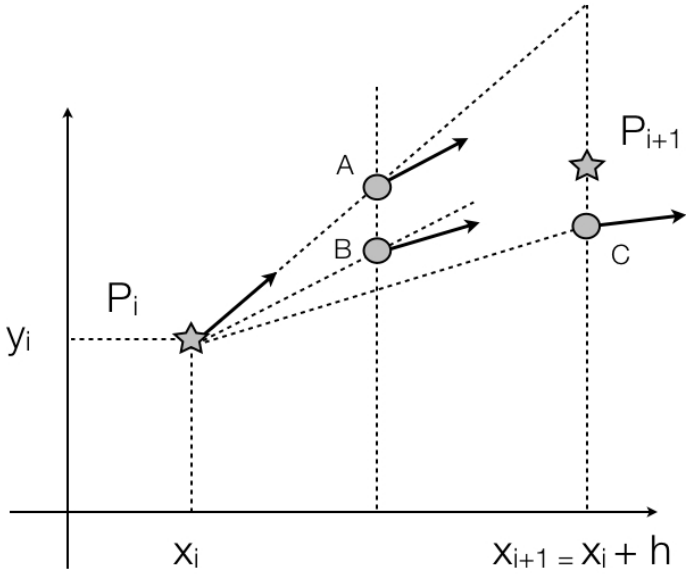


Figure 2.7 – Représentation du principe de résolution RK4

2.2.2 Application de la méthode RK4 à la génération de trajectoires

La méthode de Runge-Kutta RK4 présentée précédemment est mise en œuvre pour résoudre le PFD au point piloté C_L. Dans ce cas, plusieurs hypothèses sont imposées et après application de la méthode théorique présentée précédemment, les résultats sont donnés en équation (2.14)

$$\begin{cases} \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})}(t+dt) = \dot{\boldsymbol{\theta}}_{(R_{\text{corrigé}}/R_{fnt})}(t) + \frac{1}{6}.(\boldsymbol{m}_1 + 2.\boldsymbol{m}_2 + 2.\boldsymbol{m}_3 + \boldsymbol{m}_4) \\ \boldsymbol{\theta}_{(R_{\text{corrigé}}/R_{fnt})}(t+dt) = \boldsymbol{\theta}_{(R_{\text{corrigé}}/R_{fnt})}(t) + \frac{1}{6}.(\boldsymbol{n}_1 + 2.\boldsymbol{n}_2 + 2.\boldsymbol{n}_3 + \boldsymbol{n}_4) \end{cases} \tag{2.14}$$

où les constantes m_i et n_i sont déterminées comme illustré sur un exemple fourni en annexe 1. La fonction Matlab® $ODE45$ utilisant RK4 est implémentée dans la programmation **Geo5XField**.

Après avoir expliqué la méthode de résolution numérique du problème complet en trois dimensions, il est important de remarquer que l'étude peut alors être portée sur les projections de ces dernières dans le but de travailler avec des paramètres scalaires. Une hypothèse de dissociation du comportement de l'axe outil dans deux plans est alors appliquée afin de pouvoir travailler d'un point de vue scalaire et gérer un comportement indépendant dans un plan et dans l'autre. Pour la suite de ces travaux, le vecteur angulaire actuellement noté $\boldsymbol{\theta}_{(R_{\text{corrigé}}/R_{fnt})}$ sera noté $\boldsymbol{\theta}$ pour un souci de lisibilité.

2.3 Modèle à 2 ddl autour des vecteurs f et t dans le repère local

Cette partie montre comment est utilisé ce modèle en fraisage 5 axes en exigeant que le point C_L reste le long des trajectoires définies en FAO. Pour cela l'axe outil est représenté par un segment reliant les deux extrémités C_L et G avec une inertie correspondante. Les deux points étudiés lors des simulations à venir ne sont, pour l'instant, que les deux points extrêmes avec uniquement le point G soumis aux actions extérieures. Ce dernier va rentrer dans le voisinage d'un obstacle ponctuel qui devra le repousser afin de ne pas générer de collision. Lors de ces travaux, les angles θ_f et θ_t permettent d'obtenir le positionnement complet de l'axe outil par rapport au repère local, via la connaissance de la longueur outil ($\boldsymbol{\theta} = (\theta_f; 0; \theta_t)$). Cette paramétrisation est représentée en figure 2.8 avec, pour illustration, un outil hémisphérique.

Il est alors supposé, comme exprimé en équations (2.15) et (2.16), que le problème d'orientation de l'axe outil peut être modélisé comme deux systèmes indépendants dans deux plans différents : (C_L,**n**,**t**) et (C_L,**f**,**n**).

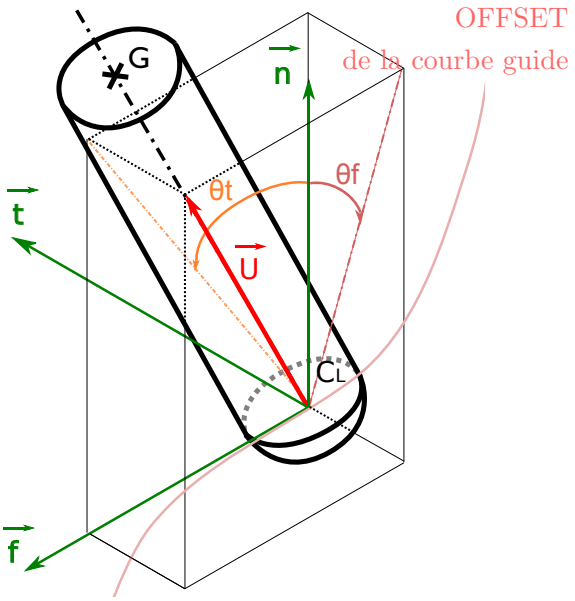


Figure 2.8 – Paramétrisation et angles d'orientation

Cette hypothèse peut être associée à un modèle de deux pendules inverses indépendants associés à deux plans du repère de Frenet. Cette hypothèse peut être justifiée notamment par le choix de l'outil utilisé qui n'impose pas d'ordre spécifique d'utilisation des deux angles en question.

$$J_f.\frac{d^2\theta_f(t)}{dt^2} = \mathcal{M}_{f|C_L}(t) \tag{2.15}$$

$$J_t.\frac{d^2\theta_t(t)}{dt^2} = \mathcal{M}_{t|C_L}(t) \tag{2.16}$$

En effet, pour un outil plus complexe tel que torique ou cylindrique, il serait important de déterminer un ordre de réorientation afin d'éviter des talonnages générant une collision locale au niveau de la surface usinée [Tournier, 2001].

3 Modélisation des actions mécaniques

Une fois que les deux équations des moments découplés sont décrites, le but est de modéliser les éléments constituant le second membre de chacune d'elles. Elles seront décomposées en deux parties principales, les actions générant des moments répulsifs $\boldsymbol{\mathcal{M}_{rep}}$ et permettant à l'axe outil d'éviter l'obstacle ponctuel concerné et les moments de rappel $\boldsymbol{\mathcal{M}_{rap}}$ permettant à l'axe outil de revenir en orientation programmée après franchissement de l'obstacle considéré. Le second membre peut alors s'écrire de la manière suivante.

$$\boldsymbol{\mathcal{M}}_{|C_L}(t) = \boldsymbol{\mathcal{M}}_{rep_{|C_L}}(t) + \boldsymbol{\mathcal{M}}_{rap_{|C_L}}(t) \tag{2.17}$$

Dans un premier temps, les travaux portent sur des obstacles ponctuels et la manière dont l'orientation de l'axe de l'outil est modifiée à l'approche de ce dernier. Afin de comprendre les différents comportements dynamiques du mouvement de l'axe outil, une étude du régime transitoire de l'équation différentielle est traitée. Le régime permanent est exploité par la suite. L'équation complète dont la résolution est étudiée est donnée en équation (2.18). Pour la suite de cette étude, la totalité des moments sera exprimée au point C_L, en conséquence, afin d'alléger l'écriture des équations, les moments ne préciseront plus le point concerné.

$$\mathbb{J}\ddot{\boldsymbol{\theta}}(t) - \boldsymbol{\mathcal{M}}_{\boldsymbol{rap}}(t) - \boldsymbol{\mathcal{M}}_{\boldsymbol{rep}}(t) = \mathbf{0} \tag{2.18}$$

L'expression des différentes composantes commence par la formulation de l'action répulsive permettant à l'outil modélisé d'éviter un obstacle élémentaire.

3.1 Modélisation de l'action répulsive

Durant cette partie de modélisation des actions répulsives, un cas simple est développé. Le point C_L de l'outil suit une trajectoire programmée, ici rectiligne pour simplifier l'exemple. Le long de cette trajectoire, le point G entre dans la zone de voisinage d'un obstacle ponctuel noté $\mathcal{O}_i$. Une illustration de la configuration de ce cas test est proposée en figure 2.9.

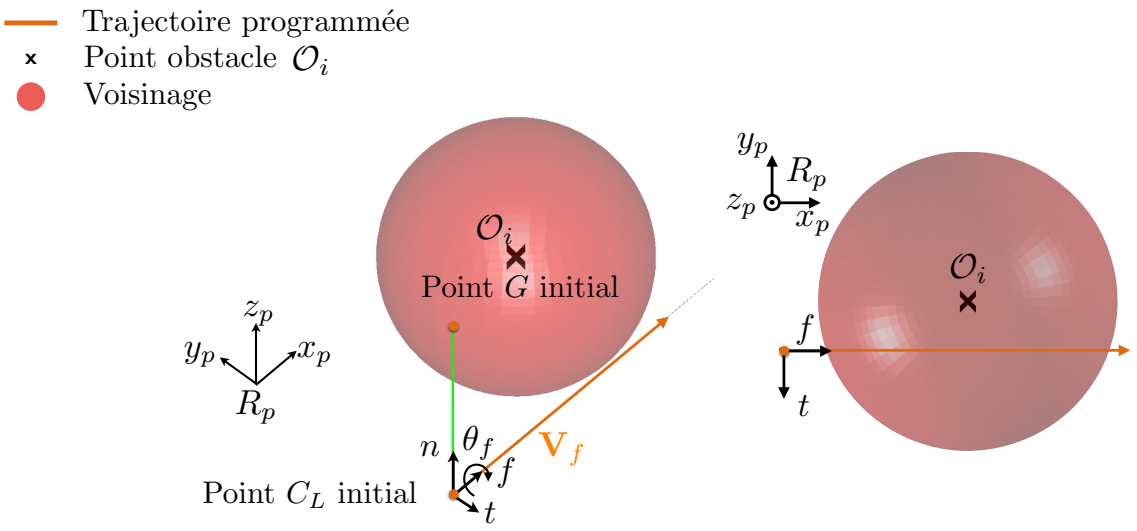


Figure 2.9 – Géométrie du cas test

La première étape est la définition de l'effort généré par l'obstacle. Pour cela l'expression du potentiel répulsif est donnée, comme l'a défini Oussama Khatib [Khatib, 1986], en équation (2.19) appliquée à l'obstacle ponctuel $\mathcal{O}_i$. Cette modélisation assure deux points très importants décrits en conclusion du premier chapitre ; tout d'abord un potentiel tendant vers l'infini à l'approche de

l'obstacle et une variation de ce dernier continue entre la zone hors voisinage et la zone dans le voisinage. De ce fait, le passage de l'une à l'autre n'engendre pas de discontinuité sur la valeur du potentiel généré par l'obstacle ponctuel concerné.

$$\mathcal{U}_{rep_i} = \begin{cases} \frac{1}{2}.\eta.(\frac{1}{r_i} - \frac{1}{r_0})^2 & \text{si} \quad r_i < r_0 \\ 0 & \text{sinon} \end{cases} \tag{2.19}$$

Avec :

— r_i : distance entre le point considéré P de l'outil (ici G) et l'obstacle ponctuel concerné O_i (équations (2.20) et (2.21)) :

$$r_i = ||\boldsymbol{O_i.G}|| \tag{2.20}$$

$$r_i = \sqrt{(x_G - x_{O_i})^2 + (y_G - y_{O_i})^2 + (z_G - z_{O_i})^2} \tag{2.21}$$

— r_0 : valeur du voisinage utilisé (rayon de la sphère représentée en figure 2.9),

— η : constante associée au potentiel répulsif.

La représentation d'un quart de ce potentiel est tracée en figure 2.10(a).

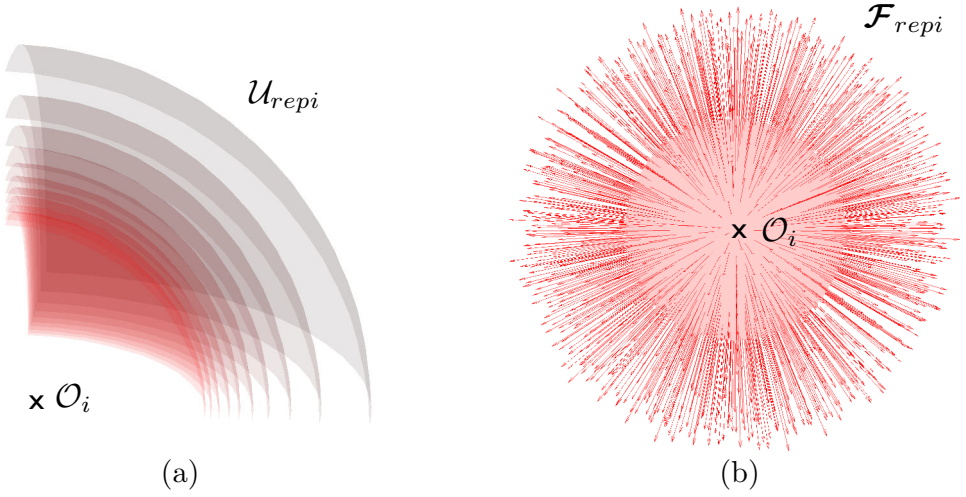


Figure 2.10 – (a) Potentiel répulsif (b) Effort répulsif

En recherchant à présent l'opposé du gradient de ce potentiel, l'expression de l'effort répulsif $\boldsymbol{\mathcal{F}_{rep_i}}$ correspondant est donnée en équation (2.22) et illustrée en figure 2.10(b).

$$\boldsymbol{\mathcal{F}_{rep_i}} = -\nabla(\mathcal{U}_{rep_i}) = -\frac{\partial \mathcal{U}_{rep_i}}{\partial r_i}.\begin{pmatrix} \frac{\partial r_i}{\partial x} \\ \frac{\partial r_i}{\partial y} \\ \frac{\partial r_i}{\partial z} \end{pmatrix} \tag{2.22}$$

En notant $\boldsymbol{u_i} = \frac{\boldsymbol{O_i.G}}{||\boldsymbol{O_i.G}||}$, l'expression devient l'équation (2.23).

$$\boldsymbol{\mathcal{F}_{rep_i}} = \begin{cases} \eta.(\frac{1}{r_i} - \frac{1}{r_0}).\frac{1}{r_i^2}.\boldsymbol{u_i} & \text{si} \quad r_i < r_0 \\ \boldsymbol{0} & \text{sinon} \end{cases} \tag{2.23}$$

L'évolution du potentiel répulsif $\mathcal{U}_{rep_i}$ et de l'effort répulsif $||\mathcal{F}_{rep_i}||$ dépend donc du voisinage utilisé r_0 et de la constante η choisie. La figure 2.11 montre que le changement de voisinage joue bel et bien son rôle.

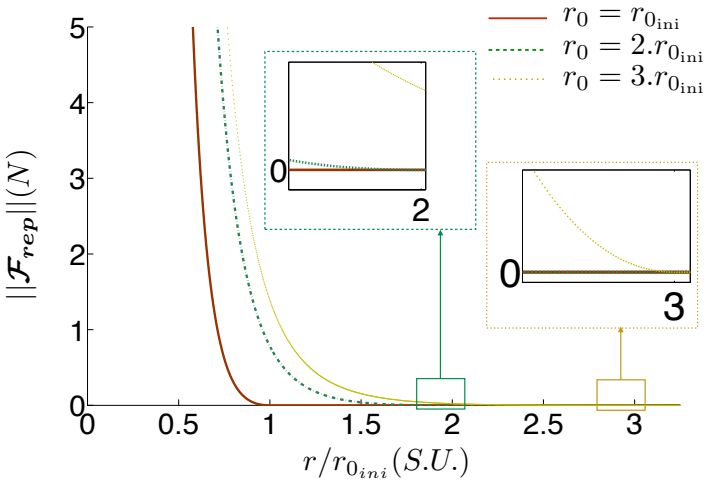


Figure 2.11 – Évolution de l'effort répulsif en fonction de $r/r_{0_{ini}}$ pour différents r_0

Un gros avantage de la formulation utilisée se situe au niveau de la fluidité du passage d'une position hors voisinage à une position soumise à un effort répulsif. En effet, la formulation appliquée comportant une continuité C_1 au niveau du passage du voisinage, visible sur la figure 2.11, de l'effort reçu assure une fluidité dans la phase de réorientation de l'axe outil plus importante qu'avec l'utilisation d'un effort répulsif discontinu au niveau de l'entrée du voisinage. Par contre, cette simulation montre bien que plus le voisinage est petit, plus l'évolution de l'effort répulsif est brutale.

Cet aperçu permet de valider l'idée que, plus la dimension de l'espace représentant le voisinage est importante, plus le point étudié est affecté tôt. Bien évidemment, la difficulté principale de la présence du voisinage dans l'expression de l'effort répulsif est que, en plus de modifier la zone où l'effort répulsif est différent de zéro, elle modifie également sa valeur (voir en équation (2.23)).

Il paraît également important de relever le fait que la valeur de la constante η peut avoir une forte influence sur la réponse du système comme illustré en figure 2.12. Cela permettrait à l'utilisateur d'imposer un évitement plus brutal avec une assurance de non-collision entre le point étudié et la zone de collision. Ce test, utilisant le même paramétrage de voisinage, permet de conclure que l'expression de l'effort répulsif via η permet de gérer une certaine zone de sécurité, dans laquelle il serait très difficile de contrer la valeur de ce dernier.

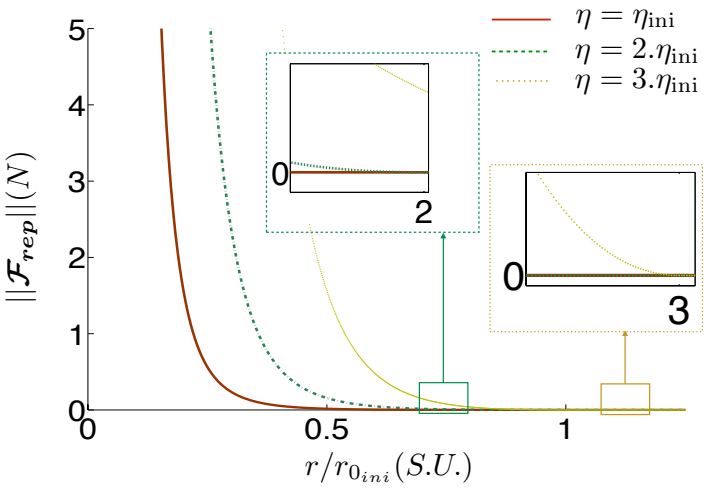


Figure 2.12 – Évolution de l'effort répulsif en fonction de $r/r_{0_{ini}}$ pour différents η

Grâce à l'utilisation de **Geo5XField**, l'assurance de non-collision est respectée avec une limite d'effort répulsif tendant vers l'infini à l'approche de l'obstacle concerné. Le moment répulsif, nécessaire pour l'expression complète du second membre de l'équation générale des moments en C_L s'obtient alors avec l'équation (2.24).

$$\boldsymbol{\mathcal{M}_{rep_i}} = \boldsymbol{C_L G} \wedge \boldsymbol{\mathcal{F}_{rep_i}} \tag{2.24}$$

Le résultat obtenu pour cet exemple est tracé sur la figure 2.13. Afin d'avoir une meilleure clarté sur les figures présentées, seule la modification d'orientation autour de $\boldsymbol{f}$ est utilisée avec l'hypothèse de découplage du comportement dans les deux plans énoncés précédemment.

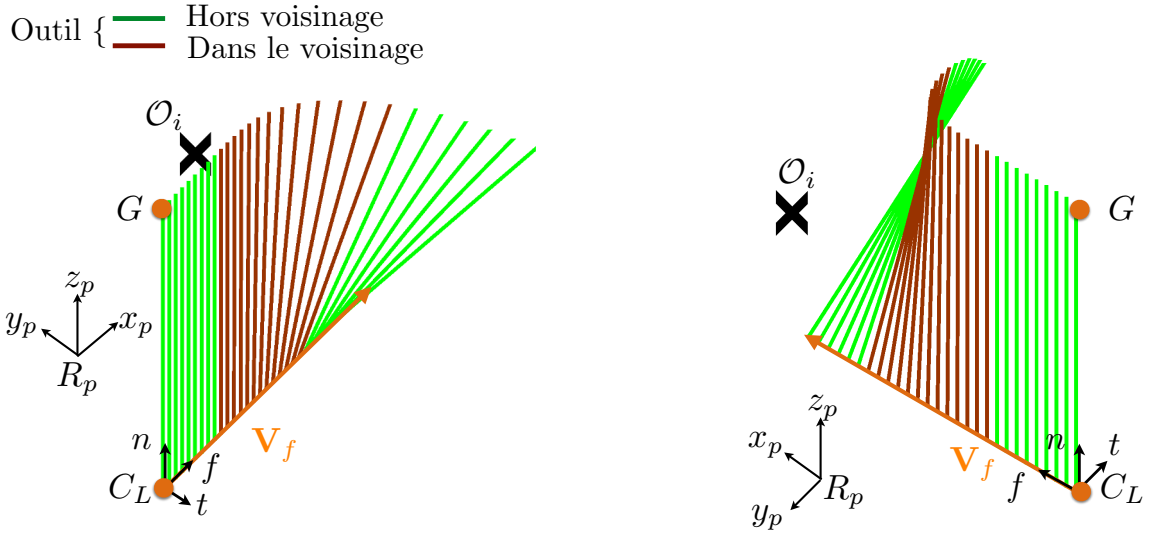


Figure 2.13 – Simulation avec moment répulsif

Cette simulation montre bien qu'il est possible d'éviter l'obstacle ponctuel émettant un effort répulsif dans son voisinage tout en conservant la trajectoire programmée du point C_L. Cependant, il est à présent indispensable de générer une action de rappel afin de fournir à l'outil une position d'équilibre. L'objectif est même de ramener l'outil dans son orientation programmée en plus d'obtenir un équilibre. En effet, la vitesse de coupe effective choisie par l'utilisateur, dans le cadre d'une utilisation d'outil hémisphérique, dépend de l'orientation de ce dernier. Une modification trop importante ou sur une durée trop longue peut influer sur la qualité de surface attendue pour la phase de finition.

3.2 Modélisation de l'action de rappel

La simulation précédente illustre bien, dans le cadre d'une étude dynamique, la nécessité de la programmation d'une action de rappel dans le but d'équilibrer le point outil étudié. L'objectif principal du rappel de l'axe outil en orientation programmée est de trouver un compromis entre une norme de ce dit effort de rappel élevé retardant l'évitement d'obstacle avec un retour en orientation initiale rapide et brusque ou une norme plus basse conduisant à une modification de l'orientation importante ainsi qu'un retour en orientation programmée plus lent. Un compromis est donc à trouver entre ces deux comportements extrêmes de réorientation de l'axe outil.

Il est important de prendre en compte la variation de cette action de rappel afin de ne pas générer d'oscillations trop importantes autour de l'orientation programmée de l'outil. Par la suite, l'action de rappel est construite afin d'obtenir un comportement semblable à celui d'un système de type ressort amortisseur.

3.2.1 Expression du moment de rappel

Dans ce type de configuration, le moment de rappel doit permettre à l'outil de revenir dans son orientation programmée quand cette dernière est modifiée. L'idée principale retenue lors de ces travaux est d'appliquer au système un moment de rappel de type ressort amortisseur angulaire illustré en figure 2.14 avec une représentation uniquement pour l'angle θ_f.

L'expression du moment de rappel obtenu avec ce type de modélisation est donnée en équation (2.25) :

$$\boldsymbol{\mathcal{M}_{rap}} = -k.(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prog}}) - c.(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_{\text{prog}}) \tag{2.25}$$

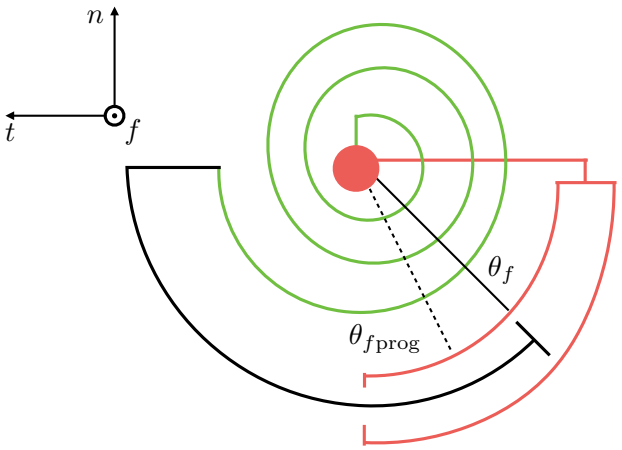


Figure 2.14 – Modélisation du ressort amortisseur angulaire selon θ_f

avec comme paramètres :

— k : coefficient de raideur,

— c : coefficient d'amortissement,

— $\boldsymbol{\theta}_{\text{prog}}$: angle programmé,

— $\dot{\boldsymbol{\theta}}_{\text{prog}}$: vitesse de rotation programmée.

Cette modélisation du couple de rappel a donc été introduite sur l'exemple précédent et la figure 2.15 illustre le retour en orientation programmée après franchissement de l'obstacle sans réglage à priori des constantes utilisées.

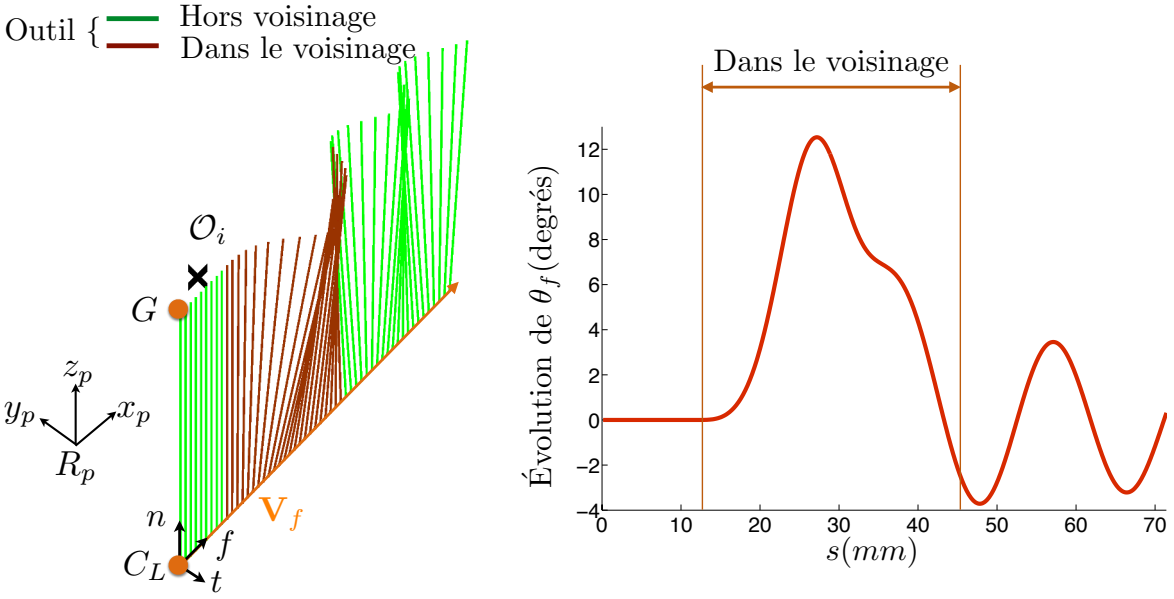


Figure 2.15 – Simulation avec effort répulsif et de rappel

L'action de rappel joue bien son rôle, cependant il reste, après être sorti du voisinage dont les zones d'entrée et de sortie sont représentées, des oscillations qui s'atténuent progressivement. L'évolution du système d'équations traduisant le Principe Fondamental de la Dynamique avec

introduction du moment de rappel et du moment répulsif, s'écrit comme donné en équations (2.26) et (2.27) avec $\mathcal{M}_{frep}$ et $\mathcal{M}_{trep}$ les projections des couples répulsifs écrits au point C_L.

$$J_f.\ddot{\theta}_f + c_f.(\dot{\theta}_f - \dot{\theta}_{fprog}) + k_f.(\theta_f - \theta_{fprog}) - \mathcal{M}_{frep_i} = 0 \tag{2.26}$$

$$J_t.\ddot{\theta}_t + c_t.(\dot{\theta}_t - \dot{\theta}_{tprog}) + k_t.(\theta_t - \theta_{tprog}) - \mathcal{M}_{trep_i} = 0 \tag{2.27}$$

Dans les simulations présentées, les paramètres d'inertie, de coefficient d'amortissement et de raideur sont considérés identiques pour les deux équations. Bien évidemment, il est envisageable d'avoir différentes valeurs dans les deux directions concernées dans le but d'attribuer une priorité de repoussement dans une direction.

Par la suite, la problématique de la présence d'oscillations après retour en orientation programmée est traitée. Pour répondre à ce problème de repositionnement correct après franchissement d'obstacle, l'analyse de la réponse de ce type de système en régime transitoire est étudiée.

3.2.2 Condition de fonctionnement apériodique

Dans le dessein de supprimer les oscillations observées, l'idée est de calculer formellement les valeurs des coefficients du système masse ressort amortisseur à utiliser afin de se trouver dans le cadre d'une évolution apériodique ou pseudo-périodique, mais améliorée. Le choix dépend du besoin principal de l'utilisateur, soit ce dernier ne souhaite aucune oscillation et donc une réponse moins réactive, soit il accepte un dépassement avec une rapidité plus élevée. Il est donc important de trouver le compromis idéal entre rapidité de retour, dépassement et oscillation autour de l'orientation programmée.

Autrement dit, cela correspond dans le domaine de l'automatique à la recherche de la réponse optimale du système étudié sur les critères de rapidité et de stabilité. Cette étude est réalisée pour un des deux angles, bien évidemment, il est également applicable dans l'autre direction de rotation. Elle peut sembler très détaillée pour le type d'équation sans second membre étudié, mais ce développement servira par la suite, lors de l'étude de la réponse du système avec la présence d'un second membre dans la zone de voisinage de l'obstacle.

L'équation différentielle traduisant le comportant d'un système masse ressort amortisseur angulaire est donnée, en supposant des angles initiaux nuls, en équation (2.28) lorsque l'outil se retrouve hors voisinage et donc avec un second membre nul :

$$J.\ddot{\theta} + c.\dot{\theta} + k.\theta = 0 \tag{2.28}$$

L'écriture sous la forme canonique de ce système du second ordre est développée en équation (2.29) :

$$\ddot{\theta} + 2.\xi.\omega_0.\dot{\theta} + \omega_0^2.\theta = 0 \tag{2.29}$$

avec :

— $\omega_0 = \sqrt{\frac{k}{J}}$: pulsation propre,

— $\xi = \frac{c}{2.\sqrt{k.J}}$: facteur d'amortissement.

Si $\xi = \sqrt{2}/2$, le résultat présente le temps de réponse le plus faible avec évidemment un léger dépassement, étant donné que le facteur d'amortissement utile est inférieur à celui représentant le passage à un régime apériodique ($\xi = 1$).

En effet, comme le rappelle la figure 2.16, la réponse peut se trouver dans trois régimes différents : le régime pseudo-périodique ($\xi < 1$), le régime apériodique critique ($\xi = 1$) et le régime apériodique ($\xi > 1$). Le cas particulier de la meilleure solution de compromis entre temps de réponse et dépassement ($\xi = \sqrt{2}/2$) est également tracé.

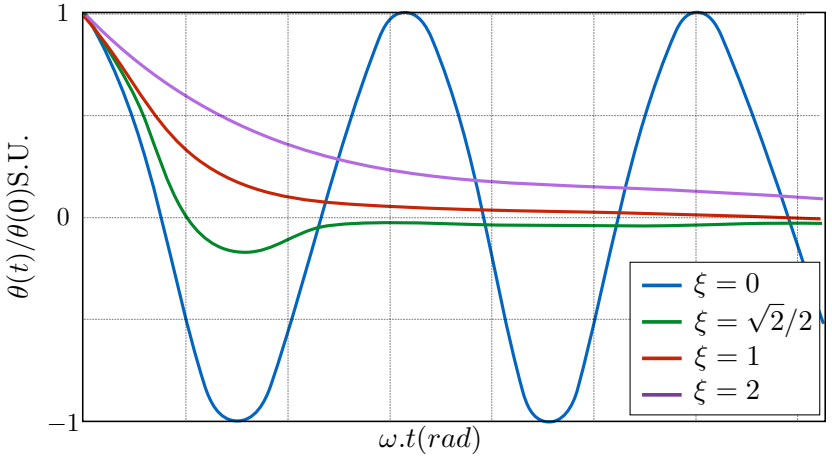


Figure 2.16 – Représentation des différents régimes

Afin de démontrer l'écriture des différents modes de ce système du second ordre, le polynôme caractéristique correspondant est énoncé, avec en inconnue le ω, en équation (2.30) :

$$\omega^2 + 2.\xi.\omega_0.\omega + \omega_0^2 = 0 \tag{2.30}$$

avec le discriminant donné en équation (2.31).

$$\Delta = 4.\omega_0^2.(\xi^2 - 1) \tag{2.31}$$

La résolution de ce système du second ordre donne deux solutions possibles données en équations(2.32).

$$\begin{cases} \omega 1 = \omega_0.(-\xi + \sqrt{\xi^2 - 1}) \\ \omega 2 = \omega_0.(-\xi - \sqrt{\xi^2 - 1}) \end{cases} \tag{2.32}$$

Durant l'étude présentée, le cas de l'utilisation d'un régime apériodique est sélectionné afin de ne jamais dépasser l'orientation programmée. Les deux racines sont alors des réels purs, donc, après calcul, l'évolution de θ est donnée en équation (2.33) :

$$\theta(t) = A_1.e^{-\frac{t}{\tau_1}} + A_2.e^{-\frac{t}{\tau_2}} \tag{2.33}$$

avec les constantes données en équations (2.34).

$$\begin{cases} \tau_1 = \frac{1}{\omega_0.(\xi - \sqrt{\xi^2-1})} \\ \tau_2 = \frac{1}{\omega_0.(\xi + \sqrt{\xi^2-1})} \\ A_1 = \frac{\tau_1.\tau 2}{\tau_1 - \tau 2}.(\dot{\theta}_0 + \frac{\theta_0}{\tau 2}) \\ A_2 = \frac{\tau_1.\tau 2}{\tau_2 - \tau 1}.(\dot{\theta}_0 + \frac{\theta_0}{\tau 1}) \end{cases} \tag{2.34}$$

Pour conclure sur cette étude formelle, le critère à retenir afin de se placer en régime apériodique est donné en équation (2.35) :

$$c \geqslant 2.\sqrt{k.J} \tag{2.35}$$

Une fois que les valeurs des coefficients permettant d'obtenir en théorie un régime apériodique sont déterminées, elles sont utilisées dans la simulation. Le résultat obtenu est alors illustré en figure 2.17. Dans ce cas, il n'y a pas de dépassement par rapport à l'orientation programmée, ce qui est, dans le cadre de cette étude un des objectifs principaux concernant le retour en orientation programmée. Il est possible de noter sur cette figure que tant que le point étudié ne rentre pas dans la zone de voisinage, sa position n'est pas modifiée, par contre, lors de la sortie du voisinage, le point étudié n'est pas tout à fait de retour et donc le repositionnement perdure.

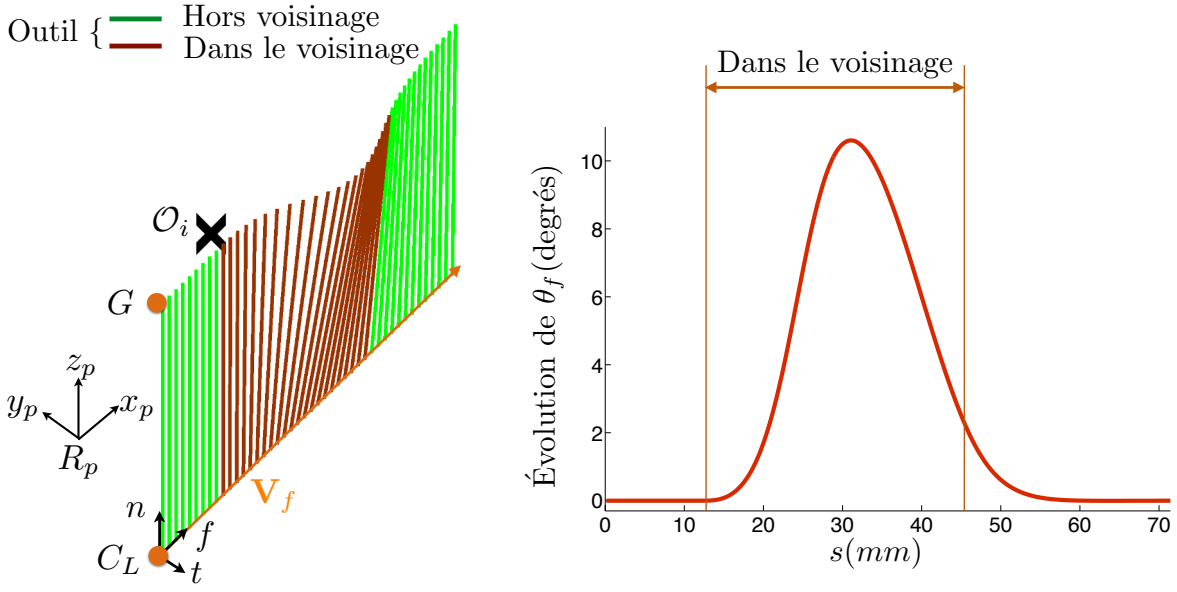


Figure 2.17 – Simulation avec effort répulsif et de rappel

3.3 Attribution de valeurs fixes aux paramètres étudiés

Dans le dessein d'obtenir plus de clarté lors de toutes les simulations réalisées par la suite, le tableau 2.1 donne les paramètres choisis permettant d'obtenir un exemple de fonctionnement souhaité. Si aucun changement de variables pour les simulations à venir n'est mentionné, alors les données ci-dessous s'appliquent. Bien évidemment, les simulations présentées sont totalement réalisables avec un choix de paramètres différent.

Paramètres	Symboles	Valeur	Unité
Valeur du voisinage	r_0	10.10^{-3}	m
Vitesse d'avance programmée	V_f	1	$m.min^{-1}$
Coefficient de $\mathcal{F}_{rep_i}$	η	10^{-7}	$N.m^{-4}$
Facteur d'amortissement	ξ	1	$S.U.$

Tableau 2.1 – Paramètres fixés

L'étude développée dans le chapitre suivant portant, en partie, sur le régime permanent du modèle étudié permettra d'attribuer une valeur aux paramètres restés libres (J,k,c).

La suite de l'étude porte sur des contraintes géométriques, cinématiques et dynamiques, pour l'instant non évoquées, qui permettent de montrer l'influence de ces paramètres. Ce paramétrage est par la suite conservé afin de pouvoir détailler chacune des modifications entreprises.

4 Effet des paramètres dynamiques sur la trajectoire obtenue

Après avoir montré que, via **Geo5xField**, il est possible d'éviter les collisions entre l'axe outil et les points considérés comme obstacles, il est alors utile de s'interroger sur le choix des paramètres permettant différentes entrées et sorties du voisinage. En effet, intuitivement, il paraît assez clair qu'en imposant une raideur plus importante le changement d'orientation sera plus tardif, mais également plus violent. L'étude est alors réalisée selon deux points de vue, tout d'abord avec une partie concernant l'aspect géométrique, avec une analyse de l'évolution des angles outil dans le repère R_{fnt} puis avec une étude cinématique et dynamique des trajectoires corrigées.

Afin de valider cette vision intuitive du comportement du système, l'entrée et la sortie de la zone de voisinage sont étudiées en fonction des variations des paramètres principaux données dans le tableau 2.2. La valeur de ξ est fixée à 1 afin d'assurer un régime apériodique.

Paramètre	Valeur minimum	Valeur maximum
r_0	$r_{0_{ini}}$	$2.r_{0_{ini}}$
J	J_{ini}	$10.J_{ini}$
k	k_{ini}	$10.k_{ini}$

Tableau 2.2 – Niveau des paramètres pour la simulation

Une partie des résultats de ces différents essais est donnée en figure 2.18 en traçant l'évolution du point G en projection dans le plan $\boldsymbol{f}$, $\boldsymbol{t}$.

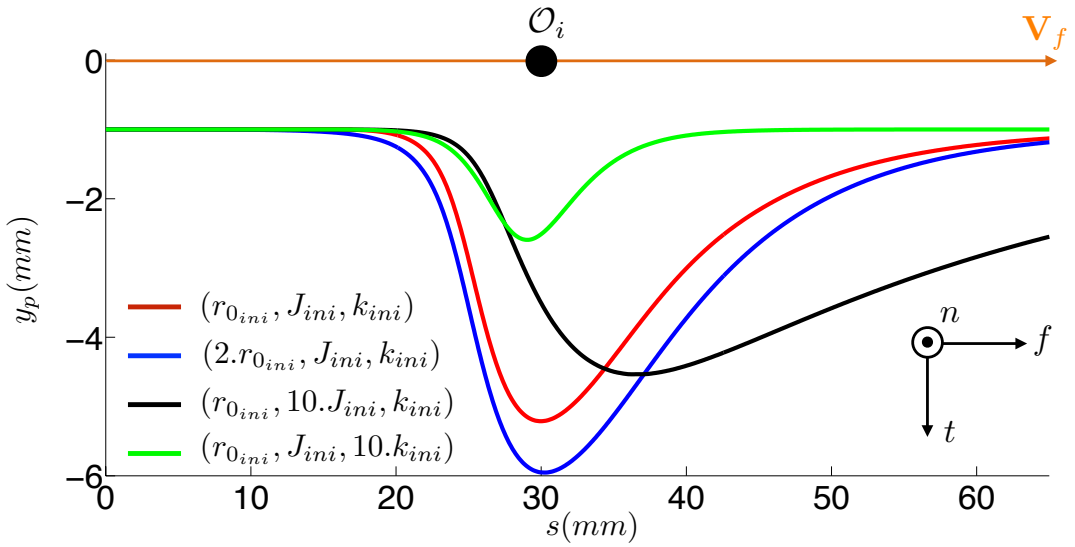


Figure 2.18 – Influence de la raideur, de l'inertie et du voisinage sur l'orientation outil

Ce graphique montre alors que le paramètre de voisinage r_0 peut être vu tel un paramètre d'anticipation permettant de modifier l'orientation outil plus ou moins loin de l'obstacle ponctuel. S'il est choisi trop grand, il risque de désorienter, de manière très faible, mais beaucoup trop tôt l'axe outil et donc la vitesse de coupe effective programmée.

En ce qui concerne le paramètre de rigidité k, son influence est la vitesse à laquelle l'orientation de l'axe outil doit être changée afin d'éviter l'obstacle pour une valeur donnée de J. Lors de la sortie du voisinage, cette figure montre qu'une valeur importante de ce paramètre impose un retour en orientation programmée plus rapide.

Comme cela pouvait être prévu, les résultats démontrent également que plus l'inertie est élevée, plus lente est l'action engendrée par l'effort répulsif. Pour ce qui concerne le retour en orientation programmée, il est également possible de remarquer que forcément, une inertie importante ralentit la réorientation.

Par la suite les travaux portent sur l'impact des paramètres J et k sur les angles d'orientations puis sur l'aspect cinématique et dynamique du choix de ces paramètres.

4.1 Étude géométrique de la modification angulaire de l'axe outil

Dans le cadre d'une étude géométrique, le paramètre jouant le rôle le plus important devrait être la raideur, car dans le PFD il représente le coefficient rattaché au paramètre géométrique θ. De ce fait, les essais devraient permettre de démontrer une plus forte influence de la modification de la raideur sur la trajectoire obtenue que d'autres paramètres. La figure 2.19 illustre l'évolution de la norme de la différence entre $\theta_{\text{corrigé}}$ et $\theta_{\text{stratégie}}$. Afin de faciliter la lecture des graphiques présentés par la suite, les courbes en fonction du voisinage ne sont pas prises en compte car leur effet est moins prononcé que les deux autres paramètres étudiés (J et k).

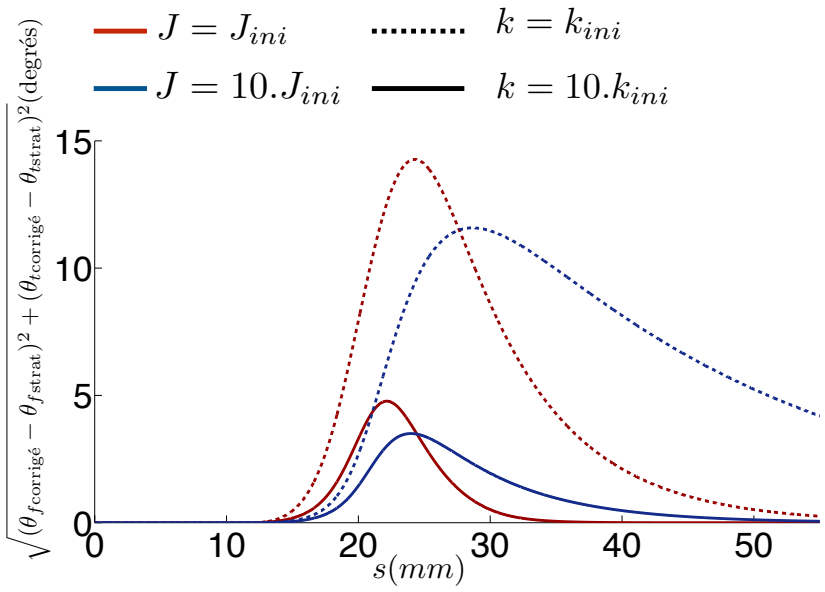


Figure 2.19 – Différence entre $\theta_{\text{corrigé}}$ et $\theta_{\text{stratégie}}$

Dans le cas de cette étude $\theta_{\text{stratégie}}$ est nul tout au long de la trajectoire. Il est important de repérer que l'impact sur la géométrie de la trajectoire est majoritairement due à la modification de la constante de raideur. En effet, deux groupes sont formés selon la valeur de la raideur utilisée. Par la suite, la valeur maximum de réorientation admissible est fixée en fonction des contraintes d'usinage définies par l'utilisateur tel que le rayon effectif de coupe admissible.

De plus, pour un utilisateur connaissant l'architecture machine utilisée et en se servant de la transformation géométrique inverse correspondante, il est alors possible de décrire la géométrie de la trajectoire obtenue dans l'espace articulaire. Il en découle une norme angulaire limite qui serait alors obtenue permettant de fixer la limite admissible à la constante k utilisée. Bien évidemment l'inertie joue également un rôle sur la trajectoire obtenue. Par conséquent, cette dernière retarde la modification d'orientation mais impose un fort ralentissement sur le retour en orientation programmée.

Par la suite, le comportement des différentes réponses permet de mettre en avant les effets des aspects cinématique et dynamique découlant du PFD.

4.2 Étude de l'impact de l'orientation de l'axe outil modifiée sur le comportement cinématique et dynamique du système

Pour une étude cinématique, la courbe correspondante est celle donnant l'évolution de la norme des variations angulaires fournie en figure 2.20.

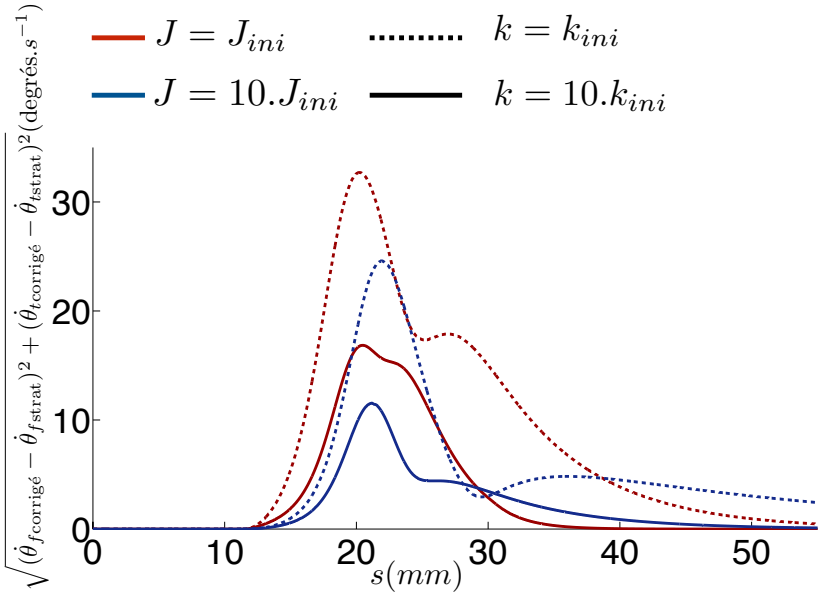


Figure 2.20 – Différence entre $\dot{\theta}_{\text{corrigé}}$ et $\dot{\theta}_{\text{stratégie}}$

Le premier point à relever est qu'aucune différence flagrante entre simulations n'est à relever et cela est cohérent. En effet, le paramètre qui joue majoritairement sur l'évolution de cette vitesse est le paramètre d'amortissement c qui est fixé dans le but de conserver un régime apériodique. Ce paramètre lie donc les paramètres d'inertie J et de raideur k. Bien évidemment, la réponse cinématique n'est pas liée uniquement à la constante d'amortissement d'après l'équation de la dynamique étudiée, c'est pour cela qu'il est tout de même possible de relever des différences entre ces quatre simulations.

Enfin, le paramètre qui, au regard de l'équation cité précédemment, devrait jouer un rôle prédominant sur l'étude de la dérivée seconde, est l'inertie associée à l'outil. Afin de vérifier cette hypothèse, l'évolution d'accélération angulaire est alors tracée en figure (2.21).

Il est alors clair que, dans le cas de l'étude dynamique, le paramètre jouant le rôle principal est l'inertie. Les courbes représentées en rouge correspondent à l'inertie la plus faible. Ces dernières sont bien éloignées des autres essais alors que, comme il était possible de s'y attendre, l'autre paramètre étudié k n'a pas une influence prédominante sur l'évolution de l'accélération. Cela indui-

rait un choix numérique pour l'inertie utilisée lors de la modélisation en fonction de l'accélération disponible sur la machine utilisée.

Il est donc possible de conclure que le PFD utilisé au cours de ces travaux assure une variation effective d'angle fluide jusqu'à l'ordre 2 où chacun des paramètres utilisés joue son rôle.

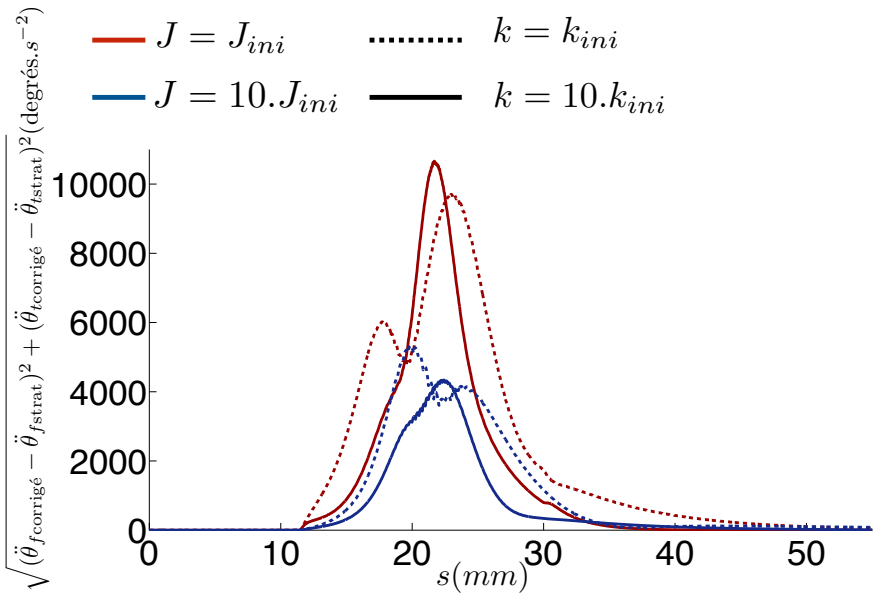


Figure 2.21 – Différence entre $\ddot{\theta}_{\text{corrigé}}$ et $\ddot{\theta}_{\text{stratégie}}$

Comme d'un point de vue géométrique, cette étude pourrait être transférée au point de vue machine afin de limiter le choix des différents paramètres en fonction de la vitesse et de l'accélération admissible maxi pour chacun des axes utilisés. Il est tout de même important de relever que les mouvements des axes, en programmation machine, sont classiquement limités par le jerk, dérivée troisième par rapport au temps [Beudaert, 2013].

Dans le cas de l'étude effectuée à l'aide du PFD, en traçant l'évolution de la dérivée troisième de la trajectoire obtenue (à voir en figure 2.22), il est possible de relever de fortes variations avec un niveau élevé du jerk notamment via la modification de la raideur dont l'augmentation fait croitre le jerk nécessaire à l'évitement considéré.

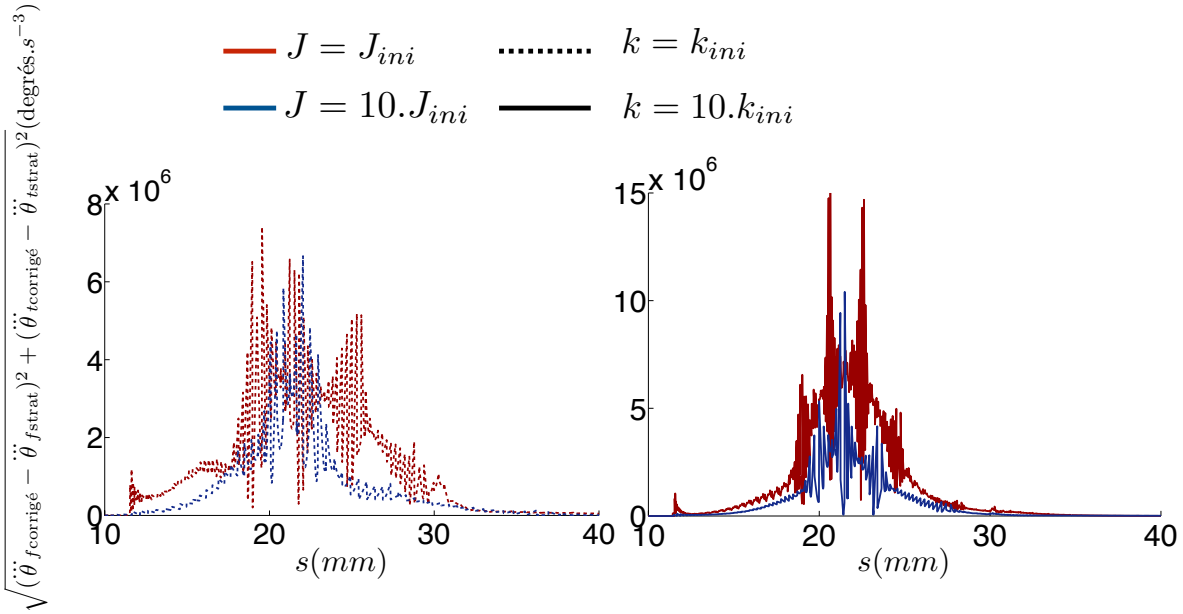


Figure 2.22 – Différence entre $\dddot{\theta}_{\text{corrigé}}$ et $\dddot{\theta}_{\text{stratégie}}$

Bien que la fluidité de variation du jerk ne soit pas traitée dans le cadre de cette étude, il serait tout à fait réaliste d'intégrer une valeur maximum fournie par la MOCN utilisée afin de limiter le choix des paramètres utilisés. Sinon il serait possible d'ajouter à la méthode **Geo5XField** cette notion de jerk et donc un coefficient en degré 3 sur le comportement dynamique de l'outil au travers d'un nouveau paramètre pondérant $\dddot{\theta}_f$ et $\dddot{\theta}_t$, aboutissant ainsi à une solution de l'équation différentielle précédente.

5 Conclusions

Une étude théorique ainsi que diverses simulations ont permis de mettre en évidence l'apport de l'utilisation d'une analogie mécanique couplée avec la méthode des potentiels et des champs répulsifs sur l'évitement de collisions globales, dans le domaine de l'usinage 5 axes.

La première étape des travaux présentés durant cette partie porte sur la description du modèle bâti sur des critères physiques. Une mise en parallèle du modèle $3D$ décrivant le mouvement d'un solide avec le positionnement, selon les repères associés aux trajectoires d'usinage a été présentée. La première hypothèse est de symboliser le mouvement de l'axe outil corrigé par rapport à l'axe outil programmé, dit de stratégie, comme un cas d'étude dynamique. Cela permet de maintenir l'orientation initiale telle que programmée par l'utilisateur initialement. Puis, la méthode de résolution utilisée sous Matlab® dans le but de résoudre l'équation différentielle du second ordre avec second membre a été présentée. Ainsi, la méthode RK4 permet, à l'aide d'approximations, de résoudre l'équation différentielle du mouvement. De plus, une hypothèse de découplage, modélisant la problématique comme l'étude de deux pendules inversés indépendants, lors des simulations est explicitée.

La modélisation des efforts répulsifs retenue est issue des travaux de Khatib [Khatib, 1986]. Ce modèle permet d'assurer l'absence de collision entre le point étudié et l'obstacle disposé dans l'environnement. Ainsi, si le point étudié se trouve dans le voisinage de l'obstacle ponctuel, il est alors soumis à une force répulsive repoussant le corps outil tout en gardant le point piloté C_L le long de la trajectoire programmée. Différents essais permettent de montrer l'influence des paramètres présents dans l'expression mathématique de cet effort répulsif.

Afin de maintenir l'orientation de l'axe outil étudié, il est important de mettre en place un effort de rappel. Pour ce faire, l'étude de l'effet d'un système ressort amortisseur angulaire est réalisée. Les différentes simulations valident, tout d'abord un retour en orientation programmée, puis, après une étude complète du réglage des paramètres présents dans l'expression du rappel, un retour apériodique.

Enfin les différents paramètres sont étudiés afin d'attribuer leur rôle pour la suite de l'étude. L'influence de ces derniers joue sur la brutalité de changement d'orientation obtenue et la vitesse de retour en orientation initiale.

Après avoir exprimé les équations à résoudre qui permettent à l'outil d'éviter des obstacles à l'aide d'efforts répulsifs, l'objectif principal est d'appliquer cette programmation sur des configurations d'usinage plus réalistes.

Tout d'abord, une étude de l'équation du mouvement en régime permanent est décrite afin d'obtenir la résolution complète. Cela permet par la suite de présenter une modélisation plus réaliste de l'environnement et son impact sur la simulation. Les surfaces considérées comme obstacles sont en fait maillées et chaque point issu de ce maillage est considéré comme un obstacle ponctuel. De ce fait, l'impact du maillage sur la trajectoire corrigée de l'axe outil est étudié. Une partie du chapitre suivant porte sur les différentes améliorations possibles concernant l'expression de l'effort répulsif.

Enfin, une nouvelle généralisation est à entreprendre, il s'agit de la modélisation de l'outil. En effet, jusqu'à maintenant, l'outil n'est représenté que par un point guide et un point recevant les différents efforts. Cela n'est pas réaliste et pas forcément adapté à tous les types de simulations d'usinage de surfaces complexes. En conséquence, un maillage de l'outil sera proposé afin de comprendre et de mettre en avant, l'intérêt de **Geo5XField**, mais également la problématique du temps de calcul.

Chapitre 3

Modélisation d'une configuration d'usinage réaliste

Ce troisième chapitre permet d'appliquer l'analogie mécanique découlant de l'écriture du PFD à des simulations réalistes. En effet, la surface obstacle est maillée et les nœuds obtenus sont alors considérés comme obstacles ponctuels. Une modélisation analytique des actions répulsives le long d'un obstacle permet d'étudier l'influence des paramètres et en particulier la taille du maillage retenu. Par la suite, la programmation obtenue est appliquée à une pièce réelle de type *impeller* permettant de montrer l'efficacité du programme obtenu, mais également ses limites. Enfin, la modélisation réaliste de l'outil utilisé est effectuée dans le but d'assurer une prise en compte complète du solide piloté au cours de la trajectoire dans le cadre d'évitement de collision.

Sommaire

1 Introduction 83

2 Résolution de l'équation du mouvement en régime permanent 84

2.1 Étude des paramètres de simulation dynamique en régime permanent 84

2.2 Résolution mécanique 87

2.3 Analyse de la réponse du système 90

3 Modélisation réaliste de l'environnement 92

3.1 Rappel langage APT 92

3.2 Maillage structuré 93

3.3 Proposition de modifications du modèle d'effort répulsif 96

3.4 Simulation sur pièce industrielle de type *Impeller* 100

4 Modélisation réaliste de l'outil 102

4.1 Maillage de l'outil cylindrique 104

4.2 Modélisation de la peau pour l'ensemble outil/porte-outil 107

5 Étude du temps de calcul de la simulation sur pièce industrielle 110

6 Conclusions et perspectives 112

1 Introduction

L'étude théorique concernant l'évitement d'un point considéré comme obstacle est à présent réalisée. Le chapitre précédent a permis de comprendre l'effet des potentiels répulsifs programmés dans le but d'éviter un point obstacle. Le rappel de type ressort-amortisseur a, lui aussi, été configuré afin d'assurer un régime apériodique lors de la sortie d'un voisinage. Le retour en orientation programmée est donc mis en place.

Il semble à présent cohérent de s'intéresser à des pièces dites complexes représentant des surfaces réellement utilisées dans le domaine de l'usinage 5 axes. Le but principal de cette partie est alors de fournir à l'utilisateur la simulation d'une phase d'usinage avec une prise en compte complète de l'environnement considéré comme obstacle et une modélisation outil proche de celui choisi pour la phase d'usinage en question. L'objectif principal de ce chapitre concerne la génération de trajectoires pour l'*impeller* assurant un usinage hors collision sur machine-outil à commande numérique 5 axes. Le type d'opération concerné est illustré par un exemple en figure 3.1.

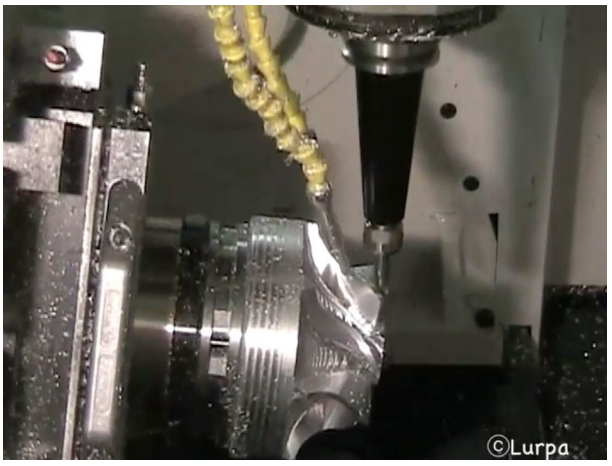


Figure 3.1 – Usinage d'un *impeller*

Dans le but de passer d'une étude simplifiée composée de points obstacles à des surfaces réelles, plusieurs étapes d'évolutions vont composer ce chapitre. En effet, dans le chapitre précédent, l'approche dynamique du problème portait uniquement sur le régime transitoire du système étudié alors que, dans le cadre de l'utilisation d'un maillage d'une surface obstacle, plusieurs points (nœuds du maillage) vont se succéder. Il est alors possible d'émettre une hypothèse sur l'écriture d'un second membre cohérent pour l'étude en régime permanent du système. Cette approche est développée afin d'obtenir une trajectoire lisse lors de la modification de la trajectoire programmée. La courbe suivie par le point C_L de l'outil considéré reste une droite, afin d'attribuer plus facilement les causes aux effets obtenus.

Par la suite, la méthode de génération de maillage structuré est alors présentée avec une description numérique de l'obstacle de type bosse étudié. L'étude effectuée a pour objectif principal

de fixer le nombre de points nécessaires représentant le maillage de cette surface obstacle. Une simulation sur un cas industriel sera alors présentée dans le but de démontrer l'efficacité de la programmation avec un environnement complexe.

Une fois que l'environnement est modélisé avec une présentation de plusieurs expressions du calcul de l'effort répulsif total en fonction du type de pièces étudié, la troisième partie de ce chapitre concerne cette fois ci, la modélisation de l'outil. En effet, jusqu'à présent, l'outil était représenté à l'aide des deux points C_L et G. Bien évidemment, il semble cohérent, afin d'assurer une réelle absence de collision, de prendre en compte la géométrie de l'outil et du porte-outil.

2 Résolution de l'équation du mouvement en régime permanent

Après avoir, dans le chapitre précédent, travaillé hors voisinage et en régime transitoire, au niveau de l'entrée et de la sortie du voisinage, une nouvelle approche est indispensable dans le but de généraliser ces travaux. Le régime permanent est étudié d'un point de vue théorique avec des points obstacles qui représentent à présent les nœuds du maillage d'une surface obstacle. Une première simulation élémentaire est alors réalisée afin de fixer de nouveaux paramètres.

2.1 Étude des paramètres de simulation dynamique en régime permanent

Le but de cette partie est de montrer que la distance entre les points du maillage de la surface obstacle, émettant chacun un effort répulsif, a une importance fondamentale sur les oscillations présentes lorsque l'outil se trouve dans le voisinage. L'étude porte sur le pire des cas, c'est-à-dire, un alignement parfait des points considérés comme obstacles représentant le maillage d'une surface obstacle plane. L'effort répulsif varie alors de manière continue quand l'obstacle est dans le voisinage, par contre, la distance entre ces points joue un rôle sur la trajectoire corrigée obtenue.

Afin de proposer un exemple cohérent avec le domaine de l'usinage, les simulations réalisées par la suite le sont avec des distances entre points d'un ordre de grandeur du *mm*. Cela permet de rester cohérent avec l'utilisation classique de la CFAO. L'exemple illustré en figure 3.2 traite d'une trajectoire programmée avec une possibilité de réorientation autour de $\mathbf{f}$ où une modélisation de l'expression du second membre est envisageable dans le cas de l'étude de la solution permanente.

La programmation présentée pour le traitement d'un point obstacle au cours du chapitre précédent est alors utilisée et les résultats obtenus sont illustrés en figure 3.3.

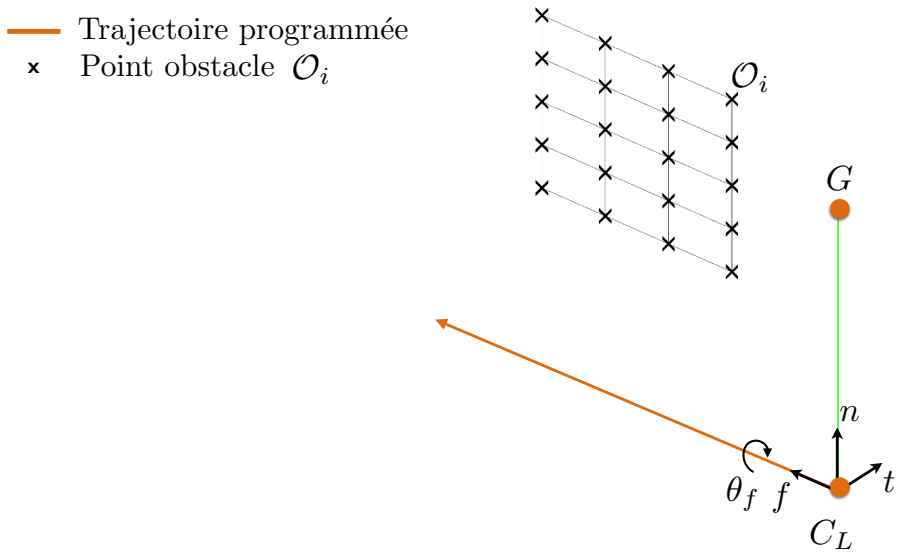


Figure 3.2 – Présentation du cas d'étude

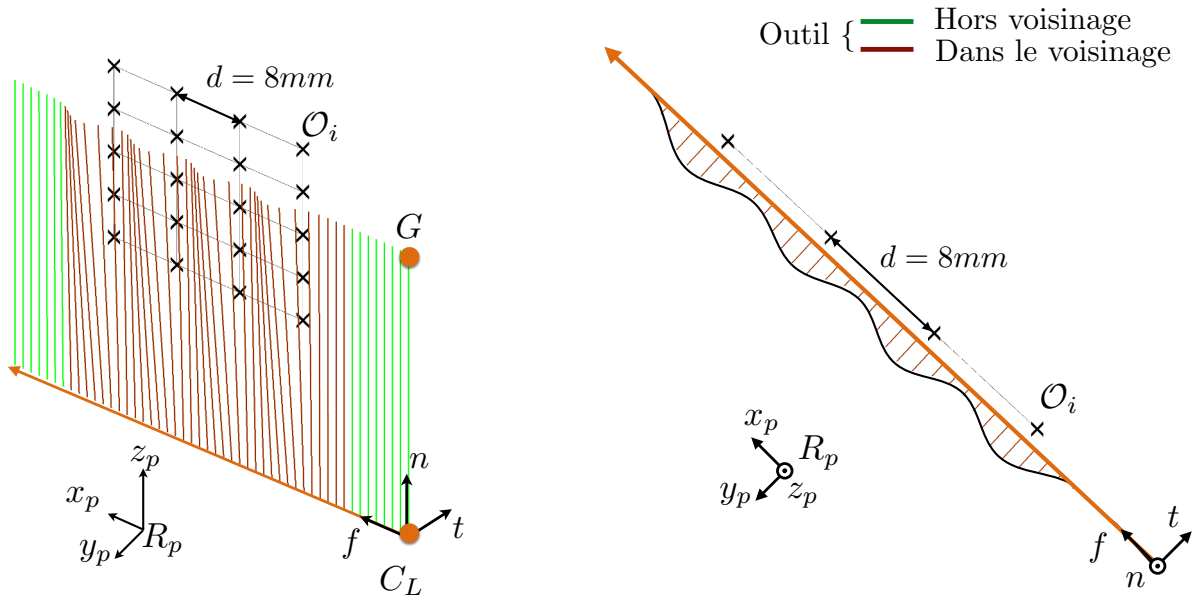


Figure 3.3 – Vision pratique : distance entre obstacles = 8*mm*

Dans cette configuration avec une distance obstacle de 8*mm*, les oscillations sont importantes dans le voisinage. Lorsque le point étudié est à l'intérieur du voisinage, l'équation différentielle du second ordre contient un second membre qui rend l'équation non linéaire. Il est important de noter que le second membre concerné est l'expression du moment répulsif apporté par l'ensemble des points O_i de l'obstacle et calculé au point C_L comme noté en équation (3.1) avec n le nombre de points obstacles.

$$\boldsymbol{\mathcal{M}_{rep}} = \sum_{i=1}^{n} (\boldsymbol{C_L G} \wedge \boldsymbol{\mathcal{F}_{rep_i}}) \tag{3.1}$$

Cette formulation ne permet pas de résolution analytique et devient donc un frein au réglage des différents paramètres. Cela justifie alors le besoin d'établir un modèle à ce second membre.

Il parait possible, pour la rotation autour de **f**, de reformuler l'expression du second membre. Cette dernière est adaptée une fois l'outil dans la zone de voisinage et après la phase transitoire d'entrée dans ce dernier. Son expression est donnée avec l'écriture de l'équation des moments autour de **f** en équation (3.2).

$$J_f.\ddot{\theta}_f(t) + c_f.\dot{\theta}_f(t) + k_f.\theta_f(t) = \mathcal{M}_0.(1 + cos(\omega.t)) \tag{3.2}$$

Avec :

— $\mathcal{M}_0$: valeur approchée de la norme du moment répulsif moyen appliqué au point étudié sur la trajectoire,

— ω : fréquence des oscillations due à la distance entre les obstacles et à la vitesse de l'outil le long de la trajectoire,

$$\omega = 2.\pi.v_0/d \tag{3.3}$$

Afin de valider le choix de cette modélisation du second membre, l'évolution de θ_f en fonction de l'abscisse curviligne peut alors être tracé, avec l'évolution réelle de $\mathcal{M}_{rep_f}$ et celle de $\mathcal{M}_0.(1 + cos(\omega.t))$, en figure 3.4.

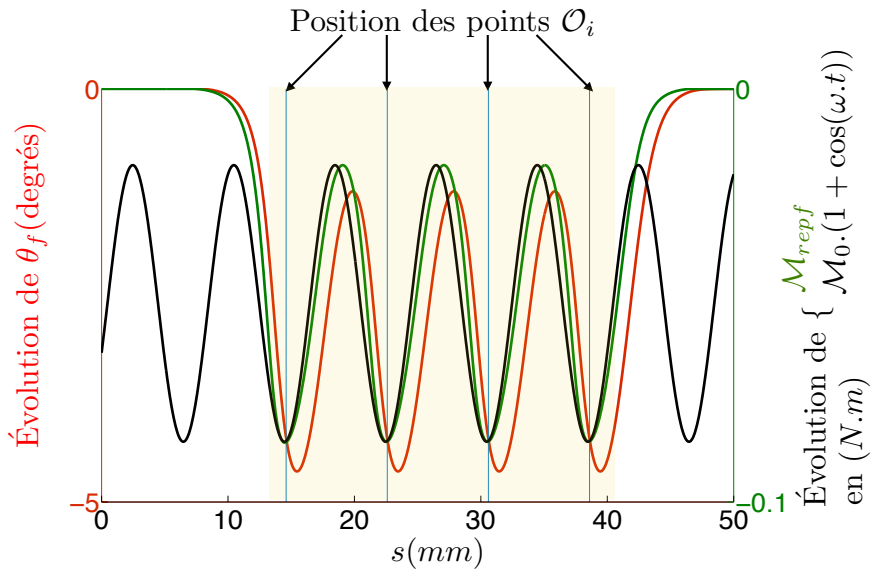


Figure 3.4 – Évolution de θ_f, $\mathcal{M}_{rep_f}$ et $\mathcal{M}_0.(1 + cos(\omega.t))$ en fonction de s

Cette courbe met en évidence une quasi-concordance du moment répulsif réel au cours de la trajectoire avec la modélisation retenue mais comme dit précédemment, uniquement en régime considéré comme permanent (surligné en jaune en figure 3.4). Le second membre employé est donc cohérent avec l'évolution réelle du moment répulsif et est utilisé par la suite pour l'étude

analytique. De plus, l'aspect dynamique est visible avec un retard de modification de l'angle étudié par rapport à l'évolution du moment répulsif.

Une fois l'équation différentielle du second ordre avec second membre explicitée, l'étude porte sur la résolution mécanique dans le but d'obtenir une distance entre obstacles permettant d'atténuer les oscillations entre les points obstacles. Cela permet d'avoir une idée des dimensions du maillage à utiliser pour la surface obstacle concernée.

2.2 Résolution mécanique

Dans le dessein de résoudre ce type d'équation différentielle, le théorème de superposition est appliqué et la solution complète se décompose en une somme de deux réponses décrites en équation (3.4).

$$\theta(t) = \theta_1(t) + \theta_2(t) \tag{3.4}$$

Avec :

— $\theta_1(t)$: solution transitoire de l'équation sans second membre,

— $\theta_2(t)$: solution permanente de l'équation particulière.

L'expression de la solution transitoire décrite dans le chapitre précédent et réexprimée en équation (3.5).

$$\theta_1(t) = A_1.e^{-\frac{t}{\tau_1}} + A_2.e^{-\frac{t}{\tau_2}} \tag{3.5}$$

L'équation (3.6) représente la solution permanente prenant en compte le déphasage du système

$$\theta_2(t) = B.cos(\omega.t - \Phi) + \frac{\mathcal{M}_0}{k} \tag{3.6}$$

avec une amplitude B définie en équation (3.7), un déphasage Φ décrit en équation (3.8) et un amortissement réduit ξ calculé en équation (3.9) de telle sorte à obtenir une réponse apériodique.

$$B = \frac{\mathcal{M}_0/k}{\sqrt{(1-(\frac{\omega}{\omega_0})^2)^2 + 4.\xi^2.\frac{\omega}{\omega_0}^2}} \tag{3.7}$$

$$tan(\Phi) = \frac{2.\xi.\frac{\omega}{\omega_0}}{(1-(\frac{\omega}{\omega_0}))^2} \tag{3.8}$$

$$\xi = \frac{c}{2.J.\omega_0} \tag{3.9}$$

Afin d'étudier la réponse du système, l'évolution de l'amplitude en fonction du rapport pulsation d'excitation ω sur la pulsation propre du système ω_0 est tracée en figure 3.2 dans le cadre d'une réponse apériodique.

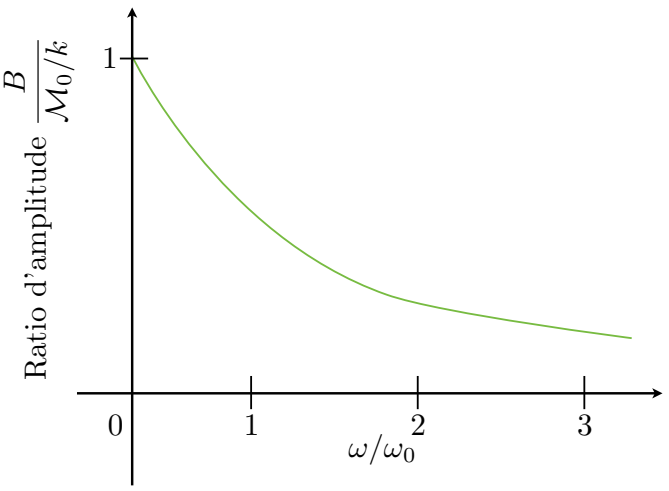


Figure 3.5 – Évolution du ratio d'amplitude adimensionnée en fonction du rapport pulsation propre de l'excitation sur pulsation propre du système

Il est possible de distinguer sur ce graphique, uniquement deux régimes différents (selon la différence entre la pulsation d'excitation et la pulsation propre) car le système possède un amortisseur réglé supprimant la notion de résonance.

— À basse fréquence d'excitation, le système répond avec l'amplitude du système étudié en statique : il ne répond que par son élasticité, car le ratio d'amplitude tend vers 1.

— À haute fréquence d'excitation, le système répond avec une amplitude ne dépendant quasiment que de l'inertie, le ratio d'amplitude tend vers 0.

Ces trois niveaux de fréquences sont plus facilement compréhensibles en traçant le diagramme de Bode correspondant au système étudié du second ordre. Dans ce cas, en passant par le domaine de Laplace, la fonction de transfert décrivant le système étudié obtenue est donnée en équation (3.10) en partant des hypothèses précédentes, nommées classiquement conditions de Heaviside.

— $\ddot{\theta}(0) = 0 rad.s^{-2}$

— $\dot{\theta}(0) = 0 rad.s^{-1}$

— $\theta(0) = 0 rad$

$$H(p) = \frac{\theta(p)}{\mathcal{M}_{rep}(p)} = \frac{1}{k + c.p + J.p^2} = \frac{1}{1 + 2.\frac{\xi}{\omega_0}.p + (\frac{p}{\omega_0})^2} \tag{3.10}$$

Les valeurs des pulsations propres du système sont alors recherchées et notées comme écrit en équation (3.11) dans le cas d'une réponse apériodique, de ce fait, l'équation comporte deux solutions réelles :

$$H(p) = \frac{K}{(1 + \tau_1.p).(1 + \tau_2.p)} \tag{3.11}$$

avec :

— $\tau_1 = \frac{1}{\omega_{c1}}$,

— $\tau_2 = \frac{1}{\omega_{c2}}$.

L'étude des deux coefficients concernés donne le système d'équations (3.12) :

$$\begin{cases} \frac{\omega_{c1}.\omega_{c2}}{\omega_{c1}+\omega_{c2}} = \frac{\omega_0}{2.\xi} \\ \omega_{c1}.\omega_{c2} = \omega_0^2 \end{cases} \tag{3.12}$$

Après développement, l'équation du second degré (3.13) permettant de déterminer les deux solutions décrites en équation (3.11) est alors obtenue.

$$\omega_{c2}^2 - 2.\xi.\omega_0.\omega_{c2} + \omega_0^2 = 0 \tag{3.13}$$

Les deux solutions de cette équation donnent alors les valeurs recherchées de ω_{c1} et ω_{c2}. Dans le cas de cet exemple, le positionnement correspond au cas critique ne donnant qu'une seule pulsation propre ($\omega_{c1} = \omega_{c2} = \omega_0$).

Afin d'être cohérent avec le domaine de la CFAO associé à l'usinage 5 axes sur MOCN, la distance entre nœuds de maillage qui générerait une pulsation forcée égale à la pulsation propre du système est fixée à $d_0 = 2mm$.

Le diagramme de Bode obtenu est représenté en figure 3.6 et permet de situer les deux exemples de répartitions d'obstacles présentés dans cette partie ($d_1 = 8mm$ et $d_2 = 1mm$). Cela permet d'avoir une vision de l'action d'un second membre dont la pulsation forcée est inférieure à la pulsation propre et un autre cas où, au contraire, cette dernière est supérieure à la pulsation propre.

— $\omega_0 = 52{,}35 rad.s^{-1} \leftrightarrow d_0 = 2mm$,
— $\omega_1 = 13{,}09 rad.s^{-1} \leftrightarrow d_1 = 8mm$,
— $\omega_2 = 104{,}72 rad.s^{-1} \leftrightarrow d_2 = 1mm$.

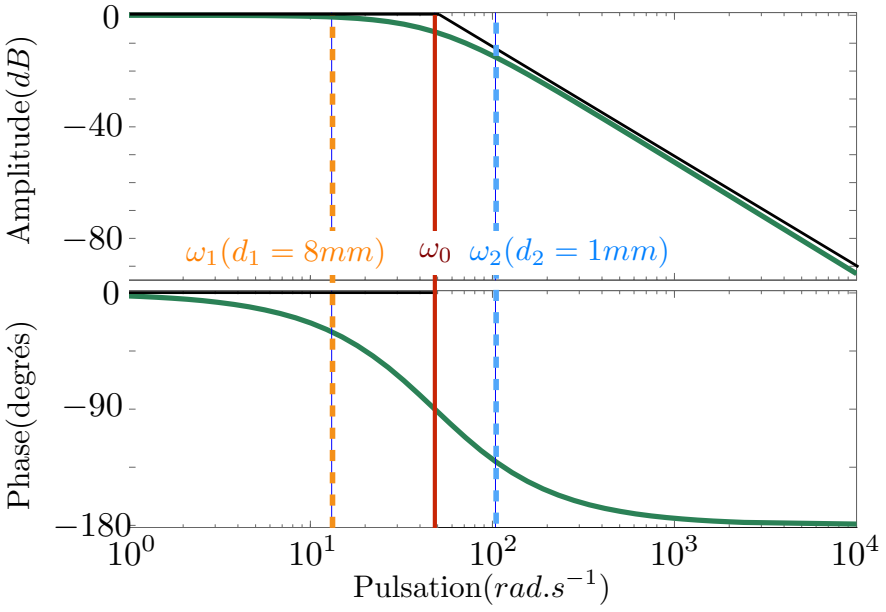


Figure 3.6 – Diagramme de Bode du système amorti

Il est alors possible de remarquer que dans cet exemple, l'affectation d'une distance inter obstacles faible, telle que $\omega > \omega_0$, doit apporter une réponse avec atténuation (par exemple $-20dB$ dans l'exemple présenté). L'analyse du comportement fréquentiel de cette modélisation explique l'influence de la distance admissible entre les points O_i représentant le maillage des surfaces obstacles sur la trajectoire obtenue.

2.3 Analyse de la réponse du système

Il semble alors clair qu'il est possible de réduire les oscillations en augmentant la pulsation forcée ω_i utilisée. En appliquant cette remarque au premier test effectué, le résultat illustré sur la figure 3.7 est obtenu en rapprochant les obstacles ($d = 1mm$).

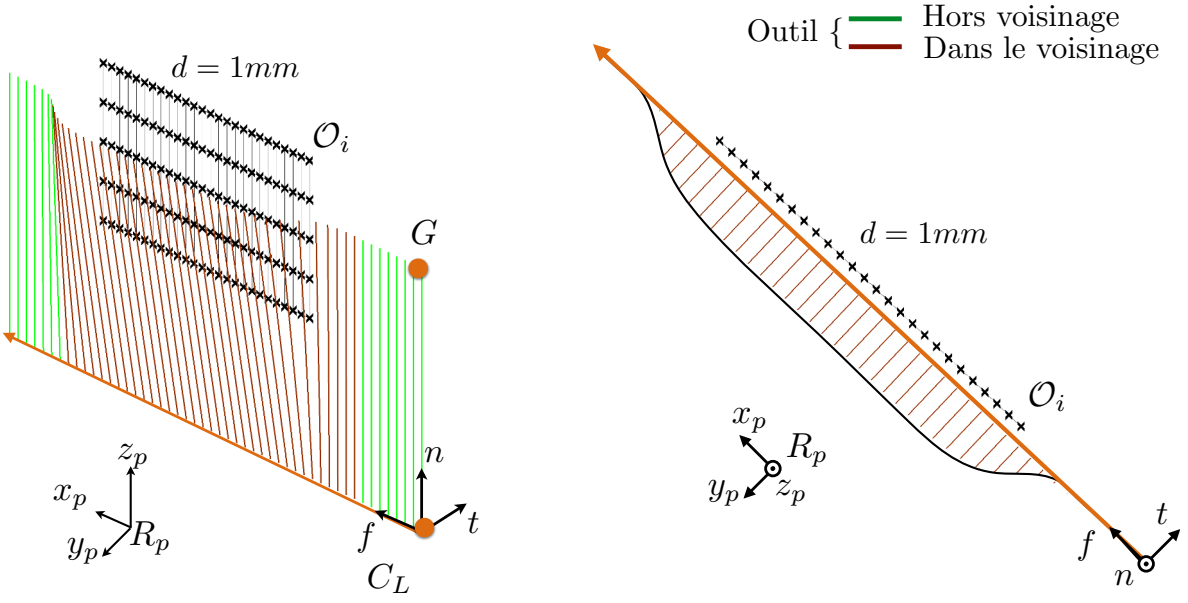


Figure 3.7 – Vision pratique : distance entre obstacles = $1mm$

La trajectoire obtenue semble beaucoup plus lisse que la précédente (non oscillante), cependant il est important de remarquer comme illustré en figure 3.8 que malgré le rapprochement des obstacles la position du point G continu d'osciller. Théoriquement, il faudrait que la distance entre obstacles soit nulle pour ne plus avoir d'oscillation.

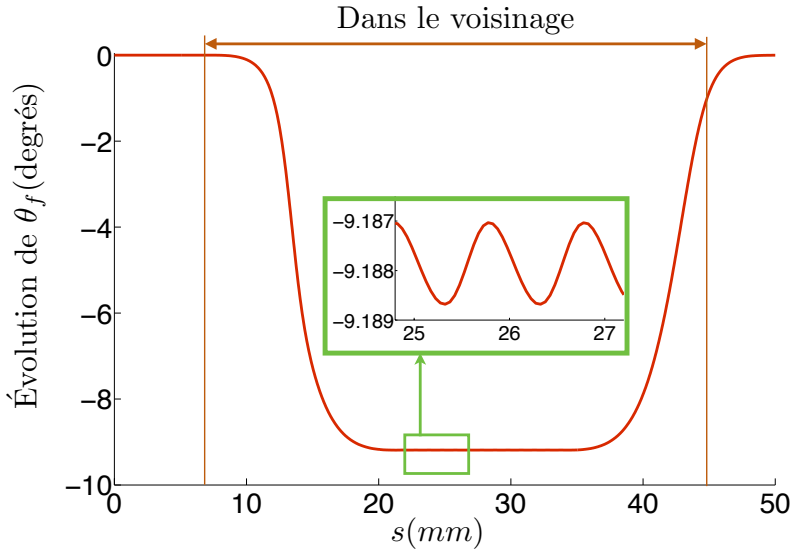


Figure 3.8 – Évolution de θ_f en fonction de s

Cette approche permet de fixer de nouveaux paramètres qui resteront constants durant la suite de ces travaux. En effet, le choix de la pulsation propre du système ω_0 dans le but de correspondre à une dimension de maillage cohérente pour l'étude effectuée donne la relation (3.14).

$$\omega_0 = \sqrt{\frac{k}{J}} = 52{,}35 rad.s^{-1} \tag{3.14}$$

Le tableau 3.1 complète alors le tableau 2.1 donné en conclusion du chapitre 2 en choisissant $k = 1 N.m.rad^{-1}$. -

Paramètres	Fixé	Calculé	Symboles	Valeur	Unité	Équation
Distance entre points	X		d_0	2.10^{-3}	m	
Pulsation propre		X	ω_0	52,35	$rad.s^{-1}$	(3.3)
Coefficient de raideur	X		k	1	$N.m.rad^{-1}$	
Inertie de l'outil		X	J	$3{,}65.10^{-4}$	$kg.m^2$	(3.14)
Coefficient d'amortissement		X	c	$38{,}20.10^{-3}$	$kg.m^2.s^{-1}$	(3.9)

Tableau 3.1 – Paramètres

Il est important de rappeler que, pour toutes les simulations à venir, la valeur de ces derniers reste inchangée sauf indications contraires. À la suite de cette étude théorique permettant de résoudre l'équation du mouvement en régime permanent et de fixer les paramètres restants, la prochaine étape de ces travaux est de tester cette programmation sur un cas d'étude plus réaliste.

3 Modélisation réaliste de l'environnement

Afin de s'approcher d'un type de surface complexe tel que ceux représentant l'*impeller* illustré en introduction, plusieurs étapes d'étude vont être dissociées. Tout d'abord le passage à une surface complexe considérée comme obstacle et de même, l'obtention d'une trajectoire issue d'une surface complexe. Ces dernières sont définies à l'aide du langage APT présenté par la suite. Cette partie est positionnée dans la détermination du modèle environnement visible sur le diagramme d'activité représentant la méthode **Geo5XField** fournie en figure 2.3.

3.1 Rappel langage APT

Le langage APT permet de modéliser l'outil utilisé, mais également l'environnement complet [APT, 1983]. En effet, il est possible de générer une trajectoire outil en fonction de trois surfaces distinctes. La figure 3.9 représente la géométrie obtenue par utilisation de la méthode APT.

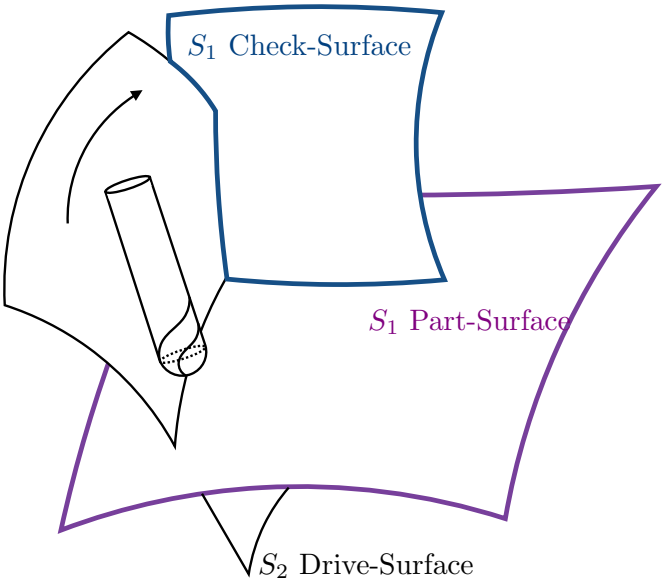


Figure 3.9 – Environnement et posage outil par la méthode APT [APT, 1983]

Il est alors possible de voir que pour chaque trajectoire d'usinage programmée, un triptyque de surfaces est modélisé :

— *Part-Surface* : surface de la pièce avec laquelle le point de coupe C_C est en contact permanent lors d'un usinage donné
— *Drive-Surface* : surface d'entraînement guidant l'outil à travers l'espace,
— *Check-Surface* : surface de contrôle représentant une surface limite de la déclaration de mouvement donnée.

Lors des travaux réalisés durant cette thèse, la *Drive-Surface* n'est pas utilisée telle quelle car la *Part-Surface* est modélisée comme une surface paramétrique. De ce fait, les trajectoires

sont générées selon des isoparamétriques pour des valeurs constantes associées à l'un des deux paramètres (u,v) de la surface. La surface d'arrêt joue le rôle d'obstacle infranchissable pour l'outil étudié.

Ce rappel permet donc de comprendre comment est défini ce qui est appelé par la suite, *Part-Surface* et *Check-Surface*. Afin de travailler principalement sur l'influence de la modélisation des *Check-Surfaces*, la première étape des travaux présentés considère une trajectoire programmée du point C_L linéaire. Cela permet d'attribuer les résultats uniquement à la modélisation de la surface obstacle concernée.

Cette dernière est modélisée comme étant une surface paramétrique de Bézier permettant de représenter un grand nombre de surfaces complexes. La formulation théorique de ce type de surface est mentionnée en annexe 2. Bien évidemment, il est important de se rappeler que les obstacles sont définis comme étant des points obstacles notés O_i. De ce fait, après avoir représenté la surface obstacle concernée, l'étude porte sur le maillage utilisé.

3.2 Maillage structuré

La méthode **Geo5XField** s'appuie sur la génération d'un potentiel et d'un champ répulsif dans un voisinage sphérique autour d'un point obstacle. Ces points représentent le maillage structuré de la surface de Bézier utilisée durant cette partie qui est construite à l'aide des points de contrôles donnés en annexe 2 (coordonnées des points P_{ij}) et illustrée en figure 3.10.

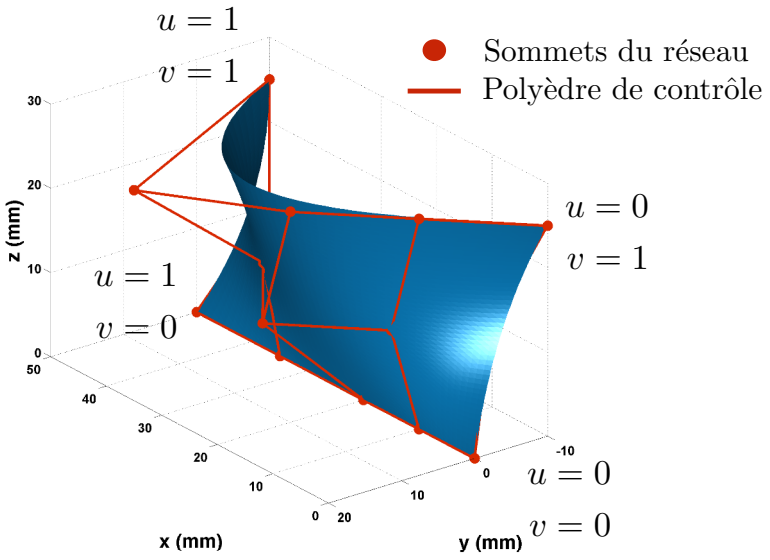


Figure 3.10 – Modélisation de la surface considérée comme obstacle lors de cette étude (*Check-Surface*)

Après avoir représenté la surface obstacle, il est important d'attribuer un maillage à cette dernière. Les points issus de ce maillage sont considérés comme obstacles et appliquent un effort

répulsif au point G de l'axe outil étudié. La mise en place de la simulation est illustrée en figure 3.11. Comme précédemment, le voisinage de chacun des points obstacles n'est pas représenté afin de ne pas surcharger la figure.

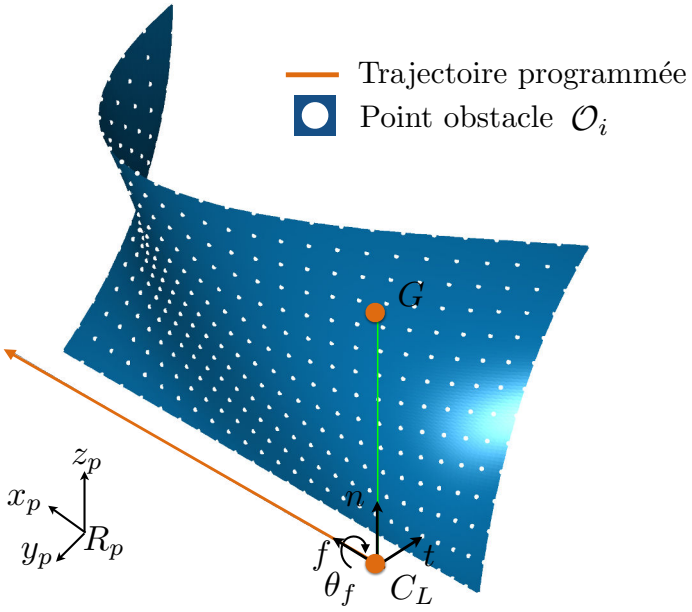


Figure 3.11 – Mise en place de la simulation

Afin de comprendre l'effet du choix de la distance entre chaque point du maillage, la programmation a été réalisée sur la surface complexe, base de l'étude durant cette partie. Concrètement, tous les paramètres, hormis la distance entre points du maillage, restent les mêmes afin de ne pas avoir de perturbations. Dans un premier temps, les points du maillage sont distants d'approximativement 3*mm* et le résultat obtenu est illustré en figure 3.12.

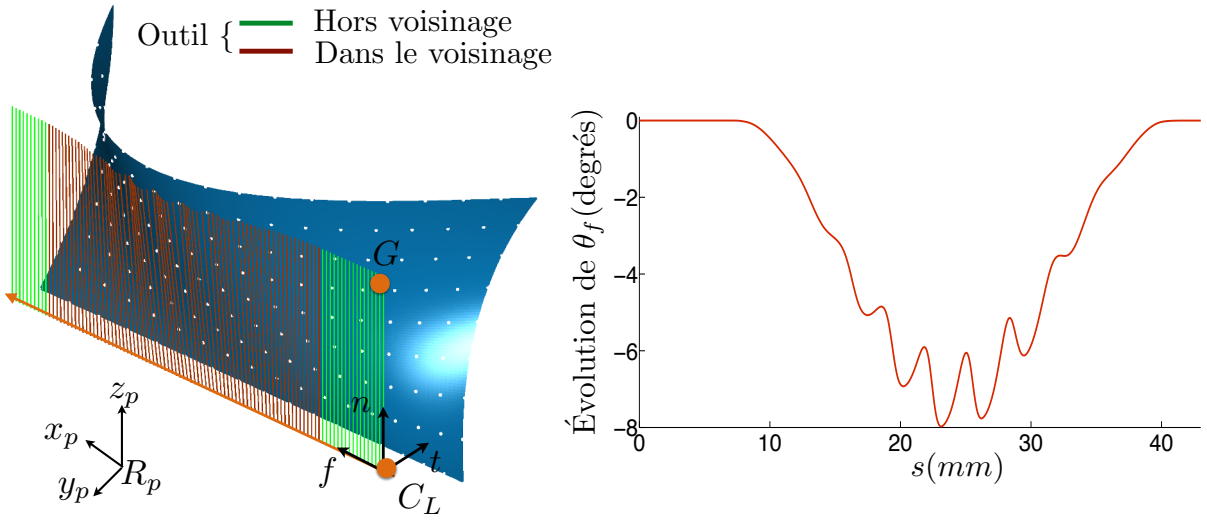


Figure 3.12 – Vision pratique : distance entre points obstacles = 3*mm*

Dans ce cas, les perturbations dues à l'éloignement des points considérés comme obstacles sont importantes. L'outil étant pour l'instant modélisé par un point dont la position est modifiée, le point G, il est alors possible pour un maillage encore plus large que celui présenté ci-dessus que ce point d'étude pénètre dans l'obstacle. Afin de valider l'étude théorique réalisée sur des lignes de points, la seconde simulation est alors effectuée avec une distance inter obstacles d'approximativement $1mm$. Le résultat illustré en figure 3.13 montre que, à l'aide d'un maillage fournissant une pulsation forcée supérieure à la pulsation propre du système, les oscillations sont très atténuées et cela permet d'obtenir une fluidité de trajectoire beaucoup plus importante et indispensable dans le cadre de l'usinage 5 axes.

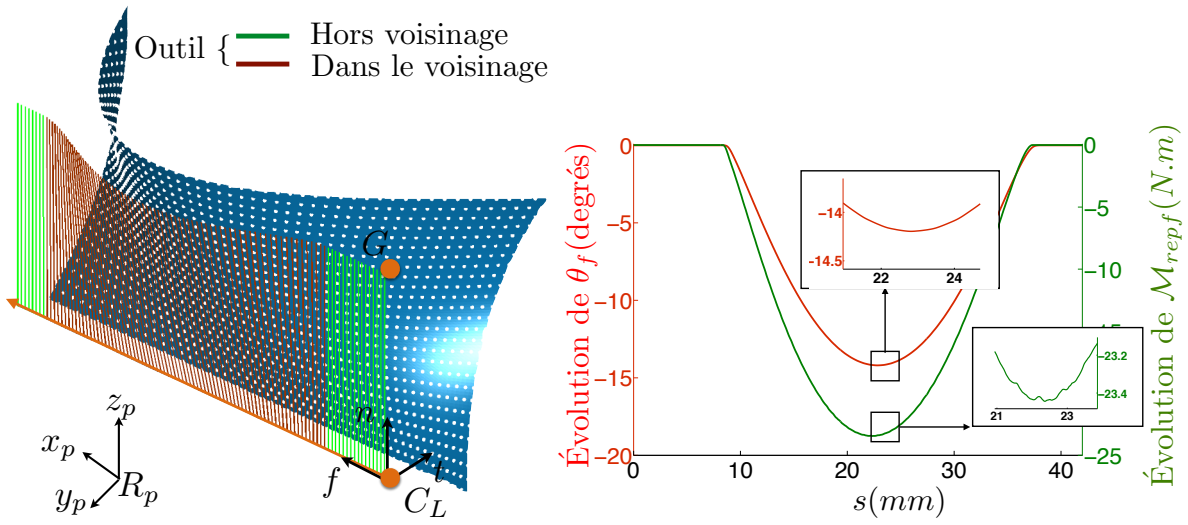


Figure 3.13 – Vision pratique : distance entre points obstacles = $1mm$

Si les discontinuités restantes se révèlent trop importantes pour l'utilisateur ou bien que le risque de pénétration entre deux lignes d'obstacles est trop grand, il serait alors envisageable de modifier le type de maillage utilisé afin d'obtenir des nœuds plus désordonnés. La figure 3.14 illustre le type de maillage non structuré qui pourrait :

— diminuer encore la présence d'oscillations sur les efforts,

— avec un maillage moins dense, obtenir une même oscillation résiduelle

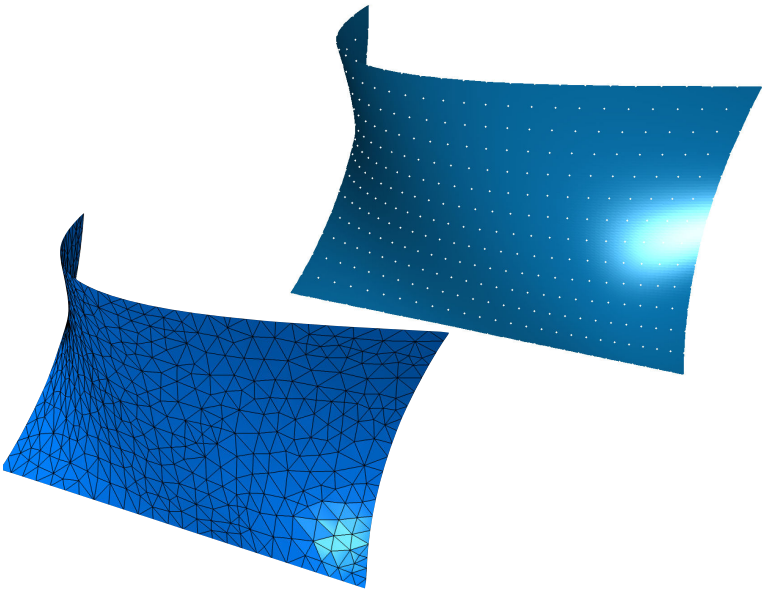


Figure 3.14 – Exemple de passage d'un maillage structuré à un non structuré

Encore une fois, il est également intéressant de relever sur la figure 3.13 l'aspect dynamique du problème mis en place. En effet, la réponse de réorientation de l'axe outil a un léger retard par rapport à l'apparition de l'effort répulsif et une meilleure continuité. Il est primordial de noter qu'avec l'aspect dynamique du problème, les perturbations visibles sur les valeurs d'effort répulsif ont une incidence plus faible sur l'étude du mouvement. Le système peut alors être comparé à un filtre naturel des faibles oscillations des actions répulsives reçues par le point étudié.

Bien que la programmation présentée précédemment fonctionne sur le cas d'une pièce complexe, il faut relever un point important. En regardant, la valeur des orientations (en degrés) des angles obtenus, le premier cas déplacent le point G jusqu'à environ 8° en figure 3.12 alors que dans l'autre cas, il atteint un angle de 15°en figure 3.13. Cela vient de l'expression de l'effort répulsif donnée précédemment en équation (3.1). Lors du passage d'une étude sur un point à un maillage comportant plusieurs points, l'effort répulsif obtenu sur le point G est en fait la somme des efforts répulsifs élémentaires obtenus depuis tous les points du maillage (même ceux dont le voisinage n'englobe pas G qui émettent de ce fait un effort répulsif nul).

Par la suite, il est alors présenté plusieurs méthodes permettant de répondre à la problématique concernant le lien entre la densité de points représentant le maillage de l'obstacle concerné et la formulation de l'effort répulsif global.

3.3 Proposition de modifications du modèle d'effort répulsif

Il semble important pour l'utilisateur que le maillage choisi pour représenter la *Check-Surface* n'ait pas forcément d'influence sur la valeur de l'effort répulsif appliqué en zone de voisinage sur l'outil. Afin de résoudre cette problématique, l'expression de l'effort répulsif n'est pas remise en

cause, le travail porte donc sur l'assemblage des efforts générés par tous les points de la surface obstacle concernée.

Afin de tester différentes expressions de l'ensemble des efforts répulsifs émanant des points représentant le maillage de la *Check-Surface*, trois méthodes sont définies et sont comparées sur trois cas d'étude concernant la surface de type bosse présentée précédemment afin de comprendre les avantages et inconvénients de chacune d'entre elles. La figure 3.15 illustre sur un cas simplifié avec cinq points obstacles, dont trois actifs, c'est-à-dire que leur voisinage englobe le point G étudié, la génération des trois modèles d'efforts répulsifs présentés par la suite. De plus l'ensemble des résultats est donné dans le tableau 3.16.

F_1 : Somme des efforts répulsifs dans le voisinage

Cette méthode est celle utilisée précédemment, dans ce cas, l'effort répulsif obtenu au niveau du point G est donné en équation (3.15), avec n_{actif} le nombre de points du maillage de l'obstacle dont le voisinage englobe le point étudié de l'outil. Avec la prise en compte réaliste de l'ensemble des points obstacles concernés, cette méthode semble être la plus cohérente.

$$F_1 = \boldsymbol{\mathcal{F}}_{rep} = \sum_{i=1}^{n_{\text{actif}}} \boldsymbol{\mathcal{F}}_{rep_i} \tag{3.15}$$

La réponse est alors fluide comme illustré en tableau 3.16 dans les trois cas d'étude. Cependant, il est également visible que sur les trois essais utilisant l'effort répulsif F_1, l'outil ne réalise pas le même évitement concernant pourtant le même obstacle. Si l'utilisateur souhaite modifier le nombre de points composants le maillage de la surface considérée comme obstacle lors de l'usinage 5 axes programmé, cela imposerait alors une étude du coefficient répulsif η permettant un évitement acceptable (en terme de modifications d'orientation obtenues).

F_2 : Moyenne des efforts répulsifs dans le voisinage

Une solution permettant de dissocier le problème du nombre de points représentants l'obstacle serait alors la génération d'une moyenne à partir des points du maillage de l'obstacle dont le voisinage englobe le point G. Cela donnerait une valeur d'effort répulsif indépendante du nombre de points. L'effort répulsif est alors obtenu comme noté en équation (3.16).

$$F_2 = \boldsymbol{\mathcal{F}}_{rep} = \frac{\sum_{i=1}^{n_{\text{actif}}} \boldsymbol{\mathcal{F}}_{rep_i}}{n_{\text{actif}}} \tag{3.16}$$

Cette méthode de calcul assure un ordre de grandeur constant même après modification du nombre de points dans le voisinage comme cela est visible en tableau 3.16. Cependant cette méthodologie octroie trop d'importance aux points les plus éloignés dans le voisinage. En effet,

la direction entre un point très éloigné et le point G serait prise en compte dans le calcul de celle de l'effort répulsif avec la même importance qu'un point plus proche. Cela amène un niveau de perturbation important.

F_3 : Homogénéisation des efforts répulsifs dans le voisinage

Afin de conserver le fait que les points obstacles, en fonction de leur distance avec le point G étudié, n'ont pas tous le même effet, la méthode proposée est alors de conserver uniquement la valeur maximale des normes comme norme du vecteur final et de réaliser la somme des directions. Cette représentation est donnée en équation (3.17) et les résultats obtenus sont fournis en figure 3.16.

$$F_3 = \mathcal{F}_{rep} = max||\mathcal{F}_{rep_i}||.\frac{\sum_{i=1}^{n_{actif}} \mathcal{F}_{rep_i}}{||\sum_{i=1}^{n_{actif}} \mathcal{F}_{rep_i}||} \tag{3.17}$$

Il est alors possible de relever que dans ce cas, le changement de point émettant la norme de l'effort répulsif maximum retenue à chaque pas de calcul est visible. Par contre, encore une fois, la valeur obtenue reste homogène, quel que soit le maillage utilisé.

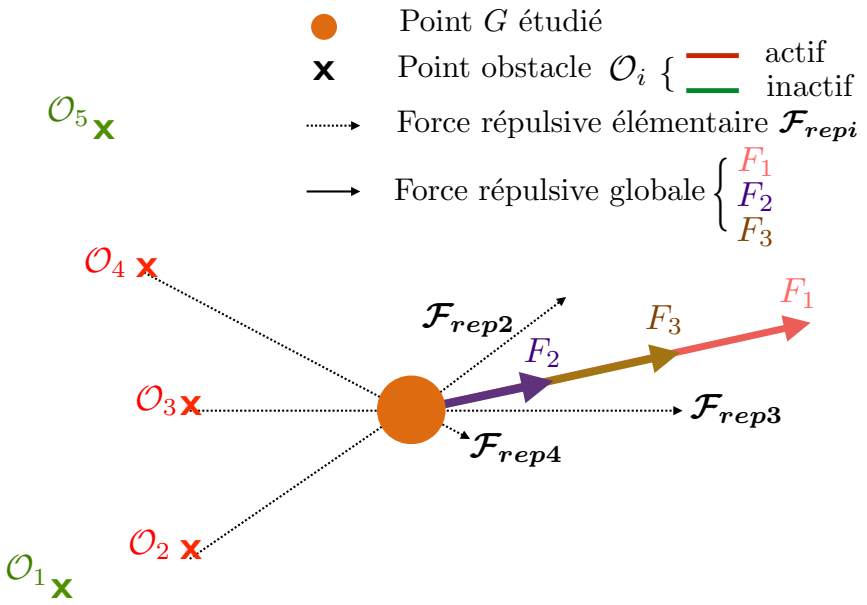


Figure 3.15 – Représentation des trois méthodes de calcul d'efforts répulsifs

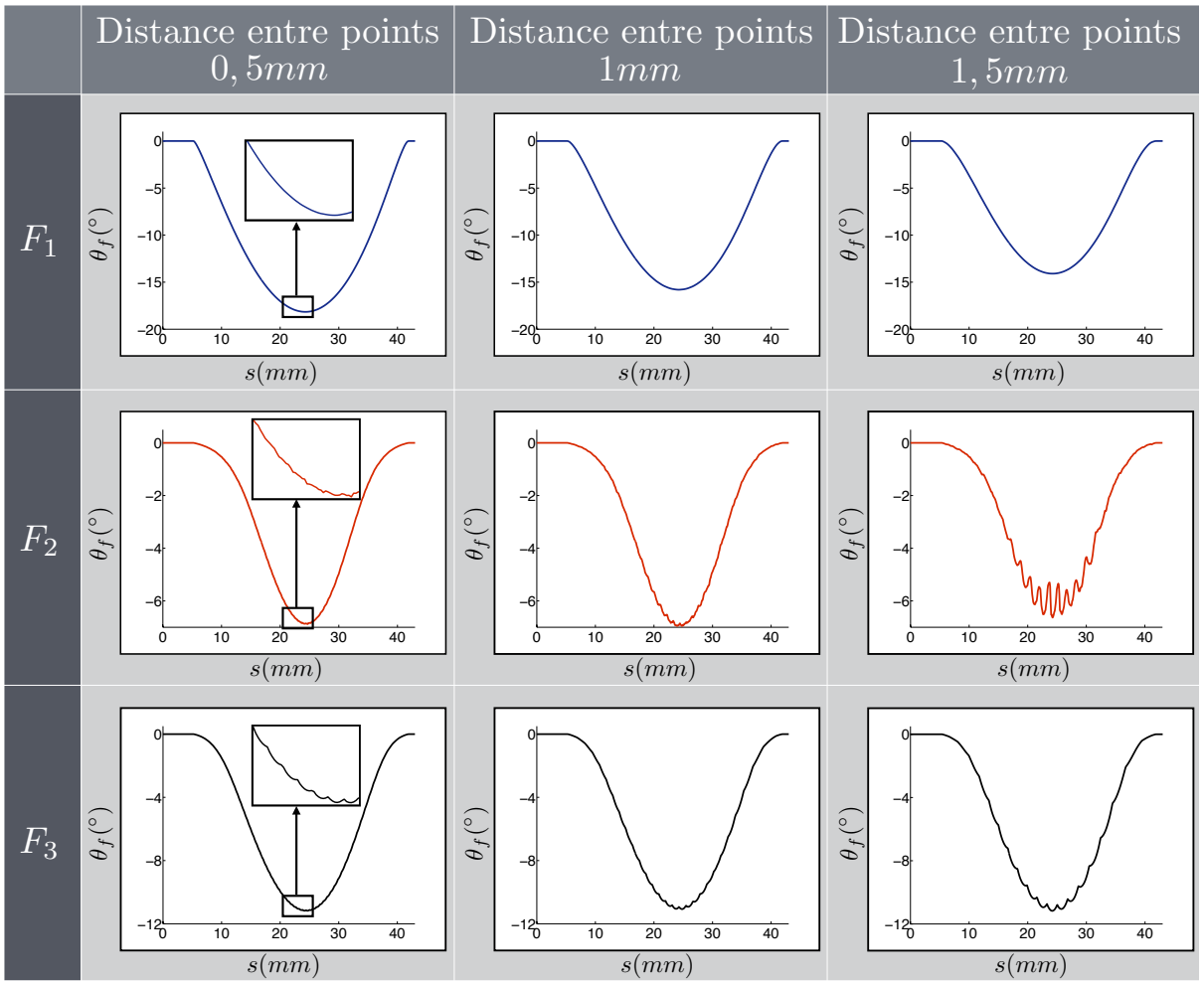


Figure 3.16 – Plan d'expérience

Pour conclure cette étude concernant les différents traitements d'un ensemble d'efforts répulsifs, il est important de rappeler que si l'intention première reste de certifier une trajectoire corrigée la plus lisse possible, il reste donc avantageux de fixer (c'est-à-dire non modifiable par l'utilisateur) la distance approximative entre les points du maillage représentant la surface obstacle afin de conserver la première méthode réalisant la somme de tous les efforts répulsifs reçus au niveau du point étudié. Il pourrait être envisagé, à la mise à disposition de l'utilisateur, un curseur permettant le choix du maillage retenu venant modifier les autres paramètres.

Afin d'appliquer les résultats obtenus dans cette partie portant sur le maillage des différentes surfaces obstacles, il parait important de tester la programmation réalisée sur une pièce industrielle de type *impeller*. Par la suite, l'écriture du paramètre $\mathcal{F}_{rep}$ représente alors la somme des efforts répulsifs au niveau du point outil étudié (F_1 (3.15)).

3.4 Simulation sur pièce industrielle de type *Impeller*

Dans le but de valider la programmation mise en place lors de cette partie, un test sur une pièce industrielle réelle est effectué. Un *impeller* étant composé d'une répétition symétrique d'un modèle initial, les travaux présentés ci-dessous portent uniquement sur le modèle décrit en figure 3.17 qui représente une décomposition en trois surfaces principales de la partie de l'*impeller* étudié illustré en figure 3.1.

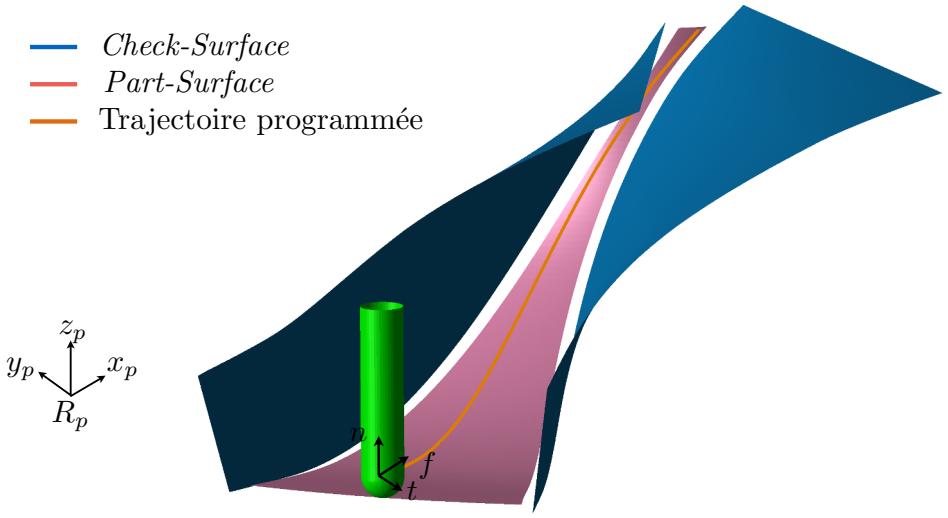


Figure 3.17 – Décomposition de l'*impeller* étudié

Dans ce cas, il y a deux *Check-Surfaces* représentant les parties obstacles du modèle et une *Part-Surface* à usiner permettant l'obtention des courbes guides du point C_L. Pour ce qui est des valeurs des constantes retenues pour cette simulation, elles sont les mêmes que celles données précédemment afin de ne pas empiéter sur l'analyse du résultat obtenu.

Cette étude porte sur la simulation effectuée avec un outil modélisé par son point G (mais représenté comme un outil cylindrique pour obtenir une illustration réaliste de la simulation) et pour une seule courbe guide de type B-spline avec dans le cas de l'étude $v = 0{,}5$ afin d'octroyer plus de visibilité aux figures illustrant le résultat obtenu. Afin de comprendre le besoin, sur cette surface, d'avoir une gestion des collisions, la trajectoire initiale sans prise en compte des *Check-Surfaces* est représentée en figure 3.18.

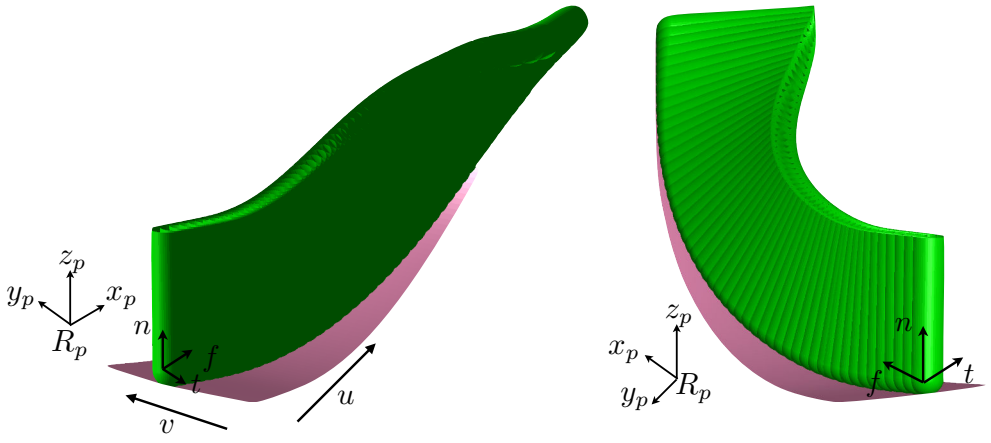


Figure 3.18 – Trajectoire calculée sans prise en compte des obstacles

Cette trajectoire est obtenue avec un axe outil dont les angles θ_{tprog} et θ_{fprog} programmés sont nuls. C'est pour cela, que l'axe outil suit la normale à la surface au point d'étude concerné. Si, sur la figure obtenue les *Check-Surfaces* sont rajoutées, il est possible de remarquer les collisions globales présentes sur la trajectoire déterminée initialement (figure 3.19).

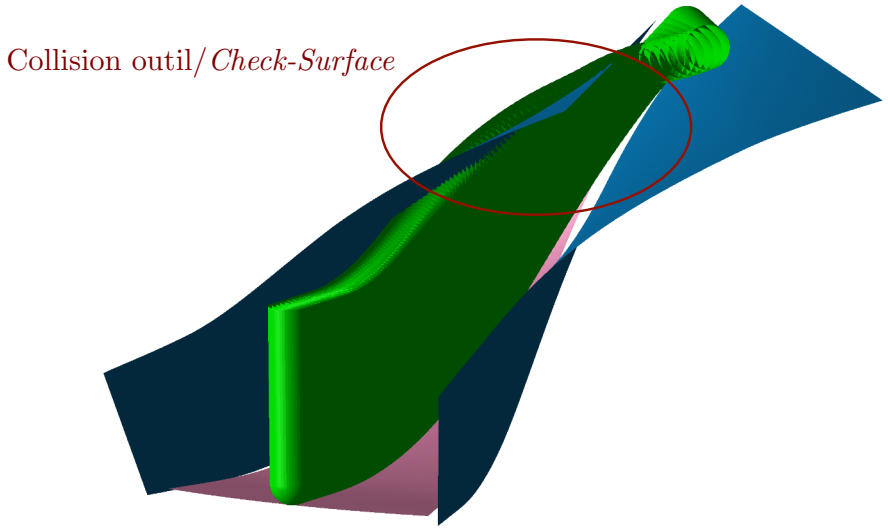


Figure 3.19 – Tracé des obstacles et vision des collisions avec la trajectoire programmée

La suite de la programmation consiste alors à définir les deux *Check-Surfaces* comme étant des obstacles répulsifs. Afin d'être cohérent, le maillage de l'obstacle est considéré comme fixé (n'évolue pas au cours de la simulation présentée) et le traitement des efforts émanant des points le représentant est obtenu avec application de la méthode notée F_1 (équation (3.15)).

Le résultat obtenu à l'aide de la programmation est illustré en figure 3.20 avec une prise en compte de la réorientation selon les angles θ_f et θ_t.

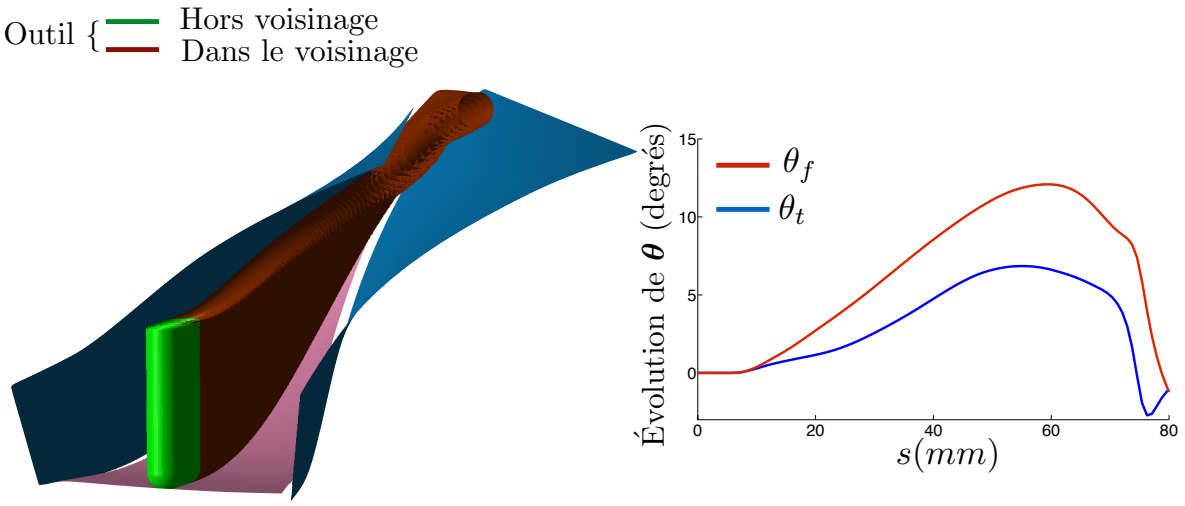


Figure 3.20 – Trajectoire calculée avec prise en compte des obstacles

L'évitement de collisions est dans ce cas réalisé avec une modification d'orientation progressive lors de l'entrée de l'outil dans la zone de voisinage. L'aspect lisse de la trajectoire corrigée est réellement présent et la possibilité de suivi d'une trajectoire définie par une courbe complexe au niveau du point C_L est alors démontré.

Une fois la partie concernant une bonne appropriation d'un ensemble de points obstacles représentant le maillage de la *Check-Surface* traitée, le point important restant à étudier est l'obtention d'une modélisation plus réaliste de l'outil qui jusqu'à maintenant n'était constitué que d'un point suivant la trajectoire programmée C_L et un point G distants d'une distance fixe h. Cela ne permet donc pas d'assurer une absence complète de collision entre l'outil réel et la surface.

4 Modélisation réaliste de l'outil

Afin de pouvoir assurer une trajectoire sans aucune collision entre l'outil utilisé et les parties de l'environnement considérées comme obstacles, il semble important de permettre d'utiliser une modélisation plus fine de l'outil. En effet, jusqu'à maintenant, seul le point G de l'outil est concerné par la réception d'efforts répulsifs. De ce fait, le cas illustré en figure 3.21 ne prendrait pas en compte la collision représentée étant donné que G n'appartient pas au voisinage.

C'est pour cela que l'axe outil reste tracé en vert, car le point étudié est à une distance du point obstacle supérieure à celle du voisinage. Une astuce qui permettrait de repousser le problème, mais sans réellement le résoudre est d'imposer un voisinage minimum correspondant à la hauteur de l'outil utilisé.

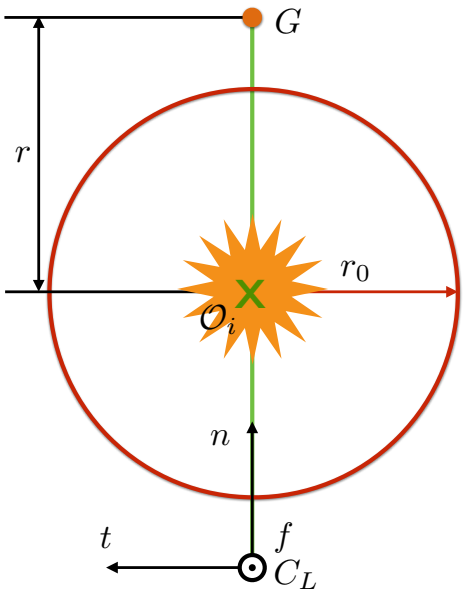


Figure 3.21 – Illustration de la problématique concernant la représentation de l'outil

Néanmoins, cette démarche de modélisation ponctuelle peut donc s'avérer trompeuse dans le cadre d'une assurance de non-collision. Afin de résoudre ce problème, la première étape des travaux présentés est l'utilisation d'un outil hémisphérique.

La figure 3.22 présente le paramétrage d'un outil hémisphérique selon le formalisme APT avec $h = L + R$, $a = \alpha = \beta = 0$, $d = 2.R$ et $b = R$.

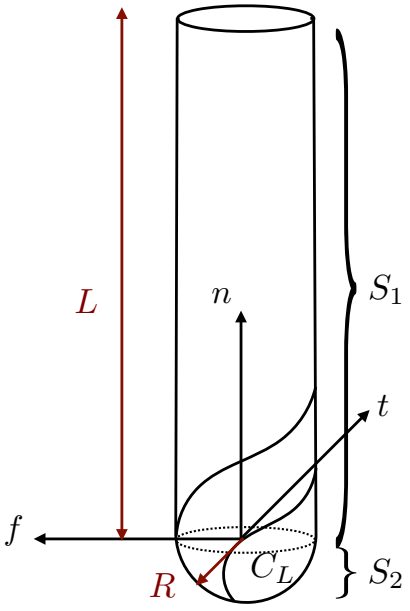


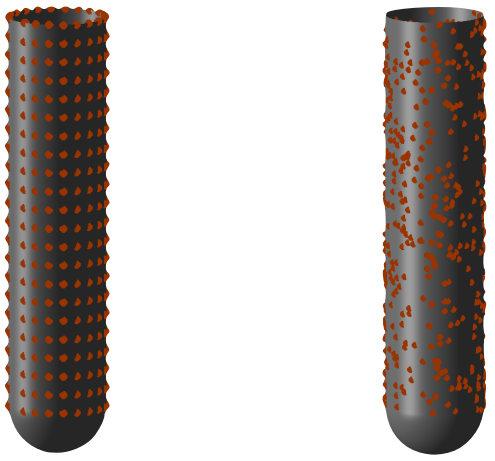
Figure 3.22 – Outil hémisphérique

L'outil est alors décomposé en deux parties implicites, le cylindre noté S_1 avec comme centre de gravité G_1 et la demi-sphère notée S_2 avec comme rayon R et comme centre le point C_L. Cette modélisation permet de ne pas prendre en compte les collisions locales touchant la partie effective de l'outil. Dans cette étude, uniquement la partie cylindrique S_1 est donc soumise aux efforts extérieurs. Cette surface cylindrique est maillée par la suite afin de pouvoir continuer d'appliquer la programmation présentée précédemment avec l'action des points des *Check-Surfaces* sur les différents points issus du maillage choisi.

4.1 Maillage de l'outil cylindrique

Il semble alors important de comprendre comment la surface représentant l'outil va être décomposée en différents points et quel rôle va jouer l'effort répulsif émanant de l'obstacle sur ces derniers obtenus après maillage de la surface obstacle considérée. L'outil de maillage utilisé cherche le plus souvent à générer une répartition ordonnée des points concernés comme illustré en figure 3.23(a).

Ce maillage peut alors amener à des zones libres (sans point du maillage) beaucoup trop ordonnées, il faudrait alors utiliser de nombreux points de représentations afin de garantir une sorte de désordre. Dans ce cas, comme pour les surfaces obstacles, il faut envisager un passage à un maillage non structuré. Pour ce faire, le programme donnant le maillage de l'outil a été amélioré et permet d'obtenir un maillage complètement aléatoire, comportant le même nombre de points et respectant le positionnement de chacun d'eux sur le cylindre modélisant l'outil utilisé. Ce résultat est donné en figure 3.23(b).



(a) Maillage ordonné

(b) Maillage désordonné

Figure 3.23 – Maillage de la surface représentant l'outil cylindrique utilisé

Ce maillage est modifiable (en terme de nombre de points, ici 500), cependant encore une fois, le moment répulsif obtenu va être modifié selon la description choisie. Dans le cas de l'outil, comme dans le cas de la surface maillée représentant l'obstacle, il est important de choisir le traitement des différents efforts répulsifs appliqués sur chacun des nœuds. Dans le cadre de la modélisation de l'outil, les points où s'appliquent les efforts répulsifs sont différents. Cette différence impose un travail sur les moments répulsifs générés réécrits au point C_L. En effet, le vecteur reliant le point C_L où sera étudié le PFD et le point du maillage de l'outil concerné P_j change pour chaque nœud.

Comme lors de l'étude de l'influence du maillage de l'obstacle, les trois méthodes décrites précédemment peuvent être appliquées avec les mêmes conclusions, de ce fait elles ne seront pas réexpliquées. Il est possible de rappeler que dans le cadre de la simulation étudiée (après avoir fixé le nombre de points représentant les maillages de l'outil noté m) la somme des moments au point C_L est donnée en équation (3.18). Bien évidemment, l'utilisation de cette méthode contraint toujours l'utilisateur à garder un maillage fixe pour l'outil utilisé afin d'assurer une certaine cohérence.

$$\boldsymbol{\mathcal{M}}_{rep} = \sum_{j=1}^{m} \boldsymbol{\mathcal{M}}_{rep_j} = \sum_{j=1}^{m} (\boldsymbol{C_L P_j} \wedge \boldsymbol{\mathcal{F}}_{rep_j}) \tag{3.18}$$

Un nouveau problème est alors soulevé. Dans le cadre d'une simulation telle que représentée en figure 3.24, l'application de cette formulation du moment répulsif global ne serait pas cohérente. Une différence entre les deux moments répulsifs est obtenue compte tenu des bras de levier.

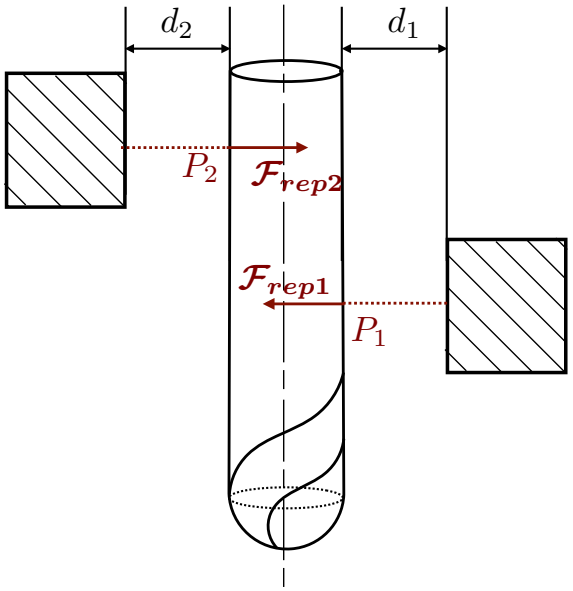


Figure 3.24 – Problématique sur la définition du moment répulsif complet

Cela impliquerait donc un mouvement de l'axe de l'outil non souhaité ou pas nécessaire. Afin d'éviter cela, il faudrait considérer l'équation obtenue un peu plus d'un point de vue machine que d'un point de vue mouvement physique. Pour ce faire, l'évolution de l'obtention du moment

répulsif serait de ne plus prendre en compte la distance entre le point étudié P_j et le point choisi pour le PFD, C_L mais uniquement la direction comme noté en équation (3.19).

$$\mathcal{M}_{rep} = \sum_{j=1}^{m} \frac{\boldsymbol{C_L P_j}}{||\boldsymbol{C_L P_j}||} \wedge \mathcal{F}_{rep_j} \tag{3.19}$$

Cette approche permet notamment de résoudre le problème pour le passage de l'outil dans des zones étroites en conservant une certaine symétrie de la trajectoire.

Une simulation est alors réalisée avec, dans un premier temps, le moment répulsif tel que défini initialement en équation (3.18) et dans un second temps avec le moment répulsif reformulé (3.19) pour un passage entre deux obstacles. Cependant, la valeur de l'effort répulsif étant modifiée, la constante η est également modifiée (d'un ordre de grandeur identique à celui de la longueur outil moyenne) afin de comparer deux simulations avec des valeurs de moments répulsifs comparables. La simulation avec le calcul du moment répulsif de base donné en équation (3.18) est illustrée en figure 3.25.

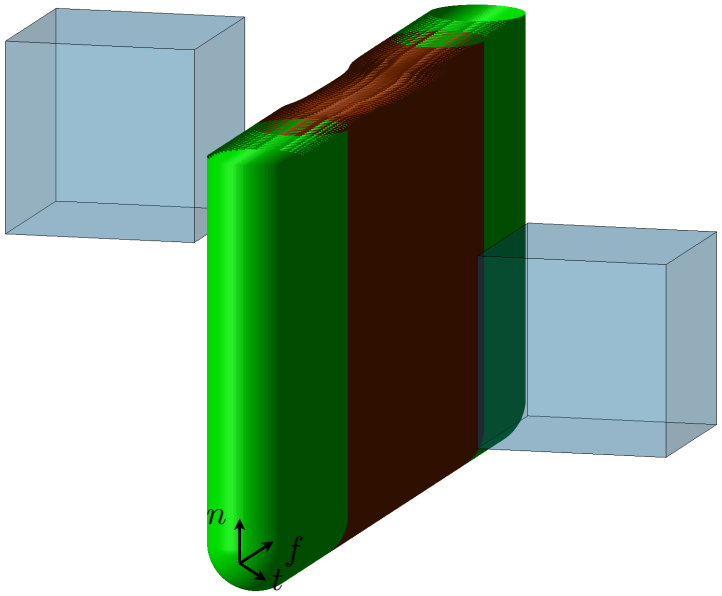


Figure 3.25 – Simulation avec la programmation de base

Il est alors possible de se rendre compte qu'il y a un mouvement, alors que les deux obstacles sont à la même distance de l'outil concerné. Ce dernier est relativement faible car les paramètres n'ont pas été modifiés mais le déplacement pourrait s'avérer plus important. Après la mise en place de la nouvelle formulation pour l'expression du moment répulsif, une nouvelle trajectoire est alors obtenue. La figure 3.26 compare la variation des angles θ_f dans les deux configurations.

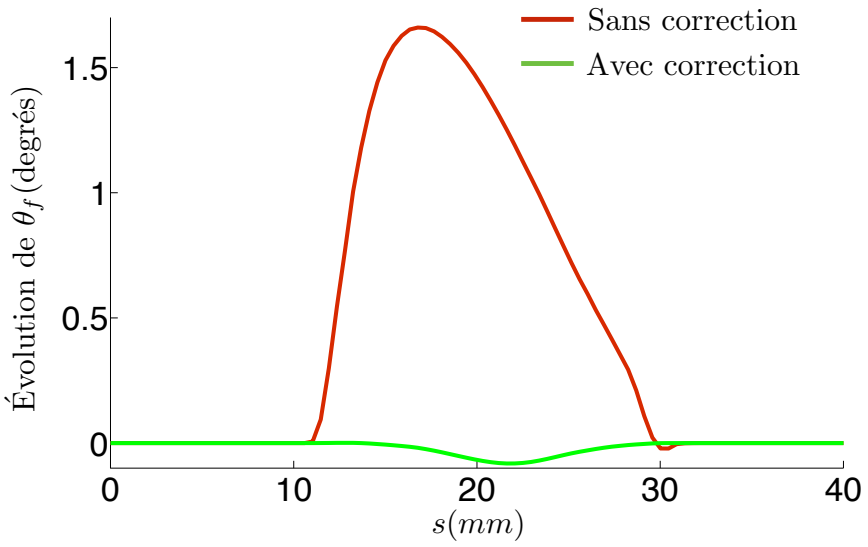


Figure 3.26 – Comparaison des modifications de θ_f obtenues

Le mouvement initial, non justifié d'un point de vue évitement d'obstacle, est atténué fortement avec la nouvelle programmation. Il est possible de remarquer qu'il reste un mouvement, car, l'outil étant maillé, les points ne sont pas répartis de manière continue.

La dernière partie de l'étude de la modélisation de l'outil porte sur la prise en compte du porte-outil si l'utilisateur a besoin d'une assurance d'absence de collision sur cet ensemble.

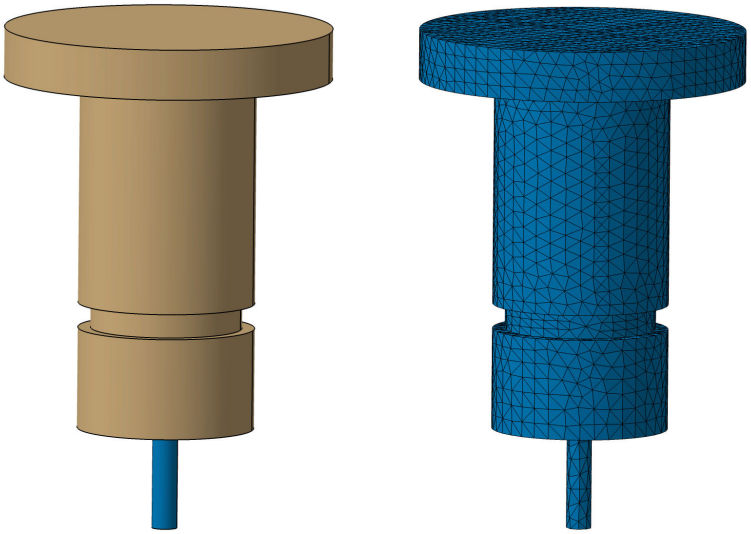
4.2 Modélisation de la peau pour l'ensemble outil/porte-outil

Afin d'être la plus complète possible et de permettre à l'utilisateur de lancer une simulation prenant en compte le maximum de risques de collisions existants, la modélisation d'un outil hémisphérique et d'un porte-outil est effectuée. Le but est alors de discrétiser les surfaces extérieures représentant l'ensemble après montage afin d'assurer une absence de collision globale. Pour ce faire il n'y aura donc pas de représentation de la demi-sphère représentant la partie coupante de l'outil. Le modèle utilisé est représenté en figure 3.27(a). Par la suite, un maillage désordonné peut alors être effectué comme illustré en figure 3.27(b). Cette figure montre le maillage plus ou moins aléatoire avec une décomposition de la surface théorique en plusieurs points (avec dans ce cas, une distance entre points approximative de 1mm).

La programmation précédente est alors utilisable avec cet ensemble outil/porte-outil et permet d'élargir la zone protégée des collisions avec les différents obstacles.

Une nouvelle simulation de type coin est réalisée afin de montrer l'impact et la programmation de la prise en compte du maillage complet ainsi que les contraintes du choix à réaliser entre un outil long et un angle de correction faible (effet canne à pêche) et un outil court (sans problème de

flexion), mais avec une obligation de réorientation et donc une modification du rayon effectif de coupe. La pièce étudiée est représentée en figure 3.28.



(a) Modélisation (b) Maillage

Figure 3.27 – Modélisation de l'ensemble outil/porte-outil sans la partie de coupe hémisphérique soumise aux collisions locales

Cette dernière comporte une trajectoire d'outil avec un virage à 90°et une surface obstacle empiétant sur la trajectoire initialement programmée. Le réglage initial est orienté avec θ_{fprog} de $-0{,}3rad$ $(-17{,}18°)$. L'empiètement de l'obstacle est d'une valeur différente sur les deux tronçons rectilignes de la trajectoire.

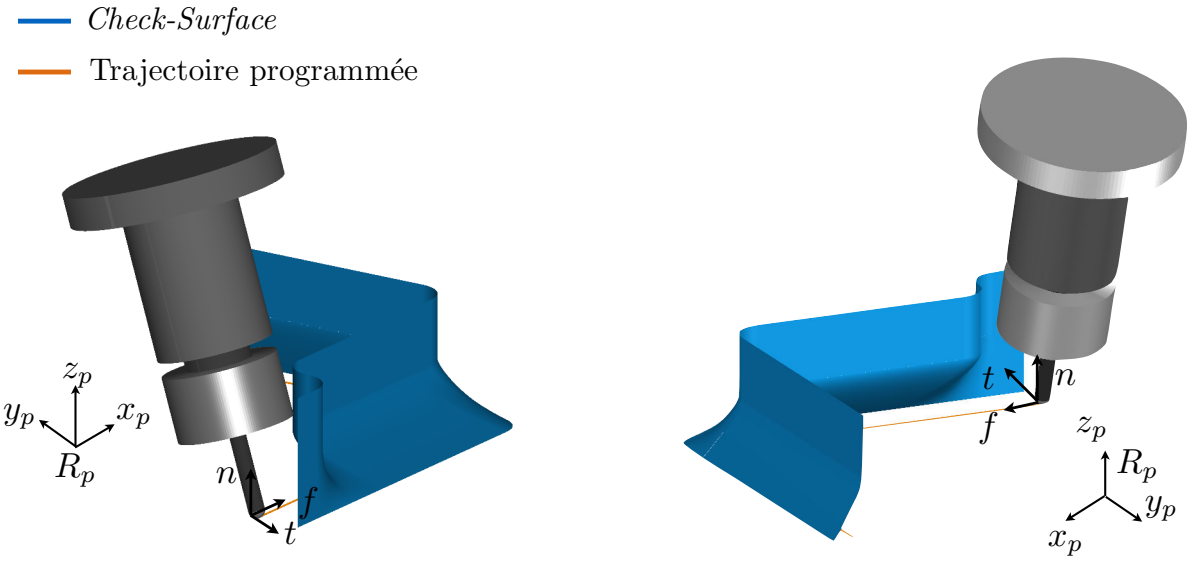


Figure 3.28 – Simulation avec outil complet

La simulation est réalisée pour deux cas, tout d'abord la longueur outil est conservé à 20*mm* et dans un second temps, cette dernière est programmée à 30*mm* en conservant les mêmes valeurs pour les autres variables. La figure 3.29 met en avant la modification de l'orientation en fonction de la longueur outil pour un point de la trajectoire. Bien évidemment, pour l'outil le plus long (en vert), la modification d'orientation est moins importante. Par contre, cela augmente la déformation due à la flexion de l'outil et les potentiels problèmes de vibration. Encore une fois ce compromis peut être choisi par l'utilisateur.

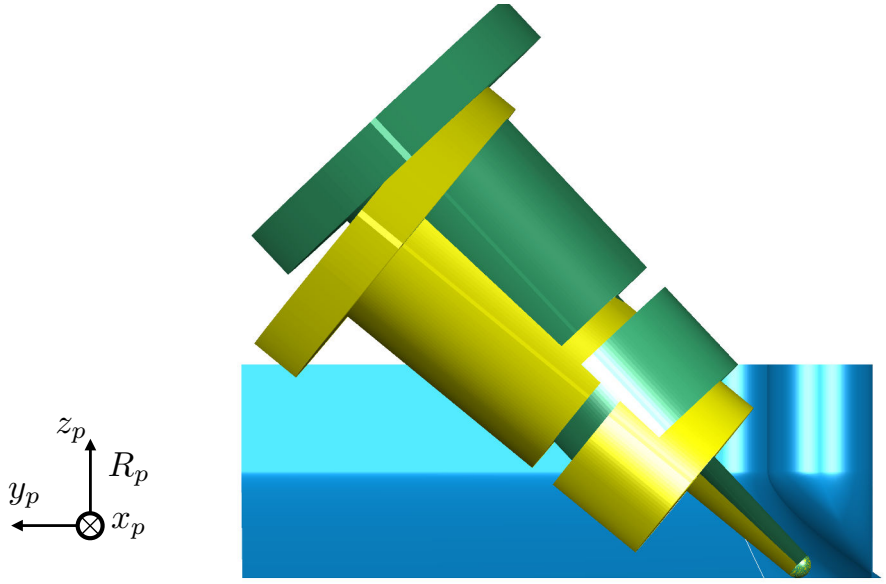


Figure 3.29 – Représentation de la différence de réorientation

La figure 3.30 représente l'évolution de $\theta_{f\,\text{corrigé}}$ et de $\theta_{t\,\text{corrigé}}$ par rapport à s pour les deux modélisations sur, cette fois-ci, l'ensemble de la trajectoire.

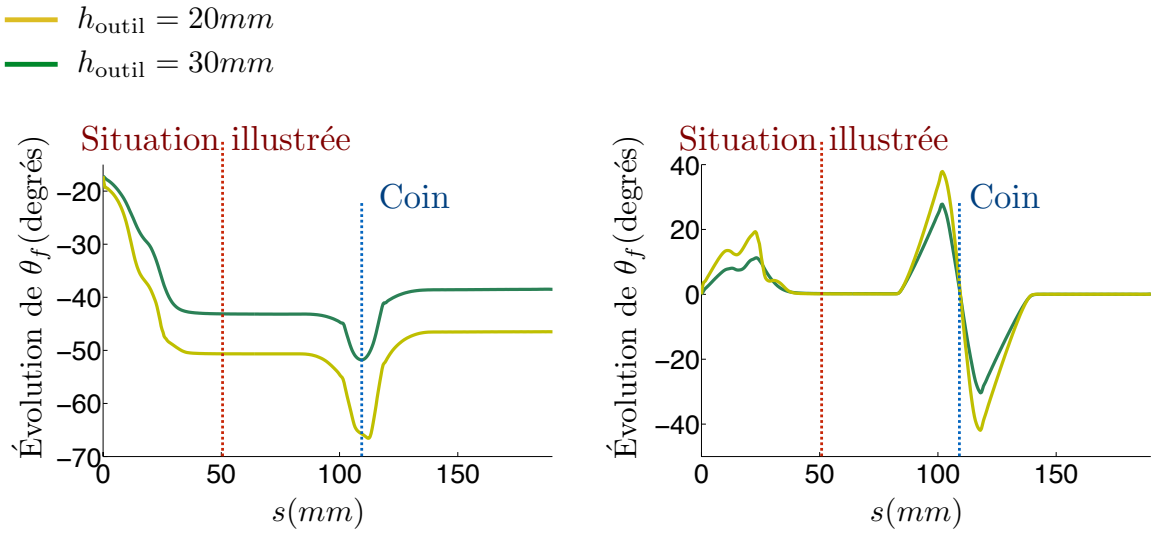


Figure 3.30 – Évolution des angles corrigés en fonction de s

Plusieurs choses sont alors à remarquer, tout d'abord, après perturbation, le retour en orientation programmée est bien effectué selon les deux directions sans oscillations. Le franchissement du coin apporte une modification d'orientation avec la prise en compte d'un effort répulsif modifiant l'orientation d'équilibre atteint le long de la première partie de la surface obstacle. Effectivement, lors de l'approche de la seconde partie de l'obstacle considéré, l'effort augmente avec la prise en compte des deux parties répulsives. Afin d'éviter cette remontée il serait possible d'utiliser F_2 ou F_3 afin de ne pas dépendre du nombre de points obstacles. Cela permet de montrer l'efficacité de la programmation dans le cadre d'un produit de type poche ouverte.

La dernière partie concernant ce chapitre va mettre en évidence un point particulier qui est le temps de calcul pour une simulation complète.

5 Étude du temps de calcul de la simulation sur pièce industrielle

Il est important de relever que pour chaque positionnement il est nécessaire de calculer les actions répulsives et de rappel exactes afin de connaître le mouvement de réorientation de l'outil. Il peut s'avérer que les temps de calcul soient alors très importants. De ce fait, la suite de l'étude présentée concerne le temps de calcul nécessaire à la programmation mise en place sur l'exemple principal de ces travaux : l'*impeller*. Afin d'être le plus clair possible, la simulation est réalisée avec les paramètres cités précédemment, dans le cas d'un outil modélisé par son point G ou par le maillage de ce dernier, avec une utilisation du moment répulsif uniquement autour de $\mathbf{f}$.

Le nombre réel de trajectoires est calculé en fonction de la hauteur de crêtes admissible dans le cadre de ce type d'usinage. Pour cet exemple, ce dernier est fixé à $h_c = 0{,}01mm$ avec un rayon effectif de coupe égale au rayon de l'outil utilisé $5mm$, la distance entre passe (prise de passe radiale) a_e est donc déterminée en équation (3.20), en considérant la formulation de la hauteur de crête pour une surface plane, ce qui correspond à la zone de départ, là où la surface est la plus large.

$$a_e = \sqrt{8.h_c.R_{eff}} = 0{,}63mm. \tag{3.20}$$

Afin de juger du temps de calcul pour une simulation d'usinage complète de l'*impeller* concerné, une partie des trajectoires suivies par le point piloté sont générées et illustrées en figure 3.31 avec la prise en compte d'une veine. En allant de $v = 0{,}2$ à $v = 0{,}8$, le nombre de passes total, en respectant la hauteur de crêtes souhaitée, pour une *Part-Surface* est de 60 avec un $\Delta v = 0{,}01$. Les simulations concernant des valeurs de v différentes de celles représentées en figure 3.31 ne sont pas étudiées expérimentalement mais par une interpolation linéaire fournie en annexe 3.

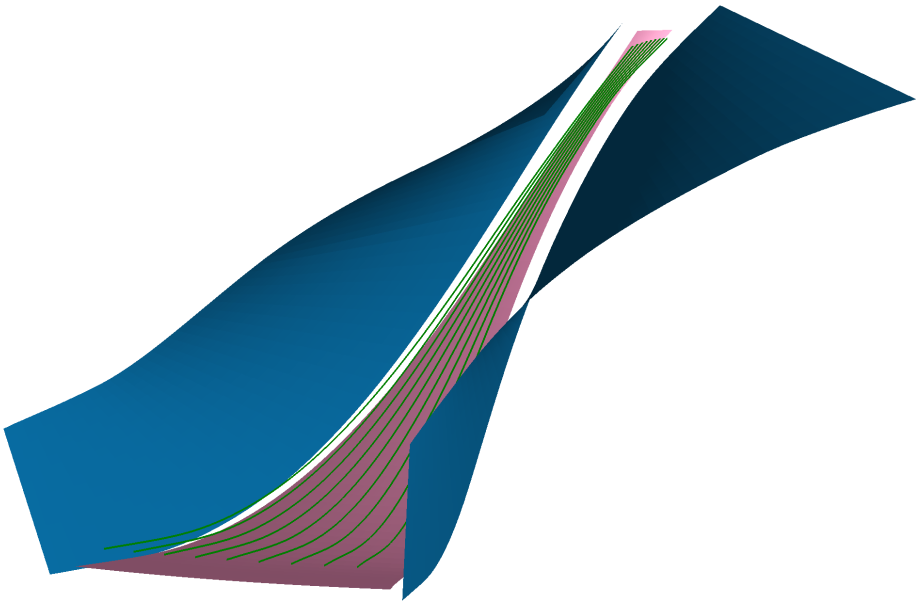


Figure 3.31 – Usinage complet de la surface guide ($v = 0{,}2 : 0{,}1 : 0{,}8$)

Une fois le temps de calcul associé à chacune des trajectoires d'usinage sur une veine, ce nombre est alors multiplié par le nombre de *Part-Surfaces* (dans le cas de cette étude 7). Lors de cette recherche temporelle, la vitesse de parcours est supposée égale à la vitesse d'avance programmée et le trajet hors usinage n'est pas pris en compte. Ainsi, le temps obtenu ci-après est réellement un minimum de la solution fournie. Les résultats ne sont pas tracés afin de ne pas encombrer ce document, mais le tableau 3.2 illustre les temps de calcul pour diverses modélisations de l'outil et pour des valeurs de v où la simulation a réellement été effectuée. Les calculs sont effectués sous Matlab® sous la forme d'un langage interprété et donc un code non optimisé, avec un ordinateur équipé d'un processeur INTEL Xeon® $CPU - ES\,1620@3.6GHz$.

La valeur donnée au niveau de la ligne notée «trajectoire complète» correspond à la somme des 60 trajectoires (celles obtenues via une interpolation linéaire) multipliée par le nombre de surface guide, ici égal à 7.

Il est alors aisé de comprendre la problématique qui peut apparaître pour un utilisateur souhaitant travailler avec une modélisation complète des éléments constituant l'environnement et l'outil. En effet, les temps de calcul deviennent très importants et rendent la programmation inutilisable dans le cadre de l'implantation dans un logiciel professionnel.

Dans le dessein d'obtenir un gain en temps de calcul, il semble important de proposer une méthode d'approximation permettant aux utilisateurs exigeants d'aboutir à un calcul de trajectoire réduit sans trop altérer l'exactitude de la correction apportée. Cela n'est pas utile pour la plupart des simulations car, en configurant un paramétrage des constantes suffisamment élevé, la plupart des cas pourraient être traités avec une modélisation simplifiée de l'outil.

Temps de calcul	sur le point G	sur le maillage outil
v = 0,2	5,1s	586,02s
v = 0,3	4,4s	412,27s
v = 0,4	3,9s	341,01s
v = 0,5	3,8s	348,31s
v = 0,6	3,6s	430,98s
v = 0,7	3,2s	581,01s
v = 0,8	2,9s	696,69s
Trajectoire complète $v = (0{,}2 : 0{,}01 : 0{,}8) * 7$	27min	54h48min

Tableau 3.2 – Étude du temps de calcul selon la modélisation de l'outil

6 Conclusions et perspectives

Une application de la méthode développée durant le chapitre 2 à une configuration d'usinage réaliste était l'objectif principal de ce chapitre. Dans un premier temps, la continuité de l'étude théorique concernant cette fois-ci le régime permanent du système a été réalisée via l'étude d'une succession de points représentant plusieurs lignes de maillage. Les résultats obtenus illustrent bien que le nombre de points notés O_i émettant des efforts répulsifs et leur disposition a un impact sur la trajectoire obtenue. L'étude du second ordre avec second membre est donc effectuée avec la description de la solution permanente, permettant par le théorème de superposition et avec la solution transitoire, d'obtenir la solution complète à l'étude effectuée.

Par la suite une modélisation réaliste de l'environnement utilisé avec les impacts des différents paramètres sur la résolution du problème est traitée. Cette partie débute par une présentation de la surface complexe utilisée avec une différentiation entre les surfaces guides *Part-Surface* et les surfaces obstacles *Check-Surface*. La suite des travaux présentés au cours de cette partie concerne, dans un premier temps, l'impact du maillage réalisé sur les surfaces obstacles et dans un second temps les différentes méthodes de prise en compte des efforts pour obtenir l'effort répulsif global à chaque instant. Enfin, une application industrielle valide la programmation et l'attribution des constantes proposée.

Puis, l'étude porte sur la modélisation de l'outil. En effet, jusque là, l'outil était représenté par le point C_L suivant la trajectoire programmée et le point G soumis aux divers efforts. Cette modélisation a permis de mettre en place la programmation, mais il parait réaliste que pour assurer une absence de collision il soit utile de prendre en compte l'outil complet, ou au moins une représentation. Pour ce faire un maillage de l'outil hémisphérique est traité avec l'apport d'un désordre permettant de ne pas influer sur la réponse dynamique du système. Enfin, la dernière

étape est la proposition d'un maillage complet de l'ensemble outil/porte-outil permettant, pour le cas d'usinage avec beaucoup de zones de collisions probables, une assurance complète d'absence de collision.

Il semble nécessaire de proposer, pour les simulations nécessitant une prise en compte d'un outil finement maillé, une approximation permettant de réduire fortement les temps de calcul sans pour autant avoir une différence trop importante par rapport au résultat corrigé. Concrètement, la méthode qui est développée durant le dernier chapitre de cette thèse concerne la présentation d'une méthode de partitionnement de l'espace dite de voxelisation. Cette dernière permettra, avec différentes configurations possibles, de pré calculer la valeur de l'effort répulsif afin de réduire le temps de calcul. Les écarts obtenus seront alors étudiés afin d'assurer un réalisme à la nouvelle trajectoire obtenue.

Chapitre 4

Pré calcul de l'action répulsive

Ce quatrième chapitre permet d'étudier une programmation permettant de diminuer les temps de calcul avec une prise en compte d'un outil complet à l'aide d'une interpolation des valeurs d'efforts répulsifs. Dans ce but, une première partie concernant la méthode de partitionnement de l'espace retenue dite de voxelisation de l'espace pour l'évaluation du champ d'effort est détaillée. Il reste cependant important, par la suite, de vérifier la correction d'orientation obtenue par rapport à celle fournie par une modélisation continue comme présentée à la fin du chapitre 3. Enfin, le gain en terme de temps de calcul par rapport à la méthode exacte est analysé.

Sommaire

1 Introduction 117

2 Voxelisation et influence sur la trajectoire obtenue 121

2.1 Paramétrage de la voxelisation 121

2.2 Influence de la méthode d'interpolation retenue 124

2.3 Influence de la taille de voxelisation sur la génération de trajectoires 125

2.4 Augmentation de l'admissibilité de la taille des voxels 129

3 Influence de la voxelisation la simulation d'usinage de l'*impeller* 130

3.1 Analyse de l'écart géométrique obtenue avec la voxelisation 130

3.2 Analyse de la complexité temporelle obtenue avec la voxelisation 132

4 Conclusions et perspectives 134

1 Introduction

La simulation réalisée précédemment sur l'*impeller* a montré son efficacité en usinage 5 axes avec évitement de collisions globales. Cependant dans le cadre d'une utilisation avancée du logiciel, il peut s'avérer utile que la simulation soit complète en ce qui concerne l'obstacle représenté, mais également l'outil modélisé. Cela nécessite donc une prise en compte de l'outil maillé complet, avec un nombre de points importants représentant ce dernier. Cette démarche entraîne alors, comme illustré au cours du chapitre 3, une augmentation significative de la durée des temps de calcul. Dans le dessein de déterminer les efforts agissant sur chaque point du maillage outil, il est nécessaire d'évaluer les distances de chaque point outil, à chaque point obstacle, et ce, pour chaque pas de calcul de trajectoire. Le nombre de calculs nécessaires est directement lié à la modélisation retenue pour chaque élément, menant à la complexité suivante :

$$\text{Complexité}_{\text{Calcul direct}} = P.N.C_S \tag{4.1}$$

avec :

— P : nombre de positionnement outil pour parcourir la trajectoire,

— N : nombre de nœuds du maillage outil,

— C_S : nombre de points obstacles du maillage des *Check-Surfaces*.

Ainsi, la génération de trajets longs sur les pièces complexes usinées en bout d'outil requiert un nombre de calculs très élevé, impactant directement sur le temps de calcul. Afin d'assurer l'absence de collision, il pourrait s'avérer utile de diminuer encore la dimension des mailles représentant l'environnement et l'outil utilisés augmentant d'autant plus le temps de calcul.

Ce chapitre s'appuie, dans un souci de comparaison cohérent, sur l'utilisation de l'*impeller* dont la figure 4.1 rappelle, d'une part la modélisation de la pièce test dans son ensemble et la zone étudiée d'autre part. Cette dernière est constituée de deux surfaces obstacles (*Check-Surfaces*), les deux parois de la veine, ainsi qu'un fond, la surface guide de la trajectoire (*Part-Surface*).

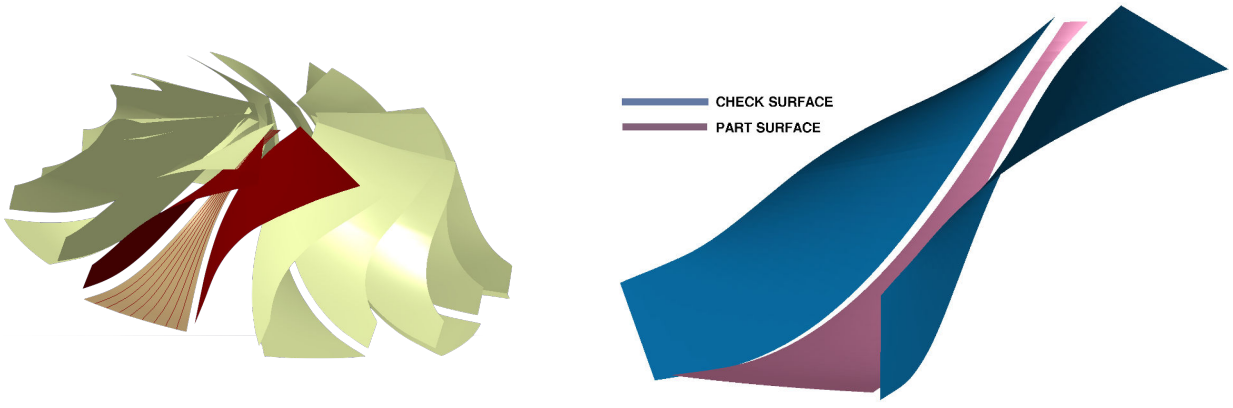


Figure 4.1 – Pièce test et détail des surfaces pour la veine étudiée

Afin de traiter la problématique concernant le temps de calcul évoquée précédemment, il est envisagé d'utiliser une méthode de partitionnement de l'espace afin de pré calculer les efforts répulsifs en certains points de ce dernier. Le modèle envisagé pour effectuer ce partitionnement est celui des voxels. La voxelisation consiste à partitionner de manière régulière un espace continu afin de représenter diverses caractéristiques de chaque sous-volume ainsi créé. Cela permet par la suite de réaliser une approximation de la solution exacte. Cette dernière, utilisée classiquement dans le domaine de la CAO, semble adaptée à la nouvelle problématique énoncée. Les informations contenues dans les voxels sont définies par une pré simulation et cette étape peut bien évidemment être réalisée de différentes manières.

Pour ce qui est de la représentation $3D$, des travaux donnent une modélisation $3D$ des voxels. Dans le cas étudié représenté en figure 4.2, l'espace 3D considéré est divisé en $NxNxN$ cubes [Kaufman et al., 1993] [Kitney et al., 1989].

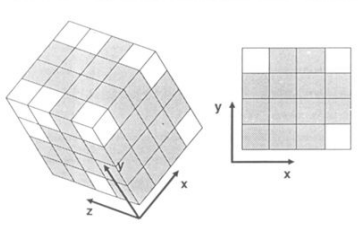


Figure 4.2 – Création des voxels [Kitney et al., 1989]

Il est alors possible, en partant d'une *Bounding Box* englobant la surface, comme illustré en figure 4.3 de déterminer les voxels permettant de représenter la surface réelle [Tarbutton et al., 2013].

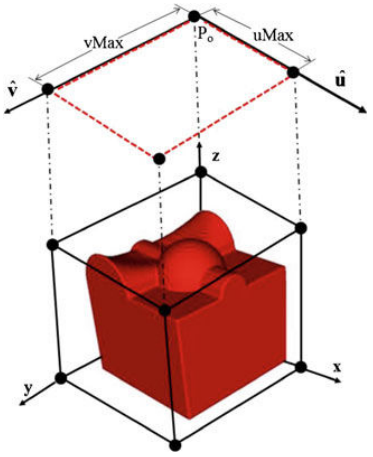
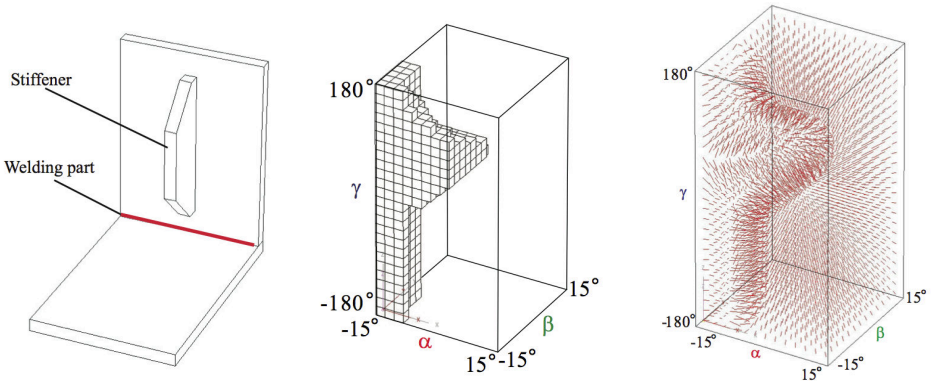


Figure 4.3 – *Bounding Box*[Tarbutton et al., 2013]

Quelques travaux dans le domaine de la fabrication (soudage) proposent une telle modélisation pour représenter les surfaces obstacles [Asakawa and Kanjo, 2013]. Ces études consistent, pour la pièce étudiée, à voxeliser le domaine de travail de la machine utilisée comme illustré en figure 4.4 et calculer l'effort répulsif en chaque coin des voxels.



(a) Pièce étudiée (b) Voxelisation du domaine actif (c) Ensemble des F_{rep} obtenus

Figure 4.4 – Méthode de voxelisation de l'environnement étudié [Asakawa and Kanjo, 2013]

De ce fait, ils obtiennent l'ensemble des efforts répulsifs sur chaque coin des voxels avant même de commencer la simulation comme représentée en figure 4.4. Dans ce cas, le calcul des efforts répulsifs est réalisé hors boucle ; en effet, lors du choix de l'utilisateur de la pièce étudiée et des coefficients choisis pour l'effort répulsif, la voxelisation reste alors la même, quel que soit l'outil utilisé, le rappel et la vitesse programmés. Par la suite, lors de la simulation, la valeur de l'effort répulsif sur le point étudié est obtenue par un calcul prenant en compte les 8 coins du voxel entourant ce point comme illustré en figure 4.5.

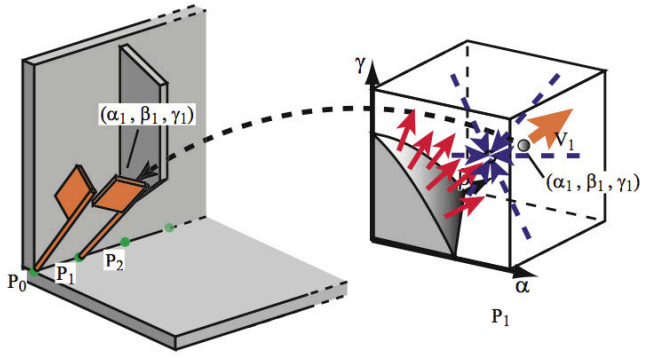


Figure 4.5 – Calcul approximé de F_{rep} sur le point étudié [Asakawa and Kanjo, 2013]

Cependant, cet article ne donne aucune explication théorique sur la méthode. D'autres auteurs ont également utilisé les méthodes de voxelisation dans le domaine de l'usinage 5 axes afin de discrétiser l'espace de travail [Jang et al., 2000] (représentation de la pièce usinée mais sans intervention de la voxelisation dans la programmation d'usinage) [Zhu and Lee, 2004a] (utilisation de dexel permettant de ne pas accorder la même importance à différentes zones de travail).

Les *octrees*, autre méthode de partitionnement de l'espace, sont une généralisation des voxels à des tailles variables, que l'on peut ainsi faire varier en fonction des caractéristiques étudiées [Ding et al., 2004]. L'octree est une structure plus simple et moins consommatrice en temps et en mémoire [Meagher, 1982]. La définition d'un octree comme étant la division régulière et récursive de l'espace est rappelée en figure 4.6 [Girardeau-Montaut, 2006].

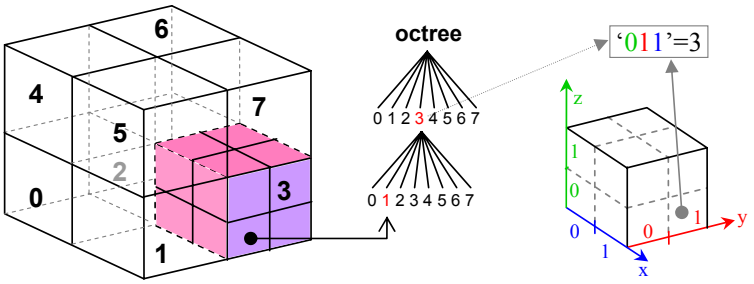


Figure 4.6 – Principe de l'octree avec codage de la position d'une sous-cellule [Girardeau-Montaut, 2006]

L'avantage est de pouvoir conserver uniquement les parties contenant un élément des pièces concernées au cours de l'étude. La figure 4.7 disponible sur le site [Nvidia, 2014] permet de comprendre l'application de cette méthode sur l'exemple classique du *Stanford Bunny* où le partitionnement est effectué avec des cubes de dimensions différentes permettant d'englober de manière rapprochée la surface considérée.

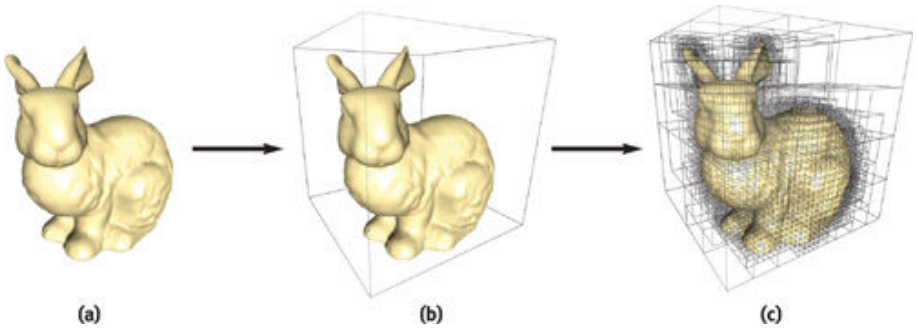


Figure 4.7 – Construction d'octree autour d'une surface maillée [Nvidia, 2014]

Dans le cas de l'étude proposée, la programmation est effectuée à l'aide de voxels et non pas d'octrees. Bien que ces derniers permettent de réduire encore les temps de calcul, il faudrait délimiter un seuil pour stopper la diminution de ces derniers. Il est alors important de rappeler que l'aspect dynamique au cours de cette étude est utilisé dans le but de supprimer les itérations géométriques de repositionnement. Se rajoute une autre problématique traitée par la suite qui montre la difficulté d'utiliser les octrees au cours de cette thèse.

Ce chapitre rappelle dans un premier temps la modélisation de l'*impeller* utilisé comme exemple au cours du chapitre précédent puis les différents principes et paramétrages d'un partitionnement de l'espace en voxels pour représenter le champ d'efforts répulsifs.

L'influence de l'approche par voxel est étudiée, tout d'abord avec une explication des méthodes d'interpolation utilisables afin d'obtenir la valeur des efforts répulsifs au cœur du voxel concerné. Puis les différences d'efforts répulsifs obtenues sur deux exemples de trajectoires simplifiés sont mises en avant. Par la suite, une application à la trajectoire d'usinage réelle afin de pouvoir comparer la trajectoire hors obstacle obtenue par voxelisation et la trajectoire exacte issue du chapitre précédent est effectuée. Il sera alors important de souligner une nouvelle problématique concernant le resserrement d'obstacles impactant l'approximation de l'effort calculée.

Enfin, l'étude de la nouvelle complexité obtenue en comparaison avec la méthode continue en équation (4.1) est effectuée avec une analyse du gain de performance d'un point de vue temps de calcul.

L'application de cette méthode à la pièce de référence avec les trajectoires d'usinage de finition programmée précédemment permet de conclure sur les forces et faiblesses de cette méthode.

2 Voxelisation et influence sur la trajectoire obtenue

Pour ce faire, cette partie concernant l'influence du traitement de la méthode de voxelisation retenue est séparée en cinq points principaux. Tout d'abord il est important de décrire le paramétrage de la voxelisation retenue puis d'expliciter les interpolations disponibles ainsi que leurs effets sur la trajectoire hors collisions obtenue. Par la suite, la problématique concernant le rapprochement d'obstacles est traitée et enfin l'application à la trajectoire réelle d'usinage de l'*impeller* est alors effectuée avec la mise en avant d'une nouvelle problématique.

2.1 Paramétrage de la voxelisation

Pour une opération d'usinage considérée, le volume de la pièce à partitionner doit englober l'espace dans lequel peut évoluer l'outil sans être en collision. En première approche, il est possible de définir un parallélépipède rectangle, dont la taille est un multiple de celle des voxels, englobant toutes les surfaces ainsi que l'espace atteignable par l'outil dont le point de contact évolue sur la surface guide. Une illustration grossière est donnée en figure 4.8 avec des cubes de dimensions 10*mm* afin de permettre une visualisation acceptable. Dans la suite de cette étude, les dimensions des voxels varieront de 2 à 0,5*mm*.

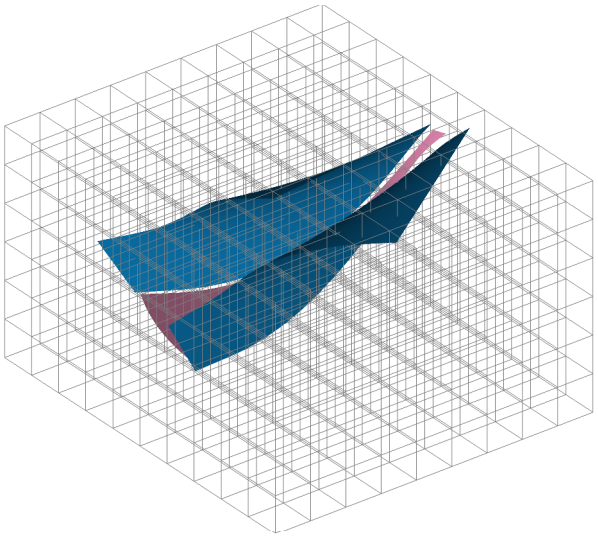


Figure 4.8 – Partitionnement de l'espace total

Pour chaque coin de voxel, le calcul de l'effort répulsif est effectué comme précédemment selon l'équation (4.2) avec n le nombre de points obstacles (nœuds du maillage) :

$$\boldsymbol{\mathcal{F}_{rep}} = \sum_{i=1}^{n} \boldsymbol{\mathcal{F}}_{repi} \quad \text{avec :} \quad \boldsymbol{\mathcal{F}}_{repi} = \begin{cases} (\eta.(\frac{1}{r_i} - \frac{1}{r_0}).\frac{1}{r_i^2}.\boldsymbol{u}_i) & \text{si} \quad r_i < r_0 \\ \mathbf{0} & \text{sinon} \end{cases} \tag{4.2}$$

Pour la programmation du calcul des efforts répulsifs, une situation est alors simple à traiter. En effet, si le voxel étudié comporte huit coins dont la distance minimum avec l'obstacle C_S est supérieure au voisinage

$$min(PTS_{C_S} - P_{coin}) > r_0 \tag{4.3}$$

alors, le voxel correspondant ne comporte que des coins avec des efforts répulsifs nuls. De ce fait, l'interpolation d'un point étudié dans ce cube est également soumise à un effort répulsif nul. L'application de cette théorie sur l'*impeller* présenté précédemment montre, en figure 4.9, la suppression des voxels non utiles à cette simulation.

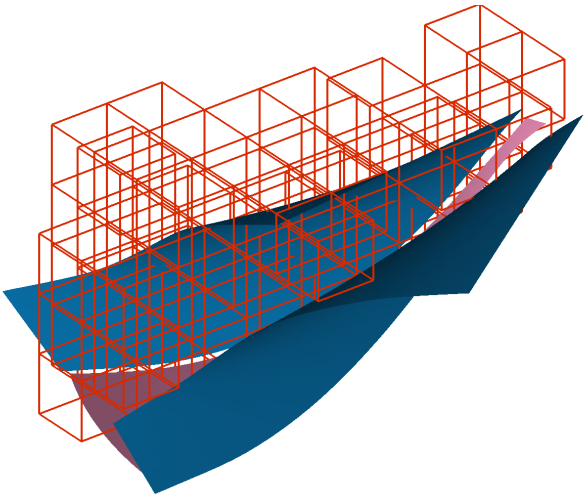
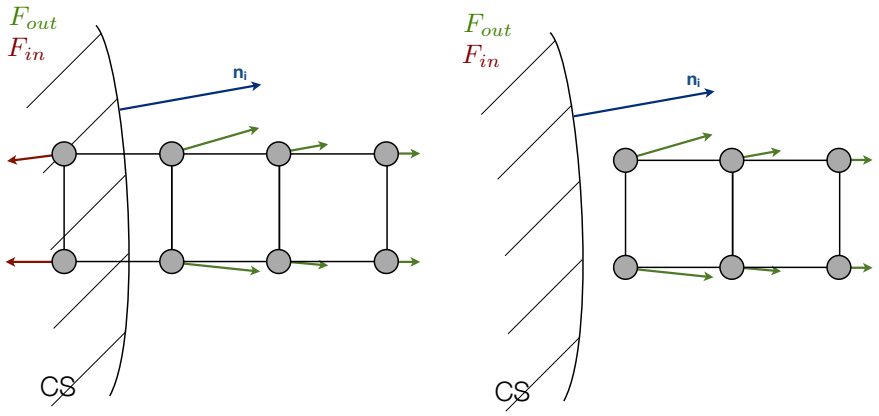


Figure 4.9 – Illustration d'une conservation de voxels utiles ($10mm$)

Pour réduire encore la quantité de calcul, il est possible d'évaluer la disposition du voxel vis-à-vis des surfaces obstacles. Un test sur le signe des produits scalaires, entre les normales extérieures matière des points sur les surfaces obstacles les plus proches et les coins des voxels, permet de détecter ceux qui sont uniquement dans la zone accessible de l'outil (hors collision) ou à cheval sur une partie d'espace en collision et l'autre non comme illustré en figure 4.10(a). L'effort tendant vers l'infini à l'approche de l'obstacle, il est alors possible de supprimer les voxels à cheval sur la *Check-Surface* comme représentés en figure 4.10(b) étant donné que la dimension des voxels utilisés lors de cette étude est suffisamment faible. Il est important de noter que l'utilisation d'octree permettrait d'atténuer l'impact de cette nouvelle problématique, car, en choisissant la dimension minimum de ces derniers, il serait possible de couvrir l'espace perdu lors de la suppression des points à l'intérieur de l'obstacle par de nouveaux voxels de dimension inférieur. L'ensemble des voxels obtenus englobe bien la pièce étudiée remarquant qu'il est possible de prendre en compte d'autres obstacles attachés à la machine si nécessaire. Cela ne ferait que rajouter un élément au calcul.



(a) Illustration du problème (b) Traitement du problème

Figure 4.10 – Voxel limite

La suite de cette partie porte justement sur l'influence du traitement de la voxelisation retenue avec, dans un premier temps, une présentation des différentes méthodes d'interpolation utilisables. En effet, les points étudiés au cours de la simulation ne tomberont jamais en concordance avec les points représentant les coins des voxels définis. Il est donc nécessaire d'insérer une phase d'interpolation et d'étudier l'erreur obtenue par rapport à la simulation continue.

2.2 Influence de la méthode d'interpolation retenue

Dans cette sous-partie, les deux principales méthodes d'interpolation utilisables sous Matlab® sont présentées afin de comprendre comment sont obtenues les valeurs des efforts répulsifs aux points étudiés du maillage de l'outil en fonction de ceux calculés au niveau des coins de chaque voxel concerné.

La méthode classiquement utilisée est la méthode d'interpolation trilinéaire. Le schéma donné en figure 4.11 illustre cette dernière [Tarbutton et al., 2013].

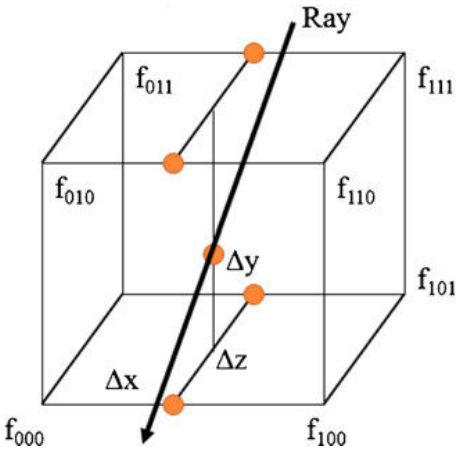


Figure 4.11 – Interpolation linéaire $3D$ [Tarbutton et al., 2013]

Avec :

$$\begin{aligned} val = {} & (1-\Delta x)(1-\Delta y)(1-\Delta z).f_{000} + (\Delta x)(1-\Delta y)(1-\Delta z).f_{100} \\ & + (1-\Delta x)(\Delta y)(1-\Delta z).f_{010} + (1-\Delta x)(1-\Delta y)(\Delta z).f_{001} \\ & + (\Delta x)(\Delta y)(1-\Delta z).f_{110} + (\Delta x)(1-\Delta y)(\Delta z).f_{101} \\ & + (1-\Delta x)(\Delta y)(\Delta z).f_{011} + (\Delta x)(\Delta y)(\Delta z).f_{111} \end{aligned} \quad (4.4)$$

Dans le but de ne pas complexifier la problématique, une interpolation linéaire, comme présentée précédemment a été utilisée au cours des différentes simulations. Cependant, il semble important de noter la différence obtenue avec un second type d'interpolation dit cubique proposé, lors de l'interpolation effectuée sous Matlab®. La figure 4.12 compare la différence entre les erreurs de l'effort répulsif obtenues par une interpolation linéaire (en bleu) et une interpolation cubique (en rouge) pour trois dimensions de voxels sur une trajectoire dite alignée dans la veine qui est décrite par la suite 4.14. Le point important à retenir au niveau de cette figure n'est donc pas l'évolution de l'erreur relative qui va être détaillée par la suite, mais l'ordre de grandeur des erreurs obtenues. En effet, cela permet de voir que le fait de travailler via une interpolation cubique plutôt que linéaire permet de se rapprocher beaucoup plus de la réalité en obtenant une erreur à peu près divisée par 10.

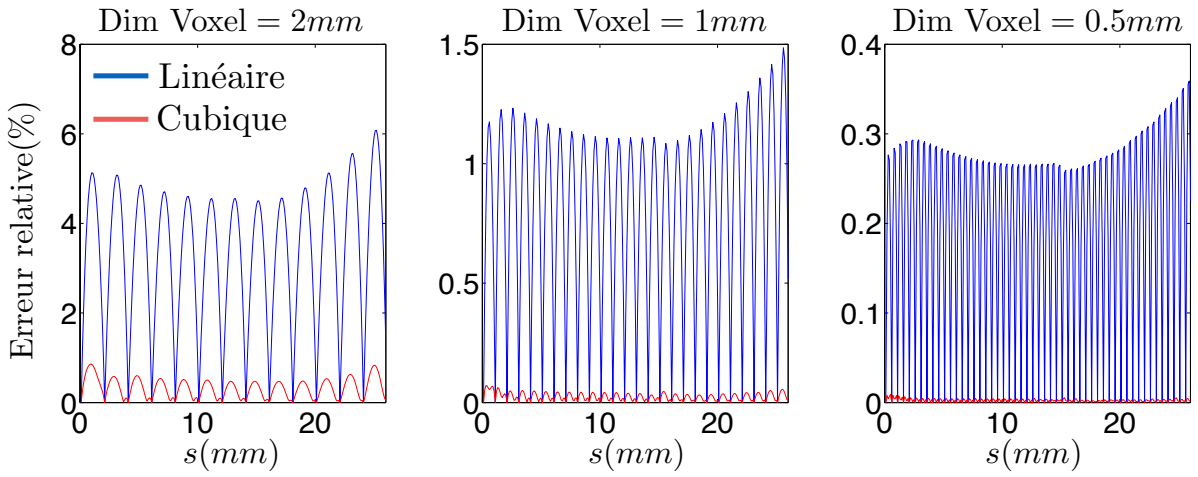


Figure 4.12 – Comparaison des méthodes d'interpolation

L'interpolation cubique s'appuie sur trois voxels consécutifs autour du point considéré pour effectuer le calcul. Il est donc important de soulever que l'utilisation d'octrees nécessite une interpolation linéaire, car le cube étudié n'a pas forcément de voisins de la même taille et donc l'interpolation cubique n'en devient que plus complexe. Par contre, il paraîtrait intéressant d'utiliser ces dits octrees avec une interpolation linéaire avec une modification, selon leur positionnement, de la taille des différents cubes afin d'optimiser le calcul aux domaines de l'espace les plus importants.

Lors des simulations à venir, l'étude est effectuée avec l'interpolation linéaire pour un souci de lisibilité et de comparaison des résultats obtenus. Bien évidemment, la programmation existe avec une interpolation cubique. Chacune des figures obtenues à l'aide de l'interpolation linéaire possède une figure équivalente avec une interpolation cubique fournie en annexe 4.

2.3 Influence de la taille de voxelisation sur la génération de trajectoires

Afin d'analyser l'influence de l'approximation induite par la modélisation du champ d'effort répulsif par des voxels, deux cas d'étude sont proposés. Étant donné la forme de l'effort en inverse de la distance aux surfaces de collision, les niveaux et gradients d'effort peuvent être très différents. À la limite du voisinage, le niveau est quasi constant et très faible alors que près des surfaces obstacles, les niveaux et variations sont très élevés. Il parait donc important de travailler sur deux cas distincts.

Tout d'abord, l'étude porte sur une droite pouvant être assimilée à une trajectoire d'usinage programmée, et dans un second temps, l'analyse est effectuée sur une droite permettant de voir l'impact du rapprochement du point d'étude vers l'obstacle.

Un point particulier des droites programmées présentées ci-dessus est que ces dernières passent par les coins des voxels. Ainsi, sur les sommets des voxels, les valeurs d'effort sont exactement

celles de la modélisation continue du champ d'effort (méthode directe), alors qu'à l'intérieur des voxels, les valeurs seront données par l'interpolation linéaire des valeurs sur les sommets. Cette vision peut être facilement illustrée en $2D$ comme donnée en figure 4.13. Ce graphique montre que par une interpolation linéaire les discontinuités d'ordre C_1 induisent des erreurs maxi par rapport à la modélisation réelle au milieu de chacun des segments représentés.

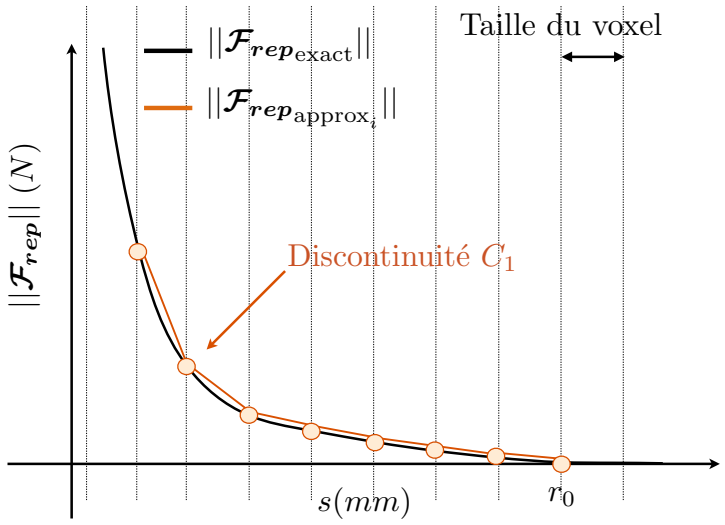


Figure 4.13 – Illustration 2D

2.3.1 Droite alignée dans la veine

Présentation du problème étudié

La figure 4.14 présente le cas d'étude d'une droite globalement alignée avec l'axe de la veine définie par les parois, donc avec une distance de chacun de ses points à peu près similaires avec l'obstacle (sans un fort rapprochement). Dans ce cas, les niveaux d'efforts sont à faible évolution et cela représente la plupart des cas d'étude.

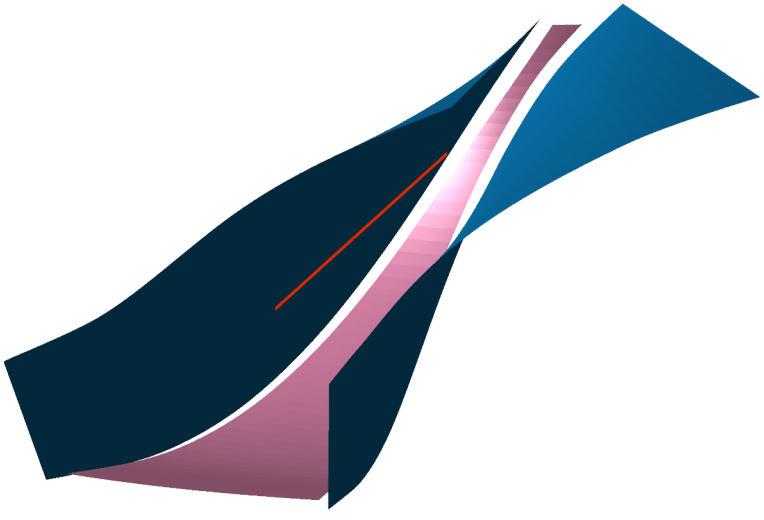


Figure 4.14 – Droite de l'espace représentant une trajectoire simulée

Analyse de l'erreur sur l'effort répulsif

En discrétisant cette droite sur la zone où évoluerait l'outil, il est possible de comparer les sollicitations d'effort répulsif auxquelles seraient potentiellement soumis les nœuds de son maillage.

La figure 4.15 permet d'analyser l'influence de la voxelisation sur l'effort répulsif pour différentes tailles de voxels. Les trois courbes bleues, vertes et noires représentent respectivement l'erreur relative obtenue par voxels de taille 2, 1 et 0.5mm calculée comme donnée en équation (4.5) (avec i la taille du voxel retenue).

$$\text{Erreur relative} = \frac{||\boldsymbol{\mathcal{F}_{rep}}||_{\text{approx}_i} - ||\boldsymbol{\mathcal{F}_{rep}}||_{\text{exact}}}{||\boldsymbol{\mathcal{F}_{rep}}||_{\text{exact}}} \tag{4.5}$$

Pour ce cas, l'erreur relative est approximativement constante par taille de voxels. Elle atteint 6% pour une taille de 2mm et est réduite à 0,5% pour un voxel de 0,5mm de côté. Bien évidemment, il est important de se rappeler que plus la voxelisation est fine, plus le temps de calcul correspondant au calcul de l'effort sur le partitionnement de l'espace est important. Cette méthode sera appliquée par la suite à la trajectoire complexe décrite précédemment dans le but de décrire l'erreur obtenue en fonction de la voxelisation choisie.

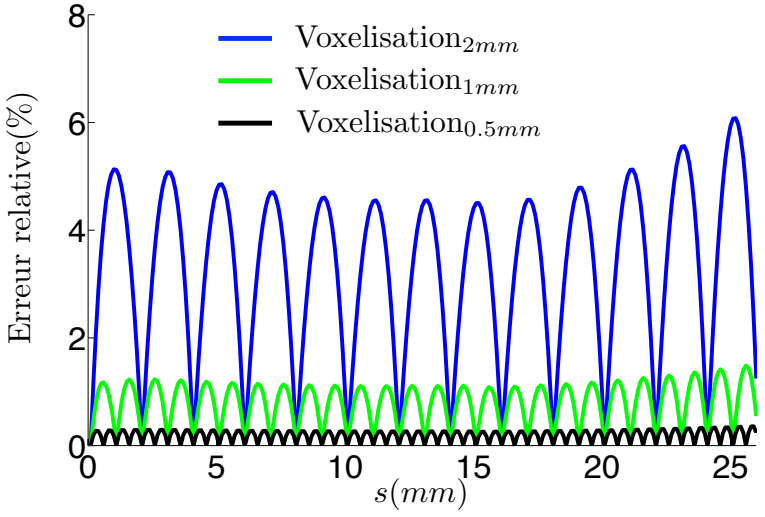


Figure 4.15 – Erreur relative de l'effort répulsif obtenu

2.3.2 Droite perpendiculaire aux surfaces obstacles

Présentation du problème étudié

La formulation de l'effort répulsif étant de type inverse, un rapprochement linéaire vers l'obstacle ne génère pas une modification linéaire de l'effort reçu. C'est pour cela qu'une seconde partie d'étude est effectuée. Cette simulation est illustrée en figure 4.16. Encore une fois cette droite passe par les coins des voxels afin de pouvoir comparer les différentes voxelisations programmées entre elles.

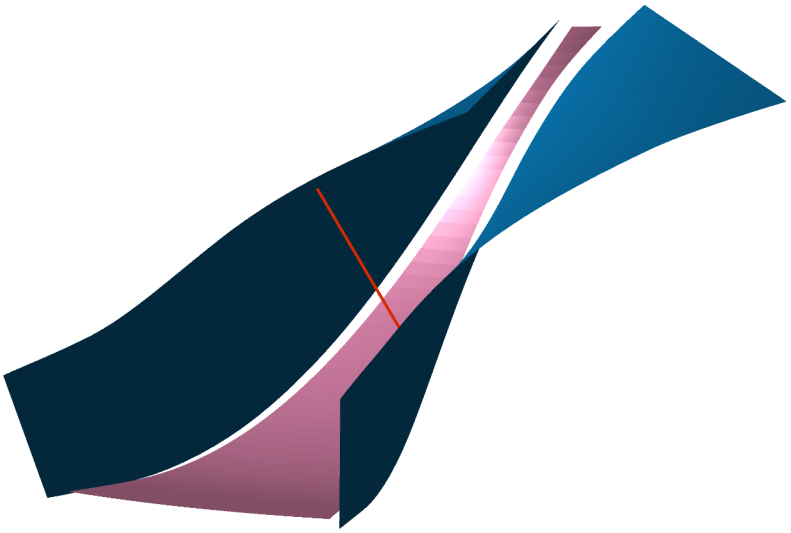


Figure 4.16 – Droite de l'espace représentant une trajectoire

Analyse de l'erreur sur l'effort répulsif

Cette sous-partie permet de comparer l'influence de la voxelisation sur l'effort répulsif pour différentes tailles de voxels dans un cadre de rapprochement du point d'étude vers l'obstacle défini comme tel. La figure 4.17 illustre cette étude.

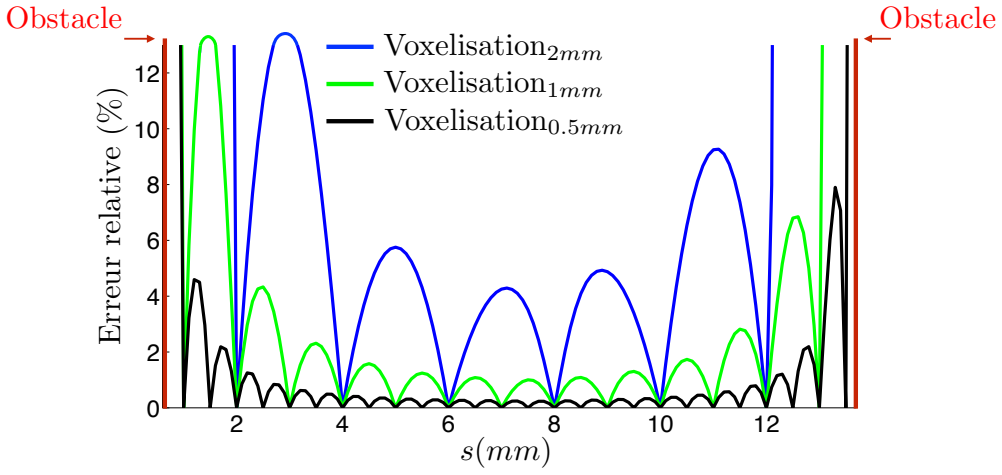


Figure 4.17 – Erreur relative de l'effort répulsif obtenu

Pour ce second cas d'étude, il est possible de constater que l'erreur relative augmente avec des niveaux plus importants d'efforts et tend vers l'infini près des surfaces obstacles. Cela permet alors de montrer qu'il semble essentiel d'utiliser un effort répulsif important permettant de tenir l'outil étudié suffisamment loin de l'obstacle, afin de ne pas avoir d'erreurs trop grandes par rapport à la simulation continue.

Au cours des travaux présentés précédemment les calculs du champ d'effort par voxels sont effectués par interpolation linéaire sur les sommets. Il est possible d'accroître la qualité de ces

résultats en utilisant une interpolation cubique, sans augmenter de manière significative les temps de calcul comme déjà expliqué. Il serait même possible d'augmenter la taille des voxels admissibles en jouant sur l'orientation de ces derniers afin d'éviter des erreurs dues, notamment, au rapprochement de deux obstacles.

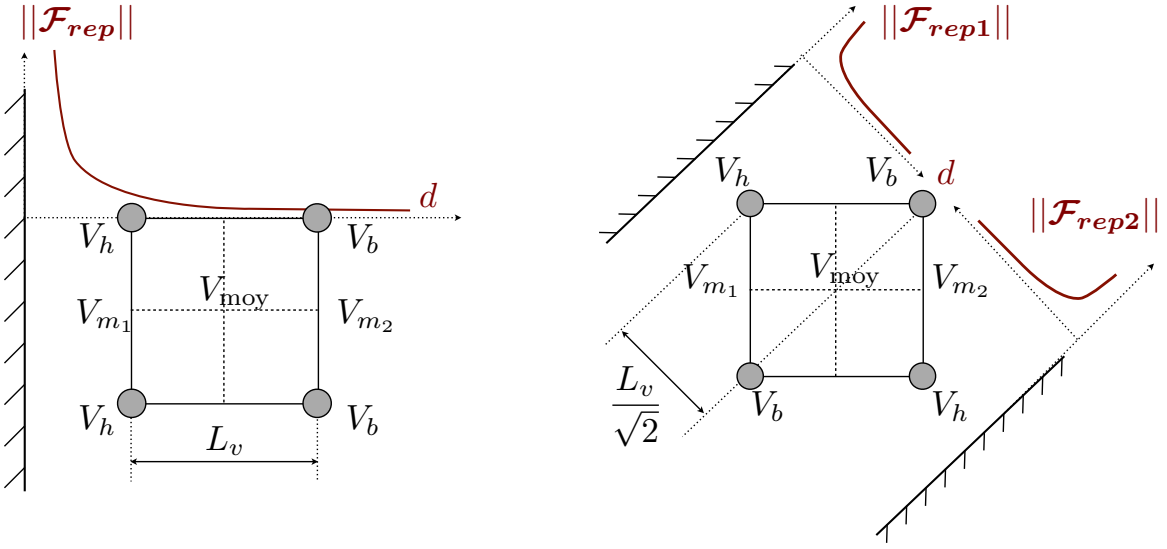
2.4 Augmentation de l'admissibilité de la taille des voxels

L'erreur obtenue à ce niveau est due au rapprochement des deux surfaces obstacles ne permettant pas d'obtenir une solution comparable au réel avec les interpolations proposées. La problématique soulevée est illustrée en figure 4.18 où un voxel élémentaire est représenté avec comme paramètres :

— V_h : valeur haute de l'effort répulsif parmi les coins étudiés,

— V_b : valeur basse de l'effort répulsif parmi les coins étudiés.

La courbe visible au-dessus du voxel est une représentation de l'effort répulsif variant selon le rapprochement du point étudié vers l'obstacle.



(a) Cas idéal (b) Cas critique

Figure 4.18 – Positionnement de la voxelisation

La figure 4.18(a) apporte une interpolation satisfaisante, car la direction d'arête du voxel est colinéaire au gradient. Dans ce cas, la valeur moyenne obtenue après interpolation est assez cohérente avec le positionnement du voxel étudié donné en équation (4.6).

$$V_{m1} = V_h, \quad V_{m2} = V_b, \quad V_{moy} = \frac{V_{m1} + V_{m2}}{2} \tag{4.6}$$

La figure 4.18(b) illustre le cas où l'interpolation linéaire fournie par Matlab® n'est pas cohérente avec la réalité. En effet, si l'orientation du cube étudié est de 45° par rapport au gradient de la surface (à la normale à la surface), cela implique une erreur (l'erreur maximale avec

cette orientation). Il est alors utile de souligner que dans ce cas formulé en équation (4.7) :

$$V_m = \frac{V_{m1} + V_{m2}}{2} = \frac{V_h + V_b}{2} \neq V_b \tag{4.7}$$

alors que logiquement le résultat devrait être $V_m = V_b$. L'erreur relative obtenue est alors explicitée en équation (4.8) :

$$\text{erreur relative}_{max} = |\frac{V_m - V_b}{V_b}| = |\frac{\frac{V_h+V_b}{2}-V_b}{V_b}| = |\frac{V_h - V_b}{2.V_b}| = \frac{L_v}{2.\sqrt{2}.V_b} \tag{4.8}$$

avec L_v la dimension du voxel étudié. Il est important de relever que, pour résoudre cette problématique, l'utilisation de voxel suffisamment petit permet de diminuer l'erreur obtenue. De plus, il serait possible d'orienter les voxels perpendiculairement à la moyenne des normales aux différentes surfaces obstacles. Cela permettrait de se rapprocher du cas idéal présenté précédemment.

Cela est un point important à relever, bien que, lors d'une utilisation actuelle de la simulation, la gestion d'une voxelisation suffisamment petite s'avère assez facile à programmer. De plus, pour des surfaces complexes l'orientation des voxels n'a plus réellement de sens.

La suite de cette partie concerne alors l'utilisation du programme utilisant la voxelisation de l'espace sur une trajectoire réelle d'usinage définie précédemment en $v = 0{,}5$ afin de relever les différences de position entre une résolution exacte et une résolution par voxelisation. De plus une étude temporelle est à prévoir afin d'estimer le gain apporté par cette méthode.

3 Influence de la voxelisation la simulation d'usinage de l'*impeller*

3.1 Analyse de l'écart géométrique obtenue avec la voxelisation

Après avoir analysé l'effet de la voxelisation sur les efforts s'exerçant sur l'outil, il est intéressant de regarder l'influence sur la correction obtenue pour une trajectoire réelle. Comme le point C_L est guidé par une courbe isoparamétrique posée sur la surface *Offset* de la surface guide de la veine, il est uniquement nécessaire de comparer les différences d'orientation de l'axe de l'outil.

La figure 4.19 représente la simulation exacte développée durant la chapitre 3 et la simulation avec des voxels de $2mm$ sur la même trajectoire pour le point piloté. Il est possible d'observer qu'il est difficile de visionner l'écart induit par la voxelisation tellement la trajectoire obtenue semble proche de la trajectoire obtenue par simulation exacte.

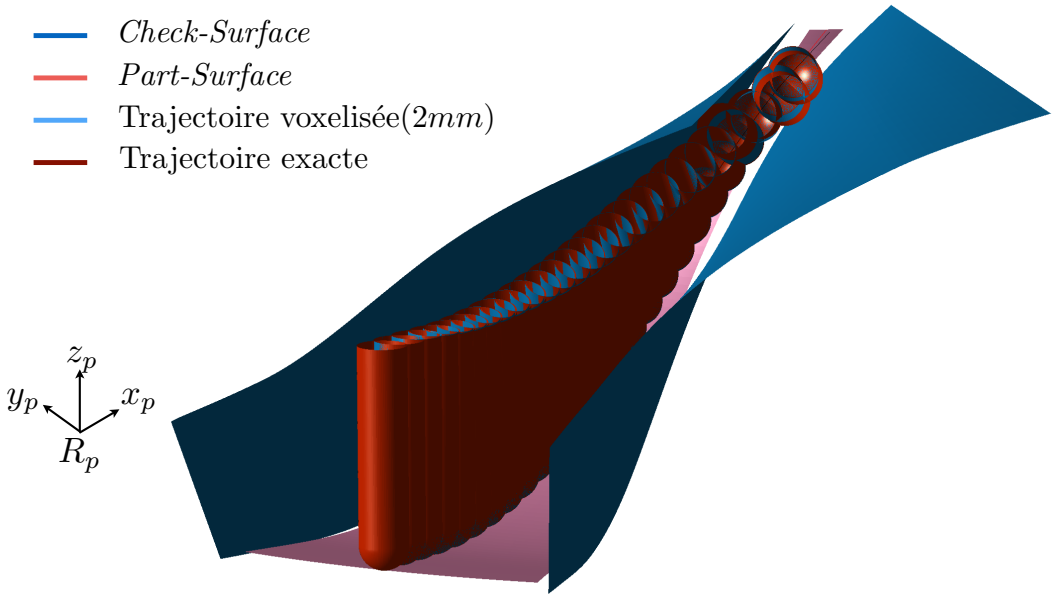


Figure 4.19 – Comparaison méthode continue et voxelisation

Pour ce faire, les courbes de la figure 4.20 comparent les évolutions de l'angle d'inclinaison latérale θ_f au niveau du point G obtenu par la résolution du PFD en utilisant l'évaluation des efforts répulsifs exacte ou par voxels.

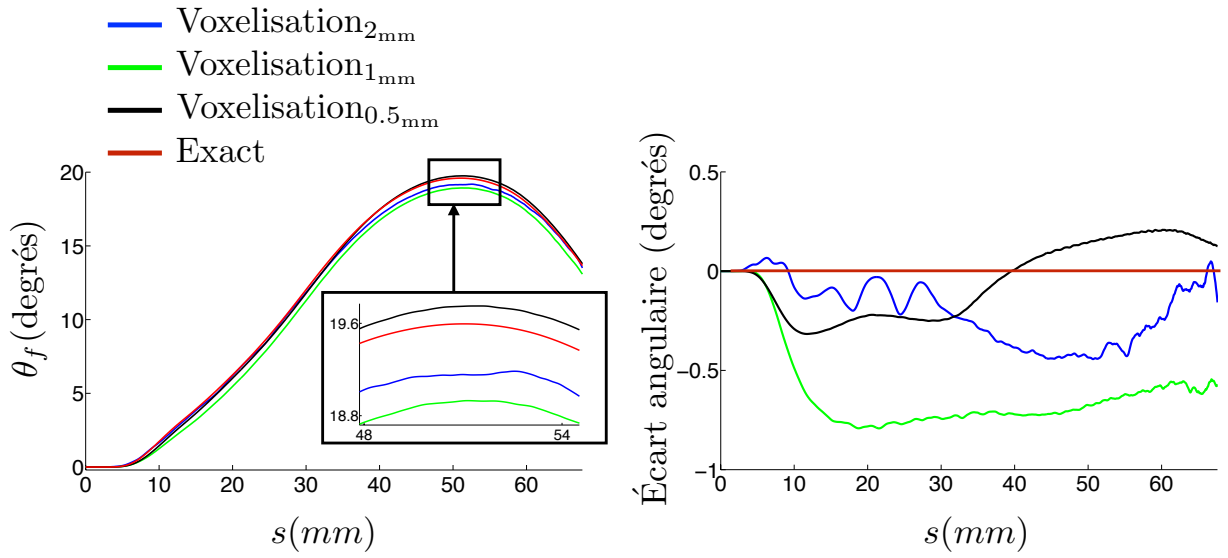


Figure 4.20 – Influence de la voxelisation sur l'orientation outil

Avec l'utilisation des voxels, les sollicitations en effort perçues par l'outil le long de la trajectoire sont différentes du fait de la causalité liée au PFD. Ainsi, les trajectoires outils sont automatiquement différentes. La figure 4.20 à gauche permet de montrer la faible sensibilité de l'orientation outil par rapport à la modélisation du champ d'effort. En effet, les évolutions de l'orientation de l'axe de l'outil sont qualitativement très proches.

Il faut noter que les voxels, s'ils sont programmés avec une taille trop importante, introduisent une oscillation de l'orientation de l'axe outil (courbe bleue sur l'écart d'angle en figure 4.20 à droite) qui pourrait s'avérer néfaste pour le suivi de la trajectoire effective sur machine. En revanche, pour des voxels de taille faible (dans ce cas 0,5mm) l'écart existe mais la trajectoire obtenue est quasiment sans oscillation. Il faut souligner que l'écart tracé en figure 4.20 est une illustration de la différence entre la trajectoire obtenue par voxelisation et la trajectoire dite exacte, cette dernière peut elle-même contenir des oscillations. De ce fait, il serait important de programmer un choix de taille de voxelisation acceptable selon la régularité d'évolution attendue.

Après avoir décrit l'influence de la voxelisation sur la trajectoire obtenue, il a été montré que l'écart avec la simulation exacte est très faible et qu'elle reste tout à fait cohérente avec l'évitement de l'ensemble obstacle concerné. Cependant, un point important reste à étudier. En effet, l'intérêt initial de la voxelisation est le gain en terme de temps de calcul via l'application de cette méthode. Pour ce faire, la formalisation de la complexité réduite ainsi que l'étude en terme de gain de temps de simulation sont réalisées au cours de cette partie.

3.2 Analyse de la complexité temporelle obtenue avec la voxelisation

3.2.1 Complexité réduite du champ d'effort voxelisé

Comme défini précédemment, pour une trajectoire possédant P positions outils, il est nécessaire pour chacune d'entre elles d'évaluer l'effort répulsif à chaque nœud du maillage outil N, en se basant sur le voxel englobant. Le nombre d'itérations est alors d'ordre de grandeur $(P.N)$. À ce calcul il faut ajouter l'évaluation de l'effort répulsif émanant des points de la surface obstacle (dont le nombre est noté C_S) en chaque coin de voxel qui dépend alors de la taille de l'espace et de la taille des voxels. Étant donnée une voxelisation, avec (L_1,L_2,L_3) les dimensions du parallélépipède englobant la zone d'évolution de l'outil et L_v la dimension caractéristique des voxels, il est alors possible de formaliser un majorant du nombre de voxels à considérer $(L_1.L_2.L_3)/L_v^3$ selon la simulation réalisée.

La complexité du calcul de la trajectoire vis-à-vis de l'effort répulsif s'écrit alors sous la forme donnée en équation (4.9).

$$\text{Complexité}_{\text{Calcul voxels}} = P.N + C_S.(\frac{L_1.L_2.L_3}{L_v^3}) \tag{4.9}$$

Si le problème est de taille suffisante il est alors intéressant d'un point de vue quantité de calculs à effectuer, de se baser sur une modélisation du champ d'effort répulsif par voxels plutôt que de l'évaluer de manière systématique par un calcul de distances vis-à-vis des nœuds des surfaces obstacles. L'équation (4.10) rappelle l'expression de la complexité par calcul direct.

$$\text{Complexité}_{\text{Calcul direct}} = P.N.C_S \tag{4.10}$$

Avec la formulation proposée, le gain en calcul est effectif dès que la taille de la trajectoire (c.-à-d. le nombre de positionnements outil P) vérifie le critère donné en équation (4.11) traduisant la demande $\text{Complexité}_{\text{Calcul voxels}} < \text{Complexité}_{\text{Calcul direct}}$.

$$P > (\frac{C_S}{C_S - 1}).(\frac{1}{N}).(\frac{L_1.L_2.L_3}{L_v^3}) \tag{4.11}$$

À l'aide de cette inéquation, il est possible de faire un bilan du gain apporté par la voxelisation en terme de temps de calcul. La taille de la surface obstacle n'a quasiment pas d'influence sur cette limite, car $\frac{C_S}{C_S-1} \approx 1$. Il est donc possible de reformuler l'inéquation précédente de manière littéraire (équation (4.12)) :

$$\text{Nombre de positionnements outil} > \frac{\text{Nombre de voxels}}{\text{Nombre de nœuds du maillage outil}} \tag{4.12}$$

Cette inéquation est cohérente, car effectivement, plus la dimension des voxels est faible, alors plus le nombre de nœuds de la voxelisation est grande, ainsi, pour un maillage outil fixé, plus le nombre de positionnements outil utile pour valider l'intérêt de la voxelisation est grand. En effet, le temps de pré simulation serait alors augmenté.

Une fois le critère permettant d'évaluer le gain en terme de nombre de calculs présenté, il semble alors essentiel d'étudier finalement le gain de performance apportée par la méthode de voxelisation, présenté au cours de ce chapitre sur l'*impeller*.

3.2.2 Gain de performance

Pour illustrer le gain de temps de calcul, il est possible de comparer le temps nécessaire à la génération de trajectoires pour une même trajectoire, en passant soit par la méthode de calcul exact de l'effort répulsif par les distances, soit par la voxelisation de l'espace avec utilisation de l'interpolation linéaire. Comme rappelé en fin du chapitre 3, la trajectoire d'usinage appliquée sur une *Part-Surface* de l'*impeller* est constituée de 60 passes successives, pour une longueur moyenne de passe de 80mm. De plus, l'*impeller* est composé de 7 *Part-Surfaces*, cela donne alors une longueur approximative d'usinage totale en terme d'abscisse curviligne fournie en équation (4.13).

$$s_{total} = 33600mm \tag{4.13}$$

Les calculs sont effectués sous Matlab® avec un ordinateur équipé d'un processeur INTEL Xeon® $CPU - ES\,1620@3.6GHz$. Le volume voxelisé est environ de taille $660cm^3$ par des voxels de dimension $2mm$ (121000 coins de voxels), $1mm$ (783432 coins de voxels) et enfin $0{,}5mm$ (5389119 coins de voxels). Le maillage outil comporte 500 nœuds, celui des surfaces obstacle en totalise 5100 environ. Le tableau 4.1 met en évidence un gain de temps important pour une trajectoire d'usinage complète.

	Méthode de calcul direct	Méthode de calcul par les voxels		
Calcul de l'effort sur le partitionnement	néant	Voxels 2*mm* 32*s*	Voxels 1*mm* 138*s*	Voxels 0,5*mm* 971*s*
Calcul des positionnements outils	54*h*48*min*	59*min*		
Temps total	54*h*48*min*	59*min*	61*min*	75*min*

Tableau 4.1 – Comparaison des temps de génération des trajectoires entre calcul exact et calcul par voxels de l'effort répulsif

Cette méthode permet alors une génération plus rapide des trajectoires. Cependant cette simulation reste longue en temps de calcul, car elle est programmée avec Matlab® sous la forme d'un langage interprété et donc un code non optimisé. Il est important de souligner que lors d'une programmation de **Geo5XField** en C^{++} il est raisonnable d'envisager un gain sur le temps de calcul de 100 à 500.

Ce gain est d'autant plus intéressant que le trajet est long, ce qui est le cas pour la pièce test, pour les pièces usinées en bout d'outil sur des formes complexes : le balayage de la surface requiert un nombre de passes élevées, qui sont d'autant plus serrées que le critère de hauteur de crête transversale est faible.

4 Conclusions et perspectives

L'intérêt des travaux développés dans cette partie est apparu lors de l'analyse des temps de calcul effectuée en fin de chapitre 3 où il était aisé de repérer une explosion combinatoire en terme de calcul de distance aux obstacles pour une prise en compte d'une géométrie réaliste pour l'ensemble porte-outil, représenté par un maillage.

L'idée principale de cette partie est donc de tenter de diminuer le temps de calcul considéré à l'aide d'une méthode de partitionnement de l'espace en étudiant l'écart réalisé avec une simulation obtenue via la formulation décrite dans le chapitre 2 et 3.

Par la suite, une analyse de l'influence de la voxelisation sur la génération de la trajectoire est réalisée afin d'évaluer l'erreur de calcul pour un effort répulsif interpolé. Cette étude est séparée en deux points principaux. Dans un premier temps, des trajectoires sont modélisées, afin

de représenter des droites passant par les coins des voxels. Cela permet d'obtenir une comparaison possible. Par la suite, la voxelisation est appliquée à l'*impeller* utilisé pour la simulation dite exacte. D'un point de vue géométrique, les orientations axe outil des trajectoires générées sont proches des orientations initialement obtenues ; la voxelisation a donc peu d'effet sur les trajets calculés par l'équilibre dynamique pour des tailles de voxels suffisamment faibles.

L'analyse de la réduction de complexité est alors évoquée, cela permet de situer le niveau de complexité associé au maillage des surfaces et de l'outil à partir duquel la voxelisation devient avantageuse en terme de temps de calcul. En effet, pour ce qui est de l'attribution des efforts répulsifs durant la simulation, la voxelisation est toujours plus rapide, mais cette dernière impose un pré calcul d'attribution d'efforts répulsifs aux coins des voxels concernés. Une analyse des gains de performance est alors réalisée et les essais montrent que d'un point de vue temps de calcul, une telle approche est très intéressante : les configurations testées réduisent fortement le temps de calcul. Ceci est d'autant plus intéressant que les nœuds de discrétisation des surfaces obstacles et outil/porte-outil sont élevés.

Afin de pouvoir comparer par rapport à l'existant, il serait intéressant de programmer ce principe de voxelisation en C^{++} afin d'obtenir une vision plus réaliste de l'avantage fourni par cette méthode.

Conclusions générales et perspectives

Conclusions

Dans ce manuscrit les travaux ont porté sur la présentation d'une nouvelle méthode d'évitement de collisions d'un point de vue dynamique en génération de trajectoires 5 axes par utilisation de champs et de potentiels répulsifs.

Afin de comprendre l'objet d'étude de cette thèse, le chapitre 1 débute par une analyse de l'existant concernant la gestion des collisions en usinage 5 axes. La première hypothèse de ces travaux est de ne traiter le problème de collision qu'au niveau des interférences globales. Actuellement, l'évitement de collision dans le domaine de l'usinage 5 axes est principalement réalisé à l'aide d'une approche géométrique et itérative. L'étude des méthodes publiées dans la littérature est réalisée en partitionnant la gestion des collisions en trois phases : la modélisation de l'environnement, l'évaluation des collisions et la correction de l'orientation de l'axe de l'outil. Cela a permis d'aborder de manière systématique un nombre important de méthodes parmi lesquelles trois méthodes principales : la méthode des enveloppes convexes, l'analyse de la visibilité, et la génération de surface *Offset*. L'étude de la phase de correction montre une volonté croissante d'optimiser l'enchainement des orientations de l'axe de l'outil pour garantir la fluidité de la trajectoire dans un contexte d'usinage 5 axes.

Un autre type de méthodes, basé sur des champs et des potentiels répulsifs et couramment utilisé dans le domaine de la robotique mobile, est ensuite exploré avec les différentes formulations existantes. L'expression de l'effort répulsif, qui tend vers l'infini quand le robot s'approche d'un obstacle, permet la modification de la trajectoire du robot mobile et garantit l'assurance de non-collision. Cependant, dans la plupart des travaux issus du domaine de la robotique mobile, les travaux ne font pas effet de l'aspect dynamique de la modification d'orientation effectuée. En effet, l'impact de l'écart obtenu entre la trajectoire réelle et la trajectoire programmée en robotique mobile est moins crucial qu'en usinage où les tolérances sont plus restrictives.

Cette étude bibliographique met trois besoins principaux en avant : l'assurance d'une continuité en UGV, le passage du 3 au 5 axes avec la prise en compte du positionnement de nouveaux obstacles et la simplicité ainsi que l'efficacité de la programmation. Afin de répondre à ces différents besoins, une vision dynamique du problème pourrait alors être abordée afin de répondre à la problématique principale de cette thèse concernant l'évitement de collisions globales avec une

modification d'orientation suffisamment lisse et un retour en orientation programmée.

Le second chapitre a pour objectif d'appliquer l'utilisation de champs répulsifs au cadre de la génération de trajectoire 5 axes. La modification du diagramme d'activité de la simulation de fabrication par rapport à celui donné au niveau du chapitre 1 illustre bien un des avantages de cette méthodologie. En effet, le système n'est plus itératif, car avec une telle modélisation, le repositionnement est assuré hors collision. Afin de tester ceci, une première étape d'étude théorique est alors réalisée avec le développement du modèle dynamique basé sur l'analogie mécanique. Afin d'exprimer les hypothèses de travail retenues au cours de cette thèse, la composition de mouvement entre l'outil et la pièce est alors décrite et rappelée en équation (4.14).

$$\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/R_p) = \mathbf{Mvt}(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}}) + \mathbf{Mvt}(\text{outil}_{\text{strat}}/R_{fnt}) + \mathbf{Mvt}(R_{fnt}/R_p) \quad (4.14)$$

Nous avons alors choisi de négliger les aspects dynamiques apportés par les mouvements relatifs $\mathbf{Mvt}(\text{outil}_{\text{strat}}/R_{fnt})$ et $\mathbf{Mvt}(R_{fnt}/R_p)$. Afin de traiter l'aspect dynamique s'appliquant au mouvement $\mathbf{Mvt}(\text{outil}_{\text{corrigé}}/\text{outil}_{\text{strat}})$ deux manières d'aborder le problème sont proposées. La première modélisation 3D s'applique à un solide quelconque et la seconde modélisation, de type pendule inversé, permet une modélisation scalaire du mouvement. Ces deux modèles mènent à la résolution d'un système d'équations différentielles ordinaires. La méthode de Runge-Kutta permet la résolution numérique de l'équation utilisée issue du Principe Fondamental de la Dynamique.

Après avoir décrit l'équation générale du mouvement retenue au cours de cette étude, la partie suivante décrit, dans un premier temps, l'utilisation de l'action répulsive utilisée issue des travaux d'Oussama Khatib [Khatib, 1986]. Cette modélisation s'appuie sur un paramètre de voisinage et un paramètre d'effort répulsif. Lors de cette formalisation, la position du point piloté C_L ne doit pas être modifiée ainsi que sa vitesse d'avance curviligne. De ce fait, lors de l'entrée du point étudié de l'outil G dans le voisinage d'un point dit obstacle O_i, un effort repousse le point G. Étant donné que le point piloté suit sa trajectoire et que la zone entre le point C_L et le point G est considérée comme indéformable, une rotation est alors enclenchée.

La deuxième étape est de rappeler l'axe outil vers son orientation programmée à l'aide d'un système ressort amortisseur en appliquant la notion d'équilibre attachée au PFD. Plusieurs réglages sont alors indispensables afin d'obtenir un retour en orientation programmée lisse, dit apériodique d'un point de vue physique pour la phase hors voisinage et donc l'absence d'un second membre. L'influence de ces différents paramètres réglables est alors étudiée afin d'en comprendre l'effet dans le cas du régime transitoire du système.

Le chapitre suivant traite de la modélisation de configurations d'usinage s'appuyant sur des surfaces et des outils réalistes. Pour cela, les surfaces doivent être maillées afin d'être prises en compte. Les points issus de ce maillage jouent alors le rôle d'obstacles. La simulation commence alors par un cas d'étude simplifié ou plusieurs points obstacles se succèdent. Le régime permanent montre bien que l'axe outil a tendance à suivre la position des différents points et donc à osciller.

Cela permet de fixer de nouveaux paramètres afin que le maillage de l'obstacle ait le moins d'influence possible sur la correction de l'orientation de l'axe de l'outil. Dans le cas de cette étude, le maillage dit de référence pour les surfaces étudiées est fixé. De même la raideur du ressort est donnée. À la suite des différentes études, cela permet alors de calculer la pulsation propre du système modélisé, l'inertie amenant une atténuation des oscillations dans la zone de voisinage et une valeur d'amortissement afin de se trouver dans le cadre d'une réponse apériodique.

L'utilisation d'une surface complexe est alors traitée et montre une nouvelle problématique. En effet, jusque là l'effort répulsif appliqué sur le point de l'outil étudié G est la somme de tous les efforts répulsifs des points du maillage de l'obstacle dont le voisinage inclue le point G. Bien évidemment, lorsque la distance entre les points du maillage diminue, l'effort répulsif total diminue. Deux nouvelles formulations sont alors présentées en expliquant les avantages et les inconvénients de chacune d'elle. Enfin, cette partie se conclut par un exemple industriel sur une pièce de type *impeller*. Cette simulation permet de valider la méthode proposée avec un évitement de la collision et une trajectoire finale lisse.

La partie suivante de ce chapitre concerne alors la modélisation de l'outil. Deux nouvelles modélisations sont alors présentées avec pour chacune d'elle une simulation particulière. Tout d'abord l'outil est maillé de manière désordonnée. Tous les points de ce dernier sont pris en compte en faisant la somme des moments répulsifs obtenus ce qui pose un problème à cause des bras de levier. Pour régler ce problème, une nouvelle formulation est alors mise en place permettant d'atténuer cette désorientation. Le dernier cas d'étude est le maillage complet d'un outil associé à son porte-outil. La nouvelle simulation permet notamment de mettre en évidence qu'il faut encore trouver un compromis entre longueurs outil utilisable et correction d'orientation maximale. De plus le passage d'un coin représentant une poche ouverte permet de valider la méthode proposée. Finalement le tableau 4.2 permet de lister les paramètres fixés, les paramètres calculés et les paramètres modifiables par l'utilisateur prenant en compte l'ensemble des problématiques mises en évidence au cours de cette partie.

Paramètres	Fixé	Calculé	Réglable	Symboles	Valeur	Unité	Équation
Valeur du voisinage	X			r_0	10.10^{-3}	m	
Vitesse d'avance programmée	X			V_f	1	$m.min^{-1}$	
Coefficient de $\mathcal{F}_{rep}$	X			η	10^{-7}	$N.m^{-4}$	
Distance entre points			X	d_0	2.10^{-3}	m	
Pulsation propre		X		ω_0	52,35	$rad.s^{-1}$	(3.3)
Coefficient de raideur			X	k	1	$N.m.rad^{-1}$	
Inertie de l'outil		X		J	$3{,}65.10^{-4}$	$kg.m^2$	(3.14)
Facteur d'amortissement	X			ξ	1	$S.U.$	
Coefficient d'amortissement		X		c	$38{,}20.10^{-3}$	$kg.m^2.s^{-1}$	(3.9)

Tableau 4.2 – Paramètres

Afin de résoudre la problématique du temps de calcul très important dans le cas d'une simulation réaliste avec de nombreuses trajectoires, la notion de partitionnement de l'espace par voxelisation est alors envisagée dans le dernier chapitre. L'idée principale est de réaliser un pré traitement du calcul de l'action répulsive et d'étudier les gains obtenus. Le pré calcul de l'effort répulsif est réalisé sur chacun des coins des voxels. L'effort en tout point est alors calculé par une interpolation des valeurs obtenues sur chacun des coins du voxel auquel appartient le point. Deux méthodes principales sont présentées avec leurs avantages et leurs inconvénients, l'interpolation linéaire et l'interpolation cubique.

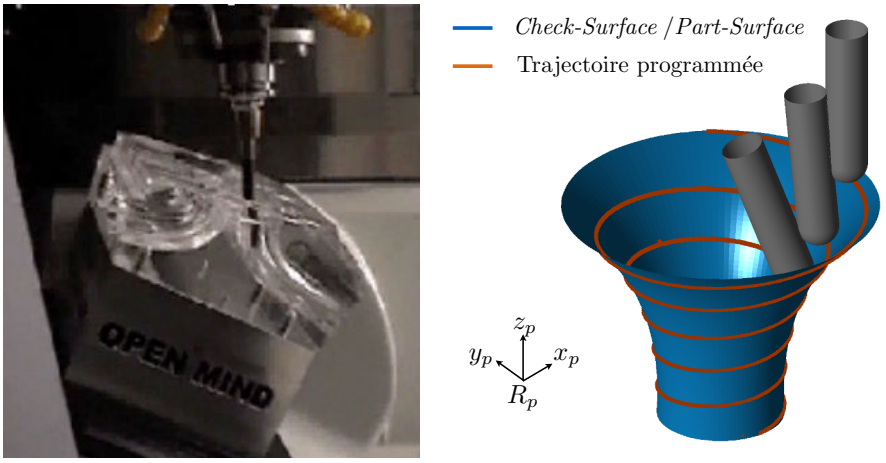
Lors de cette étude, les deux types d'interpolations ont été utilisés afin d'avoir une réelle comparaison. Il s'avère que l'erreur obtenue par rapport à la simulation directe issue du chapitre 3 est beaucoup moins importante via l'interpolation cubique. Par contre cette dernière engendre d'autres problématiques telles que le passage dans une zone étroite. L'analyse de l'écart est alors réalisée. Il est possible de voir que le résultat n'est pas très éloigné de celui obtenu par simulation directe (dite exacte). Par contre, l'utilisation de voxels plus petits, engendrant un temps de pré calcul plus important, apporte une modification d'orientation plus continue. Bien évidemment, le point fondamental de ce chapitre est de comparer les temps de calcul pour une simulation exacte et les simulations voxelisées. La première approche est l'étude de la complexité. Cela permet d'avoir une idée du paramétrage de la simulation à partir duquel la voxelisation apporte un gain de temps en fonction de la longueur de la trajectoire, le nombre de points obstacles (maillage des *Check-Surfaces*) et le nombre de nœuds représentants l'outil modélisé. Par la suite le gain de performance est mis en évidence par rapport à la simulation concluant le chapitre 3 avec, bien évidemment, un matériel informatique identique. Il semble alors clair que le gain en temps de calcul devient alors très important et pourrait convenir à un utilisateur une fois codé en C^{++}.

Perspectives

La première perspective qu'il est possible de souligner est la volonté d'augmenter la robustesse de la programmation afin d'envisager une industrialisation du logiciel. En effet, l'étude de diverses configurations reste à effectuer afin de garantir un fonctionnement sûr. De plus, afin de faciliter l'utilisation du programme, il semble indispensable de ne pas permettre à l'utilisateur une affectation complètement libre aux paramètres restant à fixer. L'idée principale serait de ne laisser, dans un cas initial, que le choix de la constante de raideur. Cela permettrait déjà de choisir un évitement plus ou moins violent avec, de même, un impact sur le retour en orientation programmée. Par la suite, il serait raisonnable d'entrevoir une possibilité de personnalisation de la simulation pour l'utilisateur averti. Dans ce cas, il serait possible de jouer sur le maillage apparenté à ω_0 représentant l'ordre de grandeur dans laquelle la simulation est réalisée.

Certain cas, non présentés au cours de cette thèse ont été étudiés tels que l'usinage d'une veine

où le cas particulier est que la *Part-Surface* est également la *Check-Surface*. Cela ne pose pas de problème pour la programmation, car avec la paramétrisation effectuée, la surface guide génère des trajectoires *Offsets*. La *Part-Surface* peut alors être considérée comme une *Check-Surface*. L'exemple fourni en figure 4.21 illustre ces propos. Par contre, les points obstacles se retrouvent très proche tout au long de la trajectoire. De ce fait l'outil suivant la courbe guide, ne sortirait pas du voisinage de la surface considérée comme obstacle.



(a) Usinage réel (b) Simulation

Figure 4.21 – Programmation d'usinage 5 axes d'une pièce dite de type veine

Toujours dans le but de compléter l'étude présentée au cours de cette thèse, il serait utile de proposer l'utilisation d'autres types d'outils. En effet, jusque là, seule l'utilisation d'outil hémisphérique est étudiée. Cela permet de ne pas s'inquiéter de l'ordre de réorientation entre θ_f et θ_t. Pour éviter un talonnage, il serait donc important de prendre cette nouvelle problématique de programmation. En effet, dans ce cas la hauteur de crête est impactée, il faudrait alors recalculer les trajectoires d'usinage du point C_L afin de ne pas générer un défaut de fabrication important dû à l'évitement de collision réalisé. Avec ce type d'outil, il serait alors utile de prendre en compte les collisions locales afin de permettre une assurance complète de non-collision.

Enfin, il faudra vérifier que les trajectoires générées sont compatibles avec la cinématique de la machine utilisée. En effet, l'étude présentée au cours de cette thèse possède l'avantage d'être le plus généraliste possible en n'imposant pas l'utilisation d'un type de machine en particulier. Cependant, il serait intéressant d'inclure dans la programmation ce type d'information permettant, grâce à la Transformation Géométrique Inverse correspondante de paramétrer la simulation de manière à respecter les limites machines autant d'un point de vue vitesse, accélération et jerk. Cela permettra de garantir une vitesse d'avance maximale et constante en fonction de la machine utilisée. Ce point est important à plusieurs niveaux, autant du point de vue de la qualité de la surface obtenue qu'en terme de productivité. De plus, l'affectation d'une machine permettrait d'obtenir les oscillations admissibles pour le moteur utilisé. En effet, l'analyse fréquentielle serait donc complétée

et permettrait un choix de constante optimal. Il parait donc important de tester l'ensemble de la programmation sur les machines permettant des comparaisons, de par l'expérience du laboratoire, tel que la MIKRON UCP710 sur laquelle une commande numérique ouverte est installée. Cela mettrait en valeur l'aspect lisse des trajectoires obtenues d'un point de vue capacités machine.

Bibliographie

[APT, 1983] (1983). *Part Programming Manual Format*. APT360.

[Abiyev et al., 2010] Abiyev, R., Ibrahim, D., and Erin, B. (2010). Navigation of mobile robots in the presence of obstacles. *Advances in engineering Software*, 41 :1179–1186.

[Ahmad et al., 2011] Ahmad, R., Tichadou, S., and Hascoet, J. (2011). New computer vision based snakes and ladders algorithm for the safe trajectory of two axis cnc machines. *Computer-Aided Design*, 44 :355–366.

[Arrasa et al., 2002] Arrasa, K., Perssonb, J., Tomatisa, N., and Siegwarta, R. (2002). Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In *International Conference on Robotics and Automation*.

[Asakawa and Kanjo, 2013] Asakawa, N. and Kanjo, Y. (2013). Collision avoidance of a welding robot for a large structure. *International Journal of Automation Technology*, 7(2) :190–195.

[Balasubramaniam et al., 2002] Balasubramaniam, M., Ho, S., and Sarma, S. (2002). Generation of collision-free 5-axis tool paths using a haptic surface. *Computer-Aided Design*, 34 :267–279.

[Balasubramaniam et al., 2000] Balasubramaniam, M., Laxmiprasad, P., Sarma, S., and Shaikh, Z. (2000). Generating 5-axis nc roughing paths directly from a tesselated representation. *Computer-Aided Design*, 32(4) :261–277.

[Balasubramaniam et al., 2003] Balasubramaniam, M., Sarma, S., and Marciniak, K. (2003). Collision-free finishing toolpaths from visibility data. *Computer-Aided Design*, 35 :359–374.

[Barequet and Peled, 2001] Barequet, G. and Peled, S. H. (2001). Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Journal of Algorithms*, 38 :91–109.

[Bentes and Saotome, 2012] Bentes, C. and Saotome, O. (2012). Dynamic swarm formation with potential fields and a* path planning in 3d environment. *Brazilian Robotics Symposium and Latin American Robotics Symposium*.

[Bergen, 1998] Bergen, G. V. D. (1998). Efficient collision detection of complex deformable models using aabb trees.

[Beudaert, 2013] Beudaert, X. (2013). *Commande numérique ouverte : interpolation optimisée pour l'usinage 5 axes grande vitesse des surfaces complexes*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Bi et al., 2010] Bi, Q., Wang, Y., and Zhu, L. (2010). Generating collision-free tool orientations for 5-axis nc machining with a short ball-end cutter. *International Journal of Production Research*, 48(24) :7337–7356.

[Bohez et al., 2003] Bohez, L., Minh, N. T. H., and Kiatsrithanakorn, B. (2003). The stencil buffer sweep plane algorithm for 5-axis cnc tool path verification. *Computer-Aided Design*, 35 :1129–1142.

[Borenstein and Koren, 1989] Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE transactions on systems, man, and cybernetics*, 19(5) :1179–1187.

[Brock and Khatib, 1998] Brock, O. and Khatib, O. (1998). Mobile manipulation : Collision-free path modification and motion coordination.

[Castagnetti, 2005] Castagnetti, C. (2005). Usinage des aubes de roues de machines hydrauliques. Master's thesis, Ecole Normale Supérieure de Cachan.

[Castagnetti, 2008] Castagnetti, C. (2008). *Amélioration du comportement cinématique des machines par l'optimisation des trajets d'usinage 5 axes*. PhD thesis, Université Blaise Pascal - Clermont II.

[Castagnetti et al., 2008] Castagnetti, C., Duc, E., and Ray, P. (2008). The domain of admissible orientation concept : A new method for five-axis tool path optimisation. *Computer-Aided Design*, 40 :938–950.

[Chang et al., 2009] Chang, J., Wang, W., and Kim, M. (2009). Efficient collision detection using a dual obb-sphere bounding volume hierarchy. *Computer-Aided Design*.

[Chaves-Jacob, 2009] Chaves-Jacob, J. (2009). *Développement d'une méthodologie de réduction des défauts géométriques : application à l'usinage 5- axes de composants de turbomachine*. PhD thesis, Arts et Métiers Paritech.

[Chen et al., 2009] Chen, H., Kuo, H., and Tsay, D. (2009). Removing tool marks of blade surfaces by smoothing five-axis point milling cutter paths. *Journal of Materials Processing Technology*, 209 :5810–5817.

[Chen and Tang, 2006] Chen, K. and Tang, K. (2006). Reliable haptic system for collision-free 5-axis tool path generation. *Computer-Aided Design Applications*, 3(5) :629–638.

[Chen et al., 2005a] Chen, T., Ye, P., and Wang, J. (2005a). Local interference detection and avoidance in five-axis nc machining of sculptured surfaces. *International Journal of Advanced Manufacturing Technology*, 25 :343–349.

[Chen et al., 2005b] Chen, Y., Yang, Z., and Lian, L. (2005b). On the development of a haptic system for rapid product development. *Computer-Aided Design*, 37 :559–569.

[Cho et al., 2012] Cho, H., Park, J., and Chung, Y. (2012). Computation of optimal tool length for 5-axis ball-ended milling of molding die. *Journal of Mechanical Science and Technology*, 26(10) :3097–3101.

[Cho et al., 1997] Cho, I., Lee, K., and Kim, J. (1997). Generation of collision-free cutter location data in five-axis milling using the potential energy method. *International Journal of Advanced Manufacturing Technology*, 13(8) :523–529.

[Choi et al., 1997] Choi, B., Kim, D., and Jerard, R. (1997). C-space approach to tool-path generation for die and mould machining. *Computer-Aided Design*, 29(9) :657–669.

[Choi et al., 1993] Choi, B., Park, J., and Jun, C. (1993). Cutter-location data optimization in 5-axis surface machining. *Computer-Aided Design*, 25(6) :377–386.

[Cohen et al., 1995] Cohen, J., Lin, M., Manocha, D., and Ponamgi, M. (1995). I-collide : An interactive and exact collision detecyion system for large-scale environments. *In Proc. of ACM Interactive 3D Graphics Conferences*, pages 189–196.

[Ding et al., 2004] Ding, S., Mannan, M., and Poo, A. (2004). Oriented bounding box and octree based global interference detection in 5-axis machining of free-form surfaces. *Computer-Aided Design*, 36 :1281–1294.

[Duc, 1998] Duc, E. (1998). *Usinages des formes gauches, contribution à l'amélioration de la qualité des trajectoires d'usinage*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Faverjon and Tournassoud, 1987] Faverjon, B. and Tournassoud, P. (1987). A local based approach for path planning of manipulators with a high number of degrees of freedom.

[Ge and Cui, 2000] Ge, S. and Cui, Y. (2000). New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, 16(5) :615–620.

[Ge and Cui, 2002] Ge, S. and Cui, Y. (2002). Dynamic motion planning for mobile robots using potential fiel method. *Automous Robots*, 13 :207–222.

[Gian et al., 2003] Gian, R., Lin, T., and Lin, A. (2003). Planning of tool orientation for five axis cavity machining. *International Journal of Advanced Manufacturing Technology*, 22 :150–160.

[Girardeau-Montaut, 2006] Girardeau-Montaut, D. (2006). *Détection de changement sur des données géométriques tridimensionnelles*. PhD thesis, École Nationale Supérieure des Télécommunications École Nationale Supérieure des Télécommunications École Nationale Supérieure de Télécommunications.

[Gottschalk et al., 1996] Gottschalk, S., Lin, M. C., and Manocha, D. (1996). Obbtree : A hierarchical structure for rapid interference detection.

[Gray et al., 2003] Gray, P., Bedi, S., and Ismail, F. (2003). Rolling ball method for 5-axis surface machining. *Computer-Aided Design*, 35 :347–357.

[Ho et al., 2003] Ho, M., Hwang, Y., and Hu, C. (2003). Five-axis tool orientation smoothing using quaternion interpolation algorithm. *International Journal of Machine Tools Manufacture*, 43 :1259–1267.

[Ho et al., 2001] Ho, S., Sarma, S., and Adachi, Y. (2001). Real-time interference analysis between a tool and an environment. *Computer-Aided Design*, 33 :935–947.

[Hosseinkhania et al., 2007] Hosseinkhania, Y., Akbaria, J., and Vafaeesefatb, A. (2007). Penetration-elimination method for five-axis cnc machining of sculptured surfaces. *International Journal of Machine Tools Manufacture*, 47 :1625–1635.

[Hsueh et al., 2007] Hsueh, Y., Hsueh, M., and Lien, H. (2007). Automatic selection of cutter orientation for preventing the collision problem on a five-axis machining. *International Journal of Advanced Manufacturing Technology*, 32 :66–77.

[Hu et al., 2013] Hu, P., Tang, K., and Lee, C. (2013). Global obstacle avoidance and minimum workpiece setups in five-axis machining. *Computer-Aided Design*, 45(10) :1222–1237.

[Hwang and Ahuja, 1992] Hwang, Y. and Ahuja, N. (1992). A potential field approach to path planning. *IEEE Transactions on robotics and automation*, 8(1) :1042–1052.

[Ilushin et al., 2005] Ilushin, O., Elber, G., and Halperin, D. (2005). Precise global collision detection in multi-axis nc-machining. *Computer-Aided Design*, 37 :909–920.

[Jang et al., 2000] Jang, D., Kim, K., and Jung, J. (2000). Voxel-based virtual multi-axis machining. *International Journal of Advanced Manufacturing Technology*, 16 :709–713.

[Jun et al., 2003] Jun, C., Cha, K., and Lee, Y. (2003). Optimizing tool orientation for 5-axis machining by configuration-space search method. *Computer-Aided Design*, 35(6) :549–566.

[Jun et al., 2002] Jun, C., Kim, D., and Park, S. (2002). A new curve-based approach to polyhedral machining. *Computer-Aided Design*, 34 :379–389.

[Kaufman et al., 1993] Kaufman, A., D.Cohen, and Yagel, R. (1993). Volume graphics. *IEEE Computer*, 6(7) :51–64.

[Khatib, 1980] Khatib, O. (1980). *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. PhD thesis, Ecole Nationale Supérieure de l'Aéoronautique et de l'Espace.

[Khatib, 1986] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1) :90–98.

[Khatib and Maitre, 1978] Khatib, O. and Maitre, J. L. (1978). Dynamic control of manipulators operating in a complex environment. *http ://cs.stanford.edu/groups/manips/images/pdfs/Khatib*$_1$978.*pdf*.

[Kitney et al., 1989] Kitney, R., Moura, L., and Straughan, K. (1989). 3-d visualization of arterial structures using ultrasound and voxel modelling. *International Journal of Cardiac Imaging*, 4 :135–143.

[Koren and Borenstein, 1991] Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *IEEE Transactions on robotics and automation*, pages 1398–1404.

[Lauwers et al., 2001] Lauwers, B., Dejonghe, P., and Kruth, J. (2001). Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation. *Computer-Aided Design*, 35(5) :421–432.

[Lee and Lee, 2007] Lee, J. and Lee, R. (2007). Interference-free toolpath generation using enveloping element for five-axis machining of spatial cam. *Journal of Materials Processing Technology*, (187-188) :10–13.

[Lee, 1997] Lee, Y. (1997). Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. *Computer-Aided Design*, 29(7) :507–521.

[Lee and Chang, 1995] Lee, Y. and Chang, T. (1995). 2-phase approach to global tool interference avoidance in 5-axis machining. *Computer-Aided Design*, 27(10) :715–729.

[Lee et al., 2000] Lee, Y., Ma, Y., and Jegadesh, G. (2000). Rolling-ball method and contour marching approach to identifying critical regions for complex surface machining. *Computer in Industry*, pages 163–180.

[Leon, 1991] Leon, J.-C. (1991). *Modélisation et construction de surfaces pour la CFAO*. Hermes.

[Li and Jerard, 1994] Li, S. and Jerard, R. (1994). 5-axis machining of sculptured surfaces with a flat-end cutter. *Computer-Aided Design*, 26(3) :165–178.

[Lin et al., 2014] Lin, Z., Fu, J., Shen, H., and Gan, W. (2014). Non-singular tool path planning by translating tool orientations in c-space. *International Journal of Advanced Manufacturing Technology*.

[Liua et al., 2004] Liua, X., Doddsb, G., McCartneyc, J., and Hindsc, B. (2004). Virtual designworks—designing 3d cad models via haptic interaction. *Computer-Aided Design*, 36 :1129–1140.

[Lozano-Perez, 1980] Lozano-Perez, T. (1980). Spatial planning : A configuration space approach.

[Makki, 2009] Makki, A. (2009). *Copiage rapide de formes sur machine outil à commande numérique en fraisage 5 axes positionne*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Makki et al., 2006] Makki, M., abd F. Thiebaut, S. L., Tournier, C., and LartigueLartigue, C. (2006). Identification de la géométrie de machines cn par mesure d'artefacts. In *1er Congrès International Francophone de Mécanique Avancée*, Université d'Alep, Syrie,. aculté de Génie Mécanique.

[Mbede et al., 2000] Mbede, J., Huang, X., and Wang, M. (2000). Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators. *Robotics and Autonomous Systems*, 32 :61–72.

[Meagher, 1982] Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19 :129–147.

[Monies, 2001] Monies, F. (2001). *Positionnements hors interférence pour l'usinage en bout et en roulant de formes gauches*. PhD thesis, Université Paul Sabatier.

[Monies et al., 2004] Monies, F., Mousseigne, M., and Redonnet, J.-M. (2004). Determining a collision-free domain for the tool in five-axis machining. *International Journal of Production Research*, 42(21) :4513–4530.

[Morikawa et al., 2006] Morikawa, M., Ishida, T., and Teramoto, K. (2006). 5-axis control tool path generation using curved surface interpolation. *JSME International Journal*, 49(4).

[Morishige et al., 1997] Morishige, K., Kase, K., and Takeuchi, Y. (1997). Collision-free tool path generation using 2-dimensional c-space for 5-axis control machining. *International Journal of Advanced Manufacturing Technology*, 13 :393–400.

[Nvidia, 2014] Nvidia (2014). Gpu gems 2 : Chapter 37. octree textures on the gpu.

[Pechard, 2009] Pechard, P. (2009). *Génération de trajectoires d'usinage grande vitesse 5 axes par flanc d'outil : intégration d'un critère de fluidité*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Pixar, 2001] Pixar (2001). Physically based modeling. rigid body simulation.

[Prevost, 2011] Prevost, D. (2011). *Modélisation avancée du couple Machine Outil / Commande Numérique dans un contexte UGV pour l'optimisation du calcul de trajectoires multi axes*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Redonnet et al., 2000] Redonnet, J., Rubio, W., Monies, F., and Dessein, G. (2000). Optimising tool positioning for end-mill machining of free-form surfaces on 5-axis machines for both semi-finishing and finishing. *International Journal of Advanced Manufacturing Technology*, 16 :383–391.

[Rehsteiner and Renker, 1993] Rehsteiner, F. and Renker, H. (1993). Collision-free five-axis milling of twisted ruled surfaces. *Annals of the CIRP*, 42(1) :457–461.

[Roberts and Rawat, 2007] Roberts, C. and Rawat, R. (2007). An approach for tool accessibility evaluation for multi-axis machining models. *Computer-Aided Design Applications*, 4(1-4) :385–394.

[Rubio, 1993] Rubio, W. (1993). *Génération de trajectoires du centre de l'outil pour l'usinage de surfaces complexes sur machines à trois et cinq axes*. PhD thesis, Université Paul Sabatier.

[Rubio et al., 1998] Rubio, W., Lagarrigue, P., Dessein, G., and Pastor, F. (1998). Calculation of tool paths for a torus mill on free-form surfaces on five-axis machines with detection and elimination of interference. *International Journal of Advanced Manufacturing Technology*, 14 :13–20.

[Rubrecht, 2011] Rubrecht, S. (2011). *Contibutions to the control of constrained robots*. PhD thesis, UPMC.

[Schmitz et al., 2007] Schmitz, T., Powell, K., Won, D., Duncan, G., Sawyer, W.-G., and Ziegert, J.-C. (2007). Shrink fit tool holder connection stiffness/damping modeling for frequency response prediction in milling. *International Journal of Machine Tools Manufacture*, 47 :1368–1380.

[Senatore et al., 2005] Senatore, J., Monies, F., Redonnet, J.-M., and Rubio, W. (2005). Analysis of improved positionning in five-axis ruled surface milling using envelope surface. *Computer-Aided Design*, 37(10) :989–998.

[Song and Kumar, 2002] Song, P. and Kumar, R. (2002). A potential field based approach to multi-robot manipulation. *IEEE International Conference on Robotics and Automation*, 2 :1217–1222.

[Takeuchi et al., 1995] Takeuchi, Y., Morishige, K., Yokoyama, M., and Hisaki, T. (1995). 5-axis control machining using side-milling cutter. *Journal of Advanced Automation Technology*, 7(1) :60– 65.

[Tarbutton et al., 2013] Tarbutton, J., Kurfess, T., Tucker, T., and Konobrytskyi, D. (2013). Gouge-free voxel-based machining for parallel processors. *The International Journal of Advanced Manufacturing Technology*, 69(9-12) :1941–1953.

[Tönshoff and Rackow, 2000] Tönshoff, H. and Rackow, N. (2000). Optimal tool positioning for five- axis flank milling of arbitrary shaped surfaces. *Annals of the German Academic Society for Production Engineering (WGP)*, 7(1) :57–60.

[Tournier, 2001] Tournier, C. (2001). *Contribution à la conception des formes complexes : La surface d'usinage en fraisage 5 axes isocrête*. PhD thesis, Ecole Normale Supérieure de Cachan.

[Ulrich and Borenstein, 2000] Ulrich, I. and Borenstein, J. (2000). Vfh* : Local obstacle avoidance with look-ahead verification. *IEEE International Conference on Robotics and Automation*, pages 2505–2511.

[Umehara et al., 2006] Umehara, T., Teramoto, K., and Ishida, T. (2006). Tool posture determination for 5-axis control machining by area division method. *JSME International Journal*, 49(1) :35–42.

[Volpe and Khosla, 1990] Volpe, R. and Khosla, P. (1990). Manipulator control with superquadric artificial potential functions : Theory and experiment. *IEEE transactions on systems, man, and cybernetics*, 20.

[Wang and Tang, 2007] Wang, N. and Tang, K. (2007). Automatic generation of gouge-free and angular-velocity-compliant five-axis toolpath. *Computer-Aided Design*, 39(10) :841–852.

[Wang and Tang, 2008] Wang, N. and Tang, K. (2008). Five-axis tool path generation for a flat-end tool based on iso-conic partioning. *Computer-Aided Design*, 40(12) :1067–1079.

[Wang et al., 2006] Wang, Q., Li, J., and Zhou, R. (2006). Graphics-assisted approach to rapid collision detection for multi-axis machining. *International Journal of Advanced Manufacturing Technology*, 30 :853–863.

[Warren, 1989] Warren, C. (1989). Global path planning using artificial potential fields. *IEEE*.

[Woo, 1994] Woo, T. (1994). Visibility maps and spherical algorithms. *Computer-Aided Design*, 26(1) :6–16.

[Yau and Tsou, 2009] Yau, H. and Tsou, L. (2009). Efficient nc simulation for multi-axis solid machining with a universal apt cutter. *Journal of Computing and Information Science in Engineering*, 9.

[Zhiwei et al., 2012] Zhiwei, L., Hongyao, S., and Wenfeng, G. (2012). Approximate tool posture collision-free area generation for five-axis cnc finishing process using admissible area interpolation. *International Journal of Advanced Manufacturing Technology*, 62 :1191–1203.

[Zhu and Lee, 2004a] Zhu, W. and Lee, Y. (2004a). Dexel-based force–torque rendering and volume updating for 5-dof haptic product prototyping and virtual sculpting. *Computers in Industry*, 55 :125–145.

[Zhu and Lee, 2004b] Zhu, W. and Lee, Y. (2004b). Five-axis pencil-cut planning and virtual prototyping with 5-dof haptic interface. *Computer-Aided Design*, 36 :1295–1307.

Annexes

Annexe 1 : Méthode d'interpolation RK4

Présentation théorique de la méthode

La méthode de Runge-Kutta RK4 permet de résoudre un système d'équations différentielles ordinaires de la manière suivante.

Soit une équation différentielle du premier ordre donnée en équation (4.15) :

$$\frac{dy}{dx} = f(x,y) \tag{4.15}$$

La méthode RK4 utilise plusieurs points intermédiaires pour calculer la valeur de y_{i+1} à partir de la valeur y_i. Soit un point intermédiaire A d'abscisse $x_i + \frac{h}{2}$ dont la valeur de l'ordonnée est donnée par l'équation (4.16) :

$$y_{iA} = y_i + (\frac{dy}{dx})_i.\frac{h}{2} \tag{4.16}$$

Soit l'équation (4.17) :

$$y_{iA} - y_i = (\frac{dy}{dx})_i.\frac{h}{2} = \frac{m_1}{2} \tag{4.17}$$

Puis un point B d'ordonnée noté en équation (4.18) :

$$y_{iB} = y_i + (\frac{dy}{dx})_{iA}.\frac{h}{2} \tag{4.18}$$

Soit la relation (4.19)

$$y_{iB} - y_i = (\frac{dy}{dx})_{iA}.\frac{h}{2} = \frac{m_2}{2} \tag{4.19}$$

L'ordonnée d'un point C d'abscisse $x_i + h$ est alors calculée à l'aide de la relation (4.20) :

$$y_{iC} = y_i + (\frac{dy}{dx})_{iB}.h \tag{4.20}$$

Soit (4.21) :

$$y_{iC} - y_i = (\frac{dy}{dx})_{iB}.h = m_3 \tag{4.21}$$

Soit $(\frac{dy}{dx})_{iC}$ la valeur de $(\frac{dy}{dx})$ au point C. En posant comme noté en équation (4.22) :

$$(\frac{dy}{dx})_{iC}.h = m_4 \tag{4.22}$$

L'ordonnée définitive y_{i+1} du point d'abscisse $x_i + h$ est donnée par les relations (4.23) (4.24) :

$$y_{i+1} = y_i + \frac{1}{6}.((\frac{dy}{dx})_i + 2.(\frac{dy}{dx})_{iA} + 2.(\frac{dy}{dx})_{iB} + (\frac{dy}{dx})_{iC})).h \tag{4.23}$$

$$y_{i+1} = y_i + \frac{1}{6}.(m_1 + 2.m_2 + 2.m_3 + m_4) \tag{4.24}$$

La méthode de Runge Kutta d'ordre 4 définit deux suites, h étant le pas de discrétisation souhaité en x. La première permet de définir les valeurs de x :

— terme initial : x_0,
— relation de récurrence : $x_{i+1} = x_i + h$.

La deuxième permet d'évaluer les valeurs de y :

— terme initial : y_0,
— relation de récurrence : $y_{i+1} = y_i + \frac{1}{6}.(m_1 + 2.m_2 + 2.m_3 + m_4)$,
— avec,
  — $m_1 = h.f(x_i,y_i)$,
  — $m_2 = h.f(x_i + h/2,y_i + m_1/2)$,
  — $m_3 = h.f(x_i + h/2,y_i + m_2/2)$,
  — $m_4 = h.f(x_i + h/2,y_i + m_3)$.

L'idée est que la valeur suivante y_{i+1} est approchée par la somme de la valeur actuelle y_i et du produit de la taille de l'intervalle h par la pente estimée. La pente est obtenue par une moyenne pondérée de pente :

— m_1 est la pente au début de l'intervalle,
— m_2 est la pente au milieu de l'intervalle, en utilisant la pente m_1 pour calculer la valeur de y au point $x_i + h/2$ par le biais de la méthode d'Euler,
— m_3 est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente m_2 pour calculer y,
— m_4 est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant m_3.

Dans la moyenne des 4 pentes, un poids plus important est donné aux pentes au point milieu comme illustré en figure 2.7 et noté en équation (4.25).

$$pente = \frac{m_1 + 2.m_2 + 2.m_3 + m_4}{6} \tag{4.25}$$

La méthode RK4 est une méthode d'ordre 4, ce qui veut dire que l'erreur commise à chaque étape est de l'ordre de h^5, alors que l'erreur totale accumulée est de l'ordre de h^4.

Problème du premier ordre, comparaison à la solution analytique

Cet exemple va permettre de vérifier l'efficacité de la méthode RK4 sur un système dont il est possible de trouver la solution analytique. Soit l'équation différentielle (4.26) :

$$\frac{dy}{dx} = -2.x.y \qquad \text{avec} \qquad y_{(x=0)} = 1 \qquad \text{et} \qquad \Delta x = 0{,}1 \tag{4.26}$$

La solution analytique est donnée en équation (4.27) :

$$y = e^{-x^2} \tag{4.27}$$

Ce qui donne le tableau de solutions 4.3.

i	x_i	y_i
0	0	1
1	0,1	0,9900
2	0,2	0,9608
3	0,3	0,9139
4	0,4	0,8521
5	0,5	0,7788

Tableau 4.3 – Partie des solutions

La méthode RK4 s'applique comme suit : (avec $y_{i+1} = y_i + \frac{1}{6} * (m_1 + 2 * m_2 + 2 * m_3 + m_4)$) et donne :

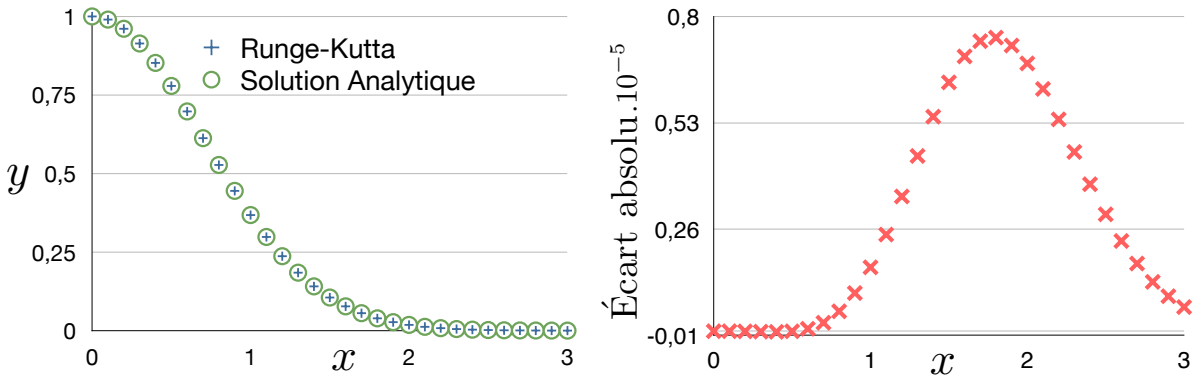
— $m_1 = -2 * x_i * y_i * \Delta x$,
— $m_2 = -2 * (x_i + \Delta x/2) * (y_i + m_1/2) * \Delta x$,
— $m_3 = -2 * (x_i + \Delta x/2) * (y_i + m_2/2) * \Delta x$,
— $m_4 = -2 * (x_i + \Delta x) * (y_i + m_3) * \Delta x$.

Les résultats obtenus par la méthode RK4 sont donnés dans le tableau 4.4.

i	x_i	m_1	m_2	m_3	m_4	y_i
0	0	0	-0,01	-0,00995	-0,0199	1
1	0,1	0,0198	-0,0294	-0,02926	-0,0384	0,9900
2	0,2	...	...	...	...	0,96
3	0,3	...	...	...	...	...
4	0,4	...	...	...	...	...
5	0,5	...	...	...	...	...

Tableau 4.4 – Partie des solutions

Si x varie de 0 à 3 pour illustrer cet exemple le résultat obtenu est tracé en figure 4.22 où il est possible de voire une correspondance entre les deux modèles.



(a) (b)

Figure 4.22 – Résultats obtenus

Cette comparaison permet de valider la cohérence de l'utilisation de la méthode RK4 par rapport à une solution analytique connue. L'analyse de l'écart relatif permet de se faire une idée de l'ordre de grandeur de l'erreur commise qui n'était pas visible sur la première figure. Par la suite, l'application de cette méthode d'interpolation est réalisée sur une équation différentielle du second ordre.

Formalisation d'un problème du second ordre pour RK4

Cet exemple va montrer comment traiter une équation différentielle du second ordre avec la méthode RK4. L'exemple du système ressort amortisseur est cohérent avec les travaux présentés au court de cette thèse. La figure 4.23 illustre le paramétrage sur lequel le problème est traité.

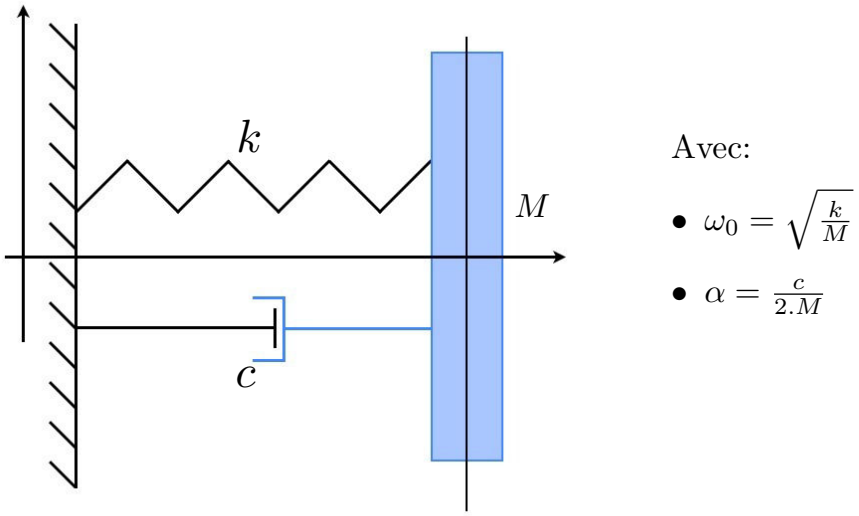


Figure 4.23 – Modélisation du problème

Soit l'équation différentielle (4.28) qui traduit le mouvement de ce système :

$$\ddot{y} + 2.\alpha.\dot{y} + \omega_0^2.y = 0 \tag{4.28}$$

La résolution par la méthode RK4 est effectuée comme suit. L'équation de base est connue ainsi que les valeurs données en équation (4.29) :

$$\alpha = 1s^{-1} \qquad v_{(t=0)} = 0m/s \qquad y_{(t=0)} \qquad \omega_0^2 = \text{variable} \qquad dt = 0{,}10s \tag{4.29}$$

La première étape de ce travail est de décomposer l'équation du second ordre en 2 équations dépendantes. Le système d'équations devient alors celui donné en équation (4.30) :

$$\begin{cases} v = \frac{dy}{dt} \\ \frac{dv}{dt} = -2.\alpha.v - \omega_0^2.y \end{cases} \tag{4.30}$$

La méthode présentée précédemment est alors appliquée à ces deux équations en jouant avec trois paramètres (t_i, y_i, v_i) :

$$v_{(i+1)} = v_{(t+dt)} = v_t + \frac{1}{6}.(m_1 + 2.m_2 + 2.m_3 + m_4) \tag{4.31}$$

$$y_{(i+1)} = y_{(t+dt)} = y_t + \frac{1}{6}.(n_1 + 2.n_2 + 2.n_3 + n_4) \tag{4.32}$$

Avec :

— $n_1 = v_t.dt$,
— $m_1 = (-2.\alpha.v_t - \omega_0^2.y_t).dt)$,
— $n_2 = (v_t + \frac{m_1}{2}).dt$,
— $m_2 = (-2.\alpha.(v_t + \frac{m_1}{2}) - \omega_0^2.(y_t + \frac{n_1}{2})).dt$,
— $n_3 = (v_t + \frac{m_2}{2}).dt$,
— $m_3 = (-2.\alpha.(v_t + \frac{m_2}{2}) - \omega_0^2.(y_t + \frac{n_2}{2})).dt$,
— $n_4 = (v_t + m_3).dt$,
— $m_4 = (-2.\alpha.(v_t + m_3) - \omega_0^2.(y_t + n_3)).dt$.

Il existe alors plusieurs types de solutions en fonction du choix de l'amortisseur.

Régime pseudo-périodique

Le régime pseudo-périodique est obtenu si l'inéquation (4.33) est respectée.

$$\alpha^2 - \omega_0^2 < 0 \tag{4.33}$$

L'exemple est programmé pour $\omega_0^2 = 50$

Régime apériodique critique

Le régime apériodique critique est obtenu si l'équation (4.34) est respectée.

$$\alpha^2 - \omega_0^2 = 0 \tag{4.34}$$

L'exemple est programmé pour $\omega_0^2 = 1$

Régime apériodique

Le régime apériodique est obtenu si l'équation (4.35) est respectée.

$$\alpha^2 - \omega_0^2 > 0 \tag{4.35}$$

L'exemple est programmé pour $\omega_0^2 = 0{,}4$.

Les résultats obtenus par l'utilisation de Runge-Kutta (RK4) sont illustrés sur la figure 4.24.

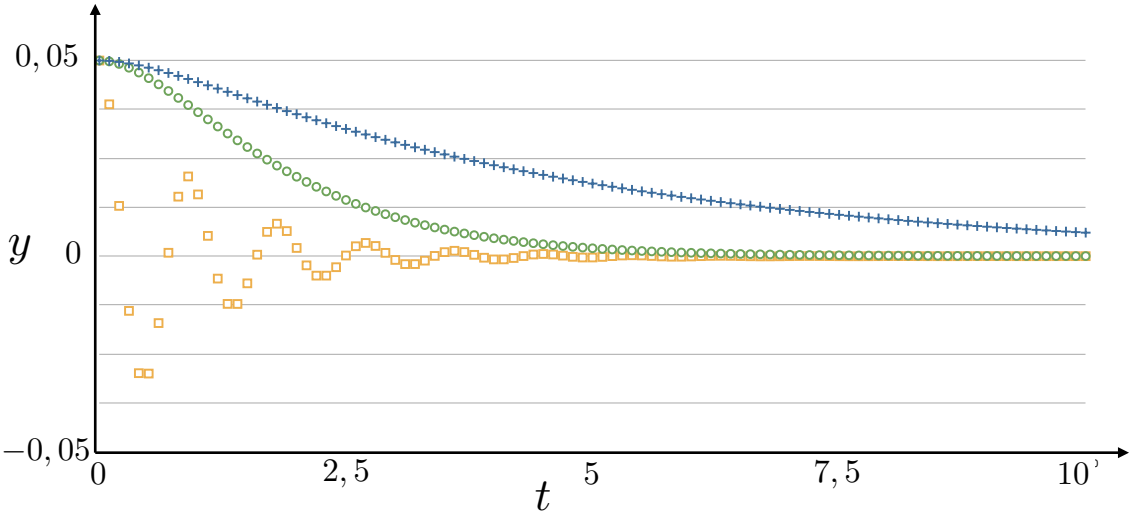


Figure 4.24 – Résultats obtenus

Ces derniers démontrent la capacité de la méthode de Runge-Kutta à résoudre une équation du second ordre avec la distinction attendue entre les différents régimes connus dans la théorie. Cette annexe permet de valider l'utilisation de cette programmation lors de ces travaux de thèse.

Annexe 2 : Surfaces de Bézier

Définition

Un carreau de Bézier peut être considéré comme le lieu d'une courbe $C(u)$ subissant un déplacement et une déformation. La courbe $C(u)$ (ou génératrice) est une courbe de Bézier définie par son polygone caractéristique PG_m (donnée en équation (4.36)).

$$C(u) = \sum_{i=0}^{m} B_{im}(u) P_i \tag{4.36}$$

Chaque sommet P_i de PG_m décrit une trajectoire $D_i(v)$ (la directrice donnée en équation (4.37)) modélisée également par une courbe de Bézier de polygone caractéristique PG_{in}.

$$D_i(v) = \sum_{j=0}^{n} B_{jn}(v) P_{ij} \tag{4.37}$$

Les sommets P_{i0} coïncident avec les sommets P_i de PG_m. L'équation paramétrique de la surface est alors donnée en équation (4.38) :

$$Q(u,v) = \sum_{i=0}^{m} B_{im}(u) D_i(v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_{im}(u).B_{jn}(v).P_{ij} \tag{4.38}$$

Les sommets P_{ij} constituent le réseau caractéristique avec la surface obtenue illustrée en figure 4.25.

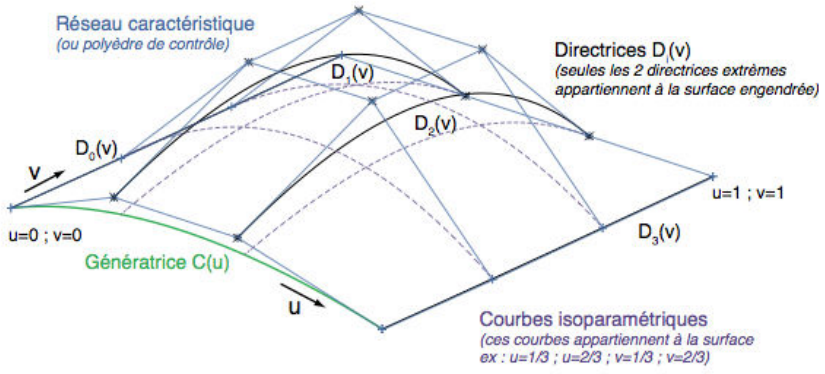


Figure 4.25 – Surface de Bézier engendrée par la génératrice $C(u)$ de degrés 3 et les directrices $D_i(v)$ de degrés 3

Coordonnées des points de contrôles utilisés

La surface de Bézier obtenue dans le chapitre 3 en figure 3.10 et rappelé ci-dessous.

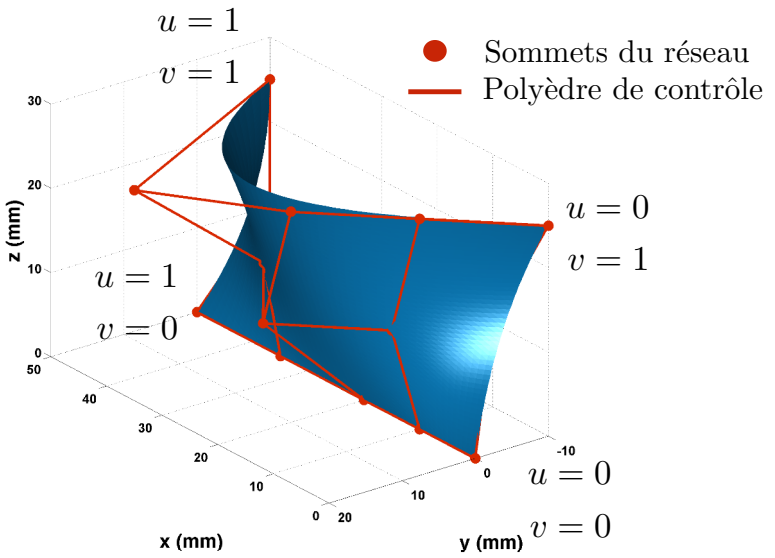


Figure 4.26 – Modélisation de la surface considérée comme obstacle lors de cette étude (*Check-Surface*)

Pour information, cette dernière est réalisée à partir des points de contrôles exprimés dans le tableau 4.5 (exprimés en *mm*).

$j\|i$	P_{0j}	P_{1j}	P_{2j}	P_{3j}	P_{4j}
P_{i0}	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}10\\0\\0\end{pmatrix}$	$\begin{pmatrix}20\\0\\0\end{pmatrix}$	$\begin{pmatrix}35\\0\\0\end{pmatrix}$	$\begin{pmatrix}50\\0\\0\end{pmatrix}$
P_{i1}	$\begin{pmatrix}10\\-10\\10\end{pmatrix}$	$\begin{pmatrix}15\\0\\10\end{pmatrix}$	$\begin{pmatrix}25\\10\\10\end{pmatrix}$	$\begin{pmatrix}38\\0\\10\end{pmatrix}$	$\begin{pmatrix}50\\-10\\10\end{pmatrix}$
P_{i2}	$\begin{pmatrix}0\\-10\\25\end{pmatrix}$	$\begin{pmatrix}10\\0\\25\end{pmatrix}$	$\begin{pmatrix}20\\10\\25\end{pmatrix}$	$\begin{pmatrix}35\\20\\25\end{pmatrix}$	$\begin{pmatrix}50\\-10\\25\end{pmatrix}$

Tableau 4.5 – Coordonnées des points de contrôles utilisés

Annexe 3 : Interpolation linéaire permettant l'obtention des temps de calculs

L'interpolation réalisée pour obtenir un temps de calcul complet a été réalisée en mode linéaire en connaissant les valeurs exactes de simulation données dans le tableau 3.2. Ces derniers permettent d'obtenir une valeur approchée du temps de calcul pour un usinage complet. La valeur finale est obtenue comme notée en équation (4.39) avec une prise en compte des sept pâles considérées comme identiques.

$$\text{TOTAL} = \sum_{v_i} (\text{Temps de simulation}(v_I)) * 7 \tag{4.39}$$

TPS DE CALCUL	Sur le point G	Sur le maillage outil
Abscisse curviligne v*		
0,2	5,1	586,02
0,21	5,03	568,645
0,22	4,96	551,27
0,23	4,89	533,895
0,24	4,82	516,52
0,25	4,75	499,145
0,26	4,68	481,77
0,27	4,61	464,395
0,28	4,54	447,02
0,29	4,47	429,645
0,3	4,4	412,27
0,31	4,35	405,144
0,32	4,3	398,018
0,33	4,25	390,892
0,34	4,2	383,766
0,35	4,15	376,64
0,36	4,1	369,514
0,37	4,05	362,388
0,38	4	355,262
0,39	3,95	348,136
0,4	3,9	341,01
0,41	3,89	341,74
0,42	3,88	342,47
0,43	3,87	343,2
0,44	3,86	343,93
0,45	3,85	344,66
0,46	3,84	345,39
0,47	3,83	346,12
0,48	3,82	346,85
0,49	3,81	347,58
0,5	3,8	348,31

(a)

0,51	3,78	356,577
0,52	3,76	364,844
0,53	3,74	373,111
0,54	3,72	381,378
0,55	3,7	389,645
0,56	3,68	397,912
0,57	3,66	406,179
0,58	3,64	414,446
0,59	3,62	422,713
0,6	3,6	430,98
0,61	3,56	445,983
0,62	3,52	460,986
0,63	3,48	475,989
0,64	3,44	490,992
0,65	3,4	505,995
0,66	3,36	520,998
0,67	3,32	536,001
0,68	3,28	551,004
0,69	3,24	566,007
0,7	3,2	581,01
0,71	3,17	592,578
0,72	3,14	604,146
0,73	3,11	615,714
0,74	3,08	627,282
0,75	3,05	638,85
0,76	3,02	650,418
0,77	2,99	661,986
0,78	2,96	673,554
0,79	2,93	685,122
0,8	2,9	696,69
TEMPS TOTAL	1631	197334,935

(b)

Figure 4.27 – Temps de calcul pour l'usinage complet de l'*impeller*

Annexe 4 : Résultats des simulations avec des droites représentant les trajectoires avec une interpolation cubique

Droite alignée dans la veine

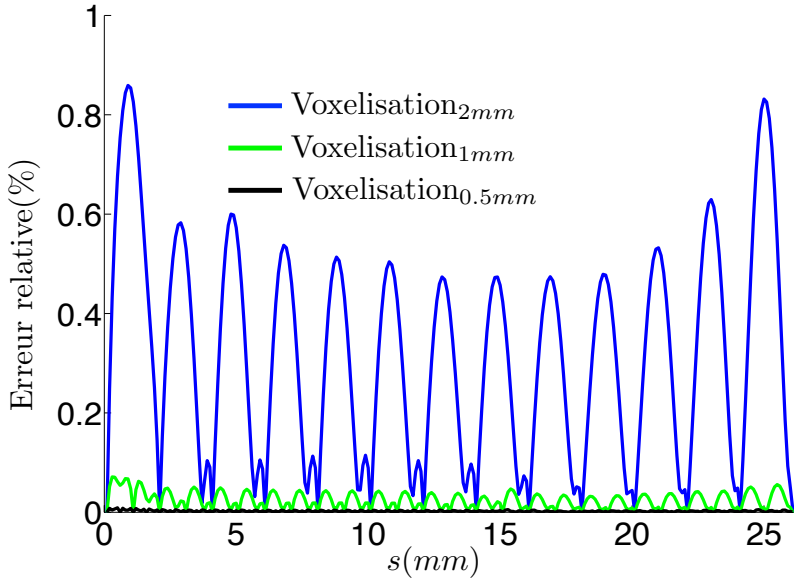


Figure 4.28 – Erreur relative de l'effort répulsif obtenu

Droite perpendiculaire aux surfaces obstacles

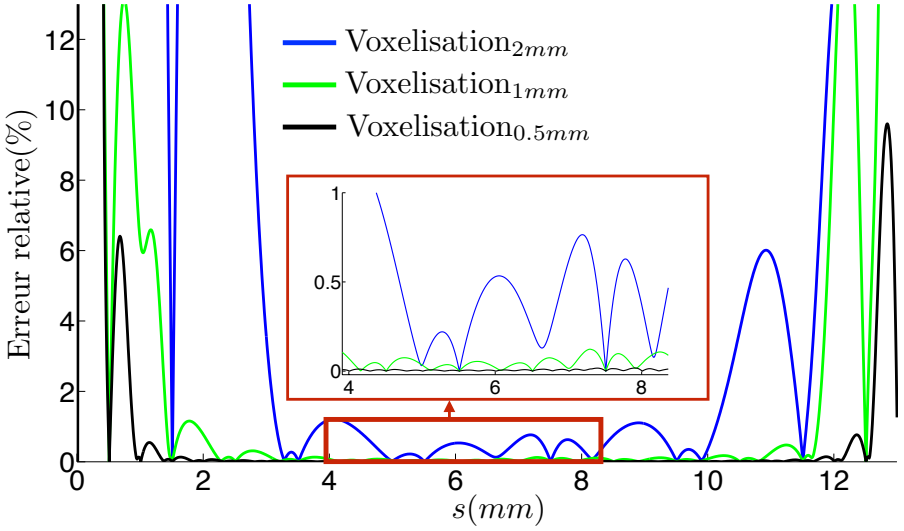


Figure 4.29 – Erreur relative de l'effort répulsif obtenu

Modélisation mécanique intégrant des champs répulsifs
pour la génération de trajectoires 5 axes hors collision

Résumé : Le processus de réalisation des pièces de formes complexes par usinage est un processus essentiel dans les domaines de l'aéronautique, de l'automobile, des moules et des matrices. Alors que l'usinage 5 axes est maintenant répandu dans les grands groupes industriels, il reste plusieurs problématiques à traiter. L'évitement de collisions le long de la trajectoire outil programmée en alors traité, notamment au niveau des interférences globales représentant une collision entre l'outil et son environnement.

Classiquement, l'évitement de collisions dans le domaine de l'usinage 5 axes peut être programmé à l'aide d'une analyse géométrique de la situation. Si une collision est détectée, alors une phase de correction et d'optimisation peuvent être utilisée afin d'obtenir une nouvelle trajectoire hors collision. Le but des travaux est alors d'utiliser une modélisation physique afin d'obtenir une trajectoire corrigée hors collision le plus lisse possible. Pour ce faire le mouvement de l'outil est alors étudié d'un point de vue dynamique afin d'éviter les réorientations brutales post correction. De plus, les éléments constituants les obstacles émettent une action répulsive à distance. Cela permet, au cours de la programmation, d'anticiper l'approche d'un obstacle et ainsi d'entamer les corrections d'orientation outil en prévision d'une possible collision.

Cette démarche de modélisation du mouvement étudiée permet alors de réaliser des simulations sur des pièces classiquement usinées dans les domaines énoncés précédemment. Dans le but de généraliser la programmation réalisée, il est alors important de comprendre comment les éléments obstacles sont représentés ainsi que la modélisation retenue pour l'outil utilisé au cours de la simulation.

Enfin, la résolution de la dernière problématique mise en avant au cours de cette thèse concerne les temps de calcul obtenus. Il a été montré, après de multiples simulations, que ces derniers peuvent exploser d'un point de vue combinatoire pour des utilisateurs exigeants (modélisation fine de l'outil et de l'environnement). Une méthode de pré calcul est alors présentée utilisant la voxelisation permettant de diminuer les temps de calcul de manière très importante sans pour autant perdre de manière importante sur la solution obtenue.

Abstract : Although 5-axis free form surfaces machining is commonly proposed in CAD/CAM software, several issues still need to be addressed and especially collision avoidance between the tool and the part. Indeed, advanced user skills are often required to define smooth tool axis orientations along the tool path in high speed machining. In the literature, the problem of collision avoidance is mainly treated as an iterative process based on local and geometrical collision tests. In this paper, an innovative method based on potential fields is used to generate 5-axis collision-free smooth tool paths. In the proposed approach, The ball-end tool is considered as a rigid body moving in 3D space on which repulsive force, deriving from a scalar potential field attached to the check surfaces, and attractive forces are acting. The resolution of the differential equations of the tool motion ensure smooth variations of the tool axis orientation. The proposed algorithm is applied on open pocket parts such as an impeller and a pocket corner to emphasize the effectiveness of this method to avoid collision. After that, it is possible to see that de calculation time can be very importante for a delicate mesh. It is for that, a voxelisation method is developed to decrease these.