

UNIVERSITÉ DE MONTRÉAL

PLANIFICATION DE TRAJECTOIRES POUR PLACEMENT AUTOMATISÉ DE
FIBRES SUR SURFACES DE GÉOMÉTRIES COMPLEXES

CLÉMENT HELY
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
JUILLET 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

PLANIFICATION DE TRAJECTOIRES POUR PLACEMENT AUTOMATISÉ DE
FIBRES SUR SURFACES DE GÉOMÉTRIES COMPLEXES

présenté par : HELY Clément

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GOURDEAU Richard, Ph. D., président

M. BIRGLEN Lionel, Ph. D., membre et directeur de recherche

M. BONEV Ilian, Ph. D., membre

REMERCIEMENTS

En premier lieu, je tiens à remercier mon directeur de recherche, M. Lionel Birglen, qui m'a accordé sa confiance en me donnant l'opportunité de travailler sur ce projet de recherche. Je le remercie également pour ses conseils avisés qui m'ont guidé tout au long de ce travail.

Je souhaite également remercier mes parents et mes proches, qui, de par leurs encouragements, m'ont permis de donner le meilleur de moi-même. Un merci particulier à mes parents pour leur soutien et leur aide dans la révision de ce mémoire.

Je n'oublierai pas mes collègues du laboratoire, Bruno, Jean-Michel et Dmitri qui ont su m'apporter conseils, soutien et divertissement tout au long de ma maîtrise.

Pour finir, je souhaite exprimer une pensée particulière à toutes les personnes que j'ai pu côtoyer grâce à l'athlétisme et qui m'ont aidé à trouver l'équilibre nécessaire pour mener à terme mon double projet scolaire et sportif.

RÉSUMÉ

Durant les 50 dernières années, l'utilisation des matériaux composites pour la fabrication de structures à hautes performances a connu un essor, dû notamment à l'industrie aéronautique. Toujours à la recherche de matériaux plus légers et performants pour la construction d'aéronefs, et devant tenir compte des contraintes de productivité pour rester compétitif, ce secteur a encouragé le développement de nouvelles technologies de production telles que le Placement Automatisé de Fibres (*AFP*). Cette technologie permet d'automatiser la production de structures en matériaux composites à l'aide de pré-imprégnés. Ces derniers se présentent sous forme de bandes composées de tresses de fibres (verre, carbone, etc.) imprégnées de résine semi-polymérisée.

La plupart des cellules d'*AFP* actuellement en service ont été conçues pour la fabrication de larges panneaux plats ou présentant de faibles courbures, comme par exemple les fuselages d'avion. Cependant le secteur de l'aéronautique, mais également les industries de production d'équipement de sport s'intéressent à présent à cette technique pour la fabrication de pièces de plus faibles dimensions et possédant des géométries plus complexes. L'objectif du projet dans lequel s'inscrit le travail présenté ici est donc de mettre au point une cellule de travail et des techniques permettant la production de telles pièces par *AFP*. La recherche présentée dans ce mémoire concerne plus particulièrement le développement de nouvelles méthodes de planification de trajectoires pour *AFP*. La majeure partie de l'étude s'est concentrée sur la génération de celles-ci sur un type de surfaces en particulier : les surfaces de révolution multi-axiales de section circulaire constante telles que des tubes en forme de « Y ».

Plusieurs algorithmes de génération de trajectoires seront présentés au travers de ce mémoire, chacun cherchant à couvrir la surface de la pièce le plus complètement possible afin de produire des objets finaux en un seul tenant. La méthodologie utilisée dans deux de ces algorithmes est de couvrir individuellement chaque branche par des trajectoires dérivant des hélices. Dans un cas l'hélice sera coupée au niveau de la jonction entre les branches (algorithme HD) et dans l'autre cas cette hélice pourra être déformée (algorithme HA) pour épouser les contours de la jonction. Les différentes branches seront ensuite liées entre elles grâce à une trajectoire définie sur l'intersection des cylindres. Ces deux méthodes ont cependant révélé posséder plusieurs défauts qui pourraient compromettre leur utilisation en pratique, ou conduire à la production de pièces présentant des défauts affaiblissant leurs propriétés mécaniques.

Deux autres algorithmes ont donc été développés afin d'améliorer les trajectoires produites.

Dans ces derniers, la méthodologie est de couvrir deux branches du « Y » à l'aide de fibres continues, c'est-à-dire non coupées. Le premier algorithme généralise une technique utilisée dans l'industrie, à savoir définir des couches dont l'orientation des fibres est constante par rapport à une référence ici choisie comme étant l'axe du cylindre sur lequel est effectuée la dépose. Des trajectoires parallèles sont ensuite générées pour couvrir le moule le plus uniformément possible. L'inconvénient majeur de cette méthode est qu'elle peut conduire à l'apparition de trajectoires présentant de fortes courbures au niveau de la jonction, pouvant conduire à des défauts lors de la fabrication.

Pour éliminer ces défauts, un dernier algorithme est présenté. Il permet d'assurer que la courbure des trajectoires ne dépasse pas une valeur limite admissible pour le procédé d'*AFP*. Le respect de cette contrainte entraîne cependant l'impossibilité de prévoir la forme complète de la trajectoire a priori et donc d'assurer une couverture parfaitement uniforme. Il est alors proposé de générer un ensemble de trajectoires couvrant la pièce puis de choisir parmi celles-ci, à l'aide d'un algorithme de sélection, celles qui sont le mieux adaptées à l'utilisation que l'on souhaite faire de la pièce. Une analyse par éléments finis est alors présentée pour aider à la définition des critères de sélection nécessaires à ce dernier algorithme.

Pour finir, des simulations ont été effectuées avec une cellule de travail formée d'un manipulateur et d'une table rotative pour vérifier la faisabilité des trajectoires générées par les différents algorithmes.

ABSTRACT

During the past 50 years, the use of composite materials drastically increase, mainly thanks to the interest of aeronautical industries for these strong and lightweight materials. To improve the productivity of composite materials manufacturing some of the largest aeronautics companies began to develop automated processes such as Automated Fibre Placement (AFP). The AFP workcells currently used by the industry were mainly developed for production of large, nearly flat, plates with low curvatures such as aircraft fuselages. However, the fields of aeronautics and sport goods production begin nowadays to show an interest for manufacturing of smaller and more complex parts. The aim of the project in which this research takes place is to design a new AFP workcell and to develop new techniques allowing production of parts with small size and complex geometry. The work presented in this thesis focuses on the path planning on multi-axial revolution surfaces, e.g. Y-shaped tubes of constant circular cross section.

Several path planning algorithms will be presented aiming at the exhaustive coverage of a mandrel with pre-impregnated (prepreg) composite tape. The methodology used in two of these algorithms is to individually cover each branch of the Y-shaped part with paths deriving from a helix. In the first one, the helix will be cut at the boundary between a branch and the junction region (algorithm HD) while in the second (algorithm HA) the pseudo-helix path can be adjusted to follow this boundary. These two methods were shown to have some drawbacks compromising their practical use and possibly leading to parts with diminished mechanical properties.

To avoid these drawbacks, two others algorithms were developed with a new methodology. With them, the aim is to cover two branches of the Y-shape with a continuous course (i.e. without cut). The first one uses a well known strategy which defines plies with a constant fibre orientation. Parallel paths are then computed to generate a full and uniform ply covering two branches. Once again this method suffers from a main drawback, namely that it can produce highly curved paths leading to manufacturing defects.

To overcome this limitation, a last algorithm is proposed ensuring that the maximal curvature of a trajectory stays below a fixed threshold. However, fulfilling this constraint prevents to predict the complete shape of the path and to ensure a perfectly uniform coverage. It is thus proposed to generate an exhaustive set of trajectories having different shapes and covering all the part. Then, a selection algorithm is used to choose the ones which are best suited according to selection criteria. To help the definition of these criteria, a finite element analysis

is conducted to give some insight concerning the best suited shapes for specific loading cases. Finally, simulations were carried out with a workcell constituted by a robotic manipulator associated with a rotary table to verify the feasibility of the paths generated by the different algorithms.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES SIGLES ET ABRÉVIATIONS	xvi
LISTE DES ANNEXES	xvii
CHAPITRE 1 INTRODUCTION	1
1.1 Composites à renfort fibreux	1
1.2 Problématique	5
1.3 Objectifs de recherche	5
CHAPITRE 2 REVUE DE LITTÉRATURE	8
2.1 Enroulement filamentaire (<i>FW</i>)	8
2.2 Placement de bande automatisé (<i>ATL</i>)	10
2.3 Placement de fibre automatisé (<i>AFP</i>)	11
2.3.1 Principe de fonctionnement	13
2.3.2 Limites du procédé	15
2.3.3 Défauts de fabrication	16
2.4 Principes de planification de trajectoire	18
2.4.1 Méthodes de programmation des robots industriels	18
2.4.2 Méthodes de génération de trajectoires	19
2.4.3 Trajectoires spécifiques à l' <i>AFP</i>	21
CHAPITRE 3 DÉMARCHE DE TRAVAIL	27
CHAPITRE 4 ARTICLE 1 : FEASIBILITY STUDY OF ROBOTIC FIBRE PLACE-	

MENT ON INTERSECTING MULTI-AXIAL REVOLUTION SURFACES	29
4.1 Abstract	29
4.2 Introduction	29
4.3 Segmentation	31
4.3.1 Segmentation in smooth regions	32
4.3.2 Cylinder extraction	33
4.4 Path Planning	33
4.4.1 Constant Placement Angle (CPA)	34
4.4.2 Maximal Curvature Constraint (MCC)	35
4.4.3 Selection Algorithm	39
4.5 Finite Element Analysis	39
4.6 Simulations	41
4.7 Discussion and future works	44
4.8 Conclusion	46
CHAPITRE 5 ASPECTS MÉTHODOLOGIQUES ET RÉSULTATS COMPLÉMEN-	
TAIRES	47
5.1 Directions principales de courbure	47
5.2 Discontinuités aux frontières entre régions	50
5.3 Méthode de l'hélice discontinue (HD)	51
5.4 Méthode de l'hélice ajustable (HA)	53
5.5 Trajectoire sur l'intersection des cylindres	58
5.6 Décalage de la courbe de référence	61
5.7 Calcul de la courbure géodésique	63
5.8 Détection des collisions	65
5.9 Simulations	68
5.10 Description générale du programme de génération de trajectoires	71
CHAPITRE 6 DISCUSSION GÉNÉRALE	73
6.1 Limitation de l'utilisation des directions de courbures principale	73
6.2 Trajectoire sur l'intersection des cylindres	74
6.3 Algorithme <i>CPA</i>	74
6.4 Algorithme <i>MCC</i>	76
6.5 Algorithme de sélection	77
6.6 Simulations	78
CHAPITRE 7 CONCLUSION	81

RÉFÉRENCES 84

ANNEXES 90

LISTE DES TABLEAUX

Tableau 4.1	FEA results	39
-------------	-----------------------	----

LISTE DES FIGURES

Figure 1.1	Fabriqué principalement avec des composites à fibres de carbone, le vélo <i>Argon 18 E119 TRI+</i> possède un cadre de seulement 1,2 kg. (tiré de Argon18)	2
Figure 1.2	Bandes de pré-imprégné (ZOLTEK®)	3
Figure 1.3	Rubans de pré-imprégné (HEXCEL®)	4
Figure 1.4	Procédé de production de pré-imprégné (Tiré de Hexply® Prepreg Technology)	4
Figure 2.1	Différents motifs d'enroulements utilisés en enroulement filamentaire (Shen (1995))	9
(a)	Enroulement polaire	9
(b)	Enroulement hélicoïdal	9
(c)	Enroulement en arceaux	9
Figure 2.2	Portique de placement automatisé de bande (MTorres®)	11
Figure 2.3	Composition du Boeing 787 Dreamliner (Roeseler et al. (2007))	12
Figure 2.4	Cellule d' <i>AFP</i> (Composites)	13
Figure 2.5	Principe de Fonctionnement d'une tête de placement (Brochure Cincinnati VIPER®, Fives)	14
Figure 2.6	Polymérisation sous autoclave (d'après brochure HEXCEL®)	14
Figure 2.7	Flambement des fibres (Tirée de Blom (2010))	15
Figure 2.8	Défauts dus à la longueur de coupe minimale (Blom (2010))	16
Figure 2.9	<i>Gaps</i> et <i>Overlaps</i> (D'après Kim et al. (2015))	16
Figure 2.10	Influence des défauts pour différents cas de chargement (Croft et al. (2011))	18
Figure 2.11	Différents motifs de chemins de couverture	20
(a)	Courbe de Lissajous	20
(b)	Courbe de Péano	20
Figure 2.12	Motif de couverture traditionnels en usinage (Lasemi et al. (2010))	21
(a)	Direction parallèle	21
(b)	Contour parallèle	21
Figure 2.13	Séquence d'empilement possible de couches à orientation de fibre constante	22
Figure 2.14	Génération de la courbe guide par intersection de la surface du moule avec un plan de référence	24

Figure 2.15	Utilisation de la méthode <i>Fast Marching</i> pour la définition de trajectoires équidistantes sur un maillage triangulé (Bruyneel et Zein (2013)). (a) Définition du maillage. (b) Définition de la courbe de référence. (c) Solution de l'équation Eikonal. (d) Définition de l'orientation des fibres pour chaque éléments	26
Figure 4.1	Smooth regions	32
Figure 4.2	Cylinder extraction	32
Figure 4.3	Local reference frame	33
Figure 4.4	Iterative algorithm for path generation	35
Figure 4.5	0° ply trajectories	36
Figure 4.6	45° ply trajectories	36
Figure 4.7	90° ply trajectory	36
Figure 4.8	Optimization algorithm	37
Figure 4.9	Minimum curvature path ending in a straight line	38
Figure 4.10	Minimum curvature path ending in a helix	38
Figure 4.11	Minimum curvature selected paths	40
Figure 4.12	Case 1 : internal pressure of 0.1MPa	40
Figure 4.13	Case 2 : traction stresses of 100N	40
Figure 4.14	Workcell definition.	43
Figure 4.15	Definition of Δt_m and Δt_s	45
Figure 5.1	Définition D'une trajectoire grâce à la direction de courbure minimale de la surface	48
Figure 5.2	Direction de courbure minimale	49
Figure 5.3	Trajectoires ayant un angle constant par rapport à l'axe équivalent du moule (gauche : 0°, droite : 45°)	50
Figure 5.4	Inversion des directions de courbure principales	50
Figure 5.5	Définition de deux poses sur une frontière	51
Figure 5.6	Recherche des nouveaux points de départ pour le calcul de l'hélice. Les lignes bleues représentent le processus de recherche des nouveaux points de départ situés sur la frontière entre la jonction et la branche couverte.	53
Figure 5.7	Méthode hélice discontinue : couverture des trois branches	54
Figure 5.8	Forme de la fonction $f(t)$	55
Figure 5.9	Hélice ajustable, variation de l'amplitude	56
Figure 5.10	Hélice ajustable, $P = 2$	56
Figure 5.11	Hélice ajustable, variation de la phase (gauche $\phi = 0$, droite $\phi = 1.7$)	56

Figure 5.12	Courbure géodésique de la trajectoire	57
Figure 5.13	Évolution de la courbure géodésique le long de la trajectoire illustrée à la figure 5.12	57
Figure 5.14	Points situés sur les intersections mathématiques des cylindres	60
Figure 5.15	Trajectoires dans la région jonction	61
(a)	Trajectoire sur l'intersection	61
(b)	Trajectoire sur l'intersection : algorithme étendu	61
Figure 5.16	Décalage d'un point de la courbe de référence	63
Figure 5.17	Extrapolation des extrémités de la courbe pour atteindre les frontières de la région de travail	63
Figure 5.18	Calcul de la courbure géodésique en un point d'une trajectoire sur une pièce en « Y »	64
Figure 5.19	Outil utilisé pour tester l'algorithme de détection de collisions	66
Figure 5.20	Méthode d'inclinaison de l'outil	66
Figure 5.21	Processus itératif d'inclinaison de l'outil	67
(a)	Pose initiale	67
(b)	Itération 1, $\psi = 1.3^\circ$	67
(c)	Itération 2, $\psi = 2.7^\circ$	67
(d)	Itération 3, $\psi = 4.0^\circ$	67
(e)	Itération 4, $\psi = 5.3^\circ$	67
(f)	Itération 5, $\psi = 6.7^\circ$	67
Figure 5.22	Évolution des valeurs articulaires le long des 15 trajectoires permettant la définition d'une couche orientée à 45°	70
Figure 5.23	Singularité du poignet du manipulateur	70
Figure 5.24	Cellule de travail comprenant le manipulateur FANUC M20iA (Université Concordia)	71
Figure 6.1	Directions principales de courbure non adaptées pour l'algorithme itératif	73
Figure 6.2	Algorithme <i>CPA</i> , manque de fibre aux alentours de la région jonction pour la trajectoire à 90°	75
(a)	Largeur de bande de 6.35 mm	75
(b)	Largeur de bande de 15.875 mm	75
Figure 6.3	Méthode « <i>Continuous tow shearing</i> » tiré de Kim et al. (2012)	76
Figure 6.4	Trajectoires effectuant des boucles sur elles-mêmes	77
(a)	Retour sur la branche de départ	77
(b)	Parcours d'une seconde branche	77
Figure 6.5	Outil utilisé pour les simulations	79

Figure 6.6	Outil traditionnel d' <i>AFP</i> (Rainer (2015))	79
Figure 6.7	Utilisation de deux manipulateurs dans la cellule de travail	80
Figure A.1	Organigramme général du programme	91
Figure A.2	Vue Générale du Programme de génération de trajectoires	93
Figure A.3	Identification des commandes	97
Figure A.4	Segmentation avec l'option Segmentation By Edge Length	101
Figure A.5	Segmentation sans l'option Segmentation By Edge Length	101
Figure A.6	Extraction des cylindres avec un seuil à 0% (à partir d'une segmentation sans l'option Segmentation By Edge Length)	101
Figure A.7	Extraction des cylindres avec un seuil à 100% (à partir d'une segmentation sans l'option Segmentation By Edge Length)	101
Figure A.8	Tableau récapitulatif des trajectoires voulues et obtenues	106
Figure B.1	Tableau des modes d'assemblage du manipulateur	113
Figure B.2	Fenêtre RoboDK	114
Figure B.3	Éditeur de programme RoboDK	115
Figure B.4	Menus déroulant de l'arbre des spécifications	116
Figure B.5	Volet d'options du repère Rb0	117
Figure C.1	Structure d'un programme .LS	120
Figure D.1	Options pour le contrôleur du M20iA	124
Figure D.2	Vue générale de la cellule de travail	124
Figure D.3	Cell Browser	126
(a)	Développement des sous-répertoires	126
(b)	Ajout du fichier .LS	126
Figure D.4	Editeur de Programme de ROBOGUIDE	126
Figure D.5	TeachPendant	128
Figure D.6	Menu du TeachPendant	130
Figure D.7	Succès du chargement du programme TP dans le TeachPendant	131
Figure D.8	Changement du dossier de sauvegarde par défaut	134

LISTE DES SIGLES ET ABRÉVIATIONS

AFP	Placement de Fibre Automatisé (<i>Automated Fibre Placement</i>)
ATL	Placement de Bande Automatisé (<i>Automated Tape Laying</i>)
FW	Enroulement Filamentaire (<i>Filament Winding</i>)
DDL	Degré de libertés
CAO	Conception Assistée par Ordinateur
CNC	Commande Numérique par Calculateur (<i>Computer Numerical Control</i>)

LISTE DES ANNEXES

Annexe A	LOGICIEL DE PLANIFICATION DE TRAJECTOIRES - MANUEL UTILISATEUR	90
Annexe B	CALCUL DES TRAJECTOIRES DANS L'ESPACE ARTICULAIRE DES ROBOTS	112
Annexe C	GÉNÉRATION ET STRUCTURE D'UN PROGRAMME .LS	118
Annexe D	UTILISATION DU LOGICIEL ROBOGUIDE	122

CHAPITRE 1 INTRODUCTION

Cette recherche s'inscrit dans le cadre d'un projet interuniversitaire proposé par le professeur Wen-Fang Xie de l'Université Concordia à Montréal visant à améliorer les techniques de production automatisées de structures en matériaux composites. Il est conduit en collaboration avec l'École de Technologie Supérieure et l'École Polytechnique de Montréal. Son but est de mettre en place une cellule robotique de Placement de Fibre Automatisé (*Automated Fibre Placement, AFP*). L'idée étant d'utiliser plusieurs robots afin de pouvoir fabriquer des structures de géométries complexes en matériaux composites. Ce mémoire traitera en particulier de la planification et de la génération de trajectoires de dépose de pré-imprégnés sur la surface des moules. Dans le but de clarifier le cadre de cette recherche, le paragraphe suivant rappellera des généralités sur les matériaux composites. Nous détaillerons ensuite les objectifs et hypothèses réalisés pour ce travail de recherche. De plus amples précisions sur les procédés de mise en forme des matériaux composites, et plus particulièrement sur l'*AFP* seront apportées au chapitre 2.

1.1 Composites à renfort fibreux

Les composites à renfort fibreux, du fait de leurs propriétés mécaniques très intéressantes, s'imposent de plus en plus comme des matériaux de construction privilégiés pour les industries cherchant à réaliser des structures à hautes performances. En effet, leur faible masse associée à leur grande raideur et résistance mécanique en font des matériaux prisés lorsqu'il s'agit de construire des structures à la fois légères et résistantes. C'est pourquoi on retrouve leur utilisation principalement dans les secteurs de l'aéronautique (fuselage, ailes, queue...) et de l'automobile (châssis de voitures de sport) qui doivent, pour rester compétitifs, réduire la consommation énergétique de leurs appareils. Comme le souligne Blom (2010), dans le cas de l'aéronautique la réduction de consommation de carburant est directement liée à la réduction de masse de la structure. Les économies réalisées en diminuant la masse d'un appareil d'un kilogramme ont été chiffré par Jones (1998) comme pouvant aller de \$55/kg dans le cas d'un petit avion civil, à \$33 000/kg pour une navette spatiale. D'autres secteurs comme ceux de l'énergie éolienne et de la conception de matériel de sport (voir figure 1.1) utilisent également des structures en matériaux composites pour obtenir les meilleures performances possibles pour leurs produits (Grant (2005)).

Les matériaux composites à renfort fibreux sont formés d'un mélange hétérogène de plusieurs constituants non miscibles possédant des propriétés mécaniques et chimiques différentes. Le



Figure 1.1 Fabriqué principalement avec des composites à fibres de carbone, le vélo *Argon 18 E119 TRI+* possède un cadre de seulement 1,2 kg. (tiré de Argon18)

but de ce mélange est de produire un matériau dont les propriétés finales sont meilleures que les propriétés de ses constituants pris individuellement. L'objectif de la création des matériaux composites est de surpasser les caractéristiques mécaniques des alliages métalliques qui sont depuis longtemps utilisés dans tous les domaines de l'industrie de fabrication mécanique. Les composants du mélange d'un matériau composite sont : une matrice, le plus souvent en polymère thermodurcissable ou thermoformable dans laquelle sont intégrées des fibres d'un autre matériau (verre, carbone etc.). La matrice permet de lier les fibres entre elles et de leur transférer les contraintes mécaniques. Ce sont les renforts fibreux, qui vont donner à la structure sa rigidité et sa résistance mécanique. Cette composition a la particularité de pouvoir conférer des propriétés mécaniques directionnelles à ces matériaux. En effet, la raideur et la résistance à la rupture d'un composite sera bien plus importante dans la direction de ses fibres que dans les directions perpendiculaires. Cette anisotropie permet alors au concepteur d'optimiser les propriétés mécaniques des pièces qu'il dessine en fonction de ses besoins.

Il existe plusieurs manières de produire et de mettre en forme les matériaux composites. Nous citerons ici succinctement quelques exemples, les procédés nous intéressant plus particulièrement dans ce mémoire étant présentés plus en détail au chapitre 2. Ces matériaux peuvent se présenter sous forme de tissus, dont les fibres sont tissées en deux ou trois dimensions, mis en forme par exemple par moulage par injection de résine liquide (Hamila (2007)). On peut

également citer les composites à particules ou à fibres courtes. Dans ces derniers les fibres sont coupées en petits morceaux et distribuées aléatoirement dans la matrice. Ils sont également généralement mis en forme par moulage. Une autre forme, qui nous intéresse plus particulièrement dans ce mémoire, est celle des pré-imprégnés. Ceux-ci peuvent être des bandes (figure 1.2) ou de fins rubans (figure 1.3) de fibres unidirectionnelles ou tissées dans plusieurs directions qui sont enduites de résine avant d'être stockées. Les proportions de fibres et de résine constituant le pré-imprégné sont contrôlées précisément et la polymérisation de cette dernière est interrompue avant solidification totale du matériau grâce à un stockage en environnement réfrigéré. Il existe différentes méthodes de mise en forme des pré-imprégnés, qui présentent toutes deux principales étapes : la dépose du matériau sur un moule puis la solidification de la matrice par chauffage et mise sous pression. Cette technique présente l'avantage d'offrir la possibilité de fabriquer des pièces dont les géométries seraient difficilement réalisable par moulage. En général ces matériaux sont utilisés pour fabriquer des composites stratifiés. C'est-à-dire, étant formé d'un empilement de couches dont la séquence d'empilement, qui est caractérisée par l'orientation des fibres dans les couches, peut être choisie de manière à optimiser les performances mécaniques de la pièce.

Si la dépose manuelle est encore couramment utilisée pour placer le pré-imprégné sur la surface du moule, l'augmentation des volumes de production de structures en composites et la forte demande des industries aéronautiques dans ce domaine a motivé dès les années 1960 le développement de procédés de fabrication automatisés. Ceux-ci ayant pour but d'augmenter la productivité ainsi que la régularité des produits par rapport à ceux réalisés à la main. Parmi eux on peut citer l'enroulement filamentaire (*Filament Winding, FW*), le placement de bandes automatisé (*Automated Tape Laying, ATL*) et le placement de fibre automatisé (*Automated Fibre Placement, AFP*). Ces trois procédés seront décrits au chapitre 2.



Figure 1.2 Bandes de pré-imprégné (ZOLTEK ®)



Figure 1.3 Rubans de pré-imprégné (HEXCEL®)

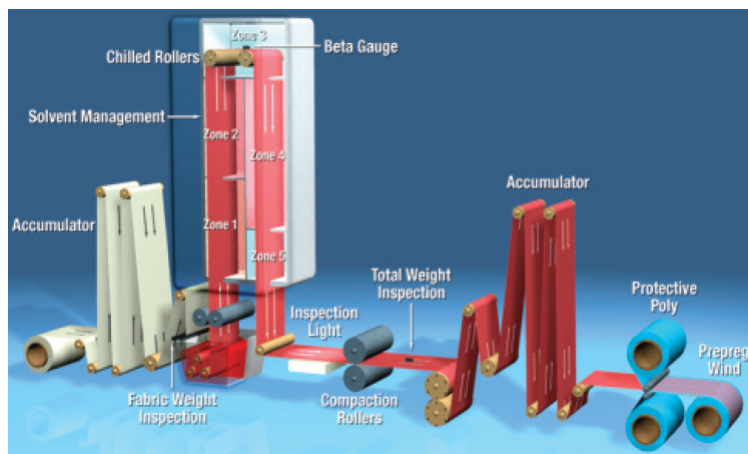


Figure 1.4 Procédé de production de pré-imprégné (Tiré de Hexply® Prepreg Technology)

1.2 Problématique

L'*AFP* est souvent cité comme étant une technique permettant la fabrication de pièces complexes en trois dimensions. En effet la grande variété de mouvements qu'offrent les robots utilisés pour déplacer les têtes de placement permet d'envisager la fabrication de produits de toutes sortes de géométries. Cependant, la plupart des études menées à ce jour en termes de génération de trajectoires se concentrent sur des pièces de surfaces ouvertes présentant une courbure légère, ou sur des pièces de révolution simples telles que des cylindres ou des cônes. Certaines études ont montré qu'il était possible de produire des trajectoires optimisées en fonction de certains cas de chargement, tout en minimisant les défauts tels que les manques (*gaps*) ou recouvrements (*overlaps*). D'autres définissent des méthodes pour obtenir des composites stratifiés à raideur variable en faisant varier la courbure des trajectoires le long de la surface des pièces. Toutefois, on constate qu'au niveau industriel, l'*AFP* n'est toujours utilisé principalement que pour fabriquer des pièces de grandes dimensions, que l'on peut souvent considérer comme localement planes, comme par exemple les fuselages et revêtements d'ailes d'avions ou les pales d'éoliennes (McCarville (2009)). Néanmoins, certaines industries, venant non seulement de l'aéronautique, mais également de la production d'équipements de sport sont intéressées par la production par *AFP* de pièces de dimensions plus faibles et de géométrie encore plus complexes comme des tubes en « T » ou en « Y » ou des structures formant des boucles fermées telles que les cadres de vélo. Les outils actuels ne permettent pas de fabriquer de tels produits. En effet, la taille imposante des machines utilisées et, en particulier des têtes de placement, limite les dimensions minimales et la complexité des formes fabricables. Pour répondre à cette nouvelle demande, des améliorations doivent donc être apportées au procédé de placement de fibre automatisé. En particulier, il est nécessaire de mettre au point des systèmes robotiques ayant une plus grande dextérité, afin de leur permettre de pouvoir travailler avec des pièces plus petites et complexes. Il est également nécessaire de mettre au point des nouvelles méthodes de planification de trajectoires spécifiquement adaptées à ce type de pièces.

1.3 Objectifs de recherche

L'objectif final de ce projet est de mettre en place une cellule de travail regroupant deux robots travaillant simultanément sur une même pièce : un bras manipulateur à 6 degrés de liberté (DDL), portant la tête de placement et un hexapode, muni à son extrémité d'une table rotative qui en fait un robot à 7 DDL, sur lequel sera fixé le moule. Le but à long terme étant de pouvoir construire une cellule d'*AFP* capable de fabriquer des structures en composite de

formes complexes et ainsi fournir des services aux industries aérospatiale et d'équipement de sport. Pour cela les objectifs suivant ont été fixés :

1. Étudier l'architecture, concevoir et réaliser la cellule de travail.
2. Développer la planification de trajectoire sur la surface du moule tout en évitant les collisions.
3. Développer un contrôleur dynamique pour coordonner les mouvements des robots travaillant en collaboration.
4. Réaliser les simulations du processus de fabrication collaboratif
5. Développer le matériel nécessaire à la mise en place de la cellule de travail et réaliser des expériences en temps réel.

La recherche présentée dans ce mémoire s'intéresse plus particulièrement au second point, c'est-à-dire la génération de trajectoires. Le but est d'investiguer de nouvelles formes de trajectoires permettant la fabrication de pièces de révolution, multi-axiales comme des tubes de section circulaire en forme de « Y ». Pour mener à bien ce travail, les objectifs à atteindre sont suivants :

1. Développer des algorithmes de planification de trajectoire permettant de couvrir des surfaces de révolution multi-axiales le plus uniformément possible.
2. Comparer les performances mécaniques des différentes formes de trajectoires générées.
3. Vérifier l'absence de collisions entre l'outil et le moule sur lequel sont déposées les bandes de pré-imprégné.
4. Développer un algorithme d'aide à la sélection de ces trajectoires en fonction des chargements que va subir la pièce.
5. Séparer la trajectoire générée dans l'espace cartésien en deux trajectoires exprimées dans l'espace articulaire des deux robots.

Pour atteindre ces objectifs, plusieurs hypothèses ont été fixées. Il est tout d'abord envisagé que l'ajout de DDL va entraîner l'augmentation de la dextérité du système de placement de fibre et ainsi permettre de mimer deux mains humaines plaçant des bandes de pré-imprégné sur une structure complexe. Pour cela plusieurs possibilités étaient envisageables. Il aurait

par exemple été possible d'utiliser deux manipulateurs, le premier déplaçant la tête de placement et le second tenant le moule sur lequel doit être placée la fibre. Cependant à cause de leur structure sérielle, les manipulateurs ne semblent pas être les meilleurs candidats pour un système de ce type. En effet, ceux-ci pourront avoir tendance à se déformer sous la charge due aux masses des têtes de placement et du moule ainsi que de la force de pression (jusqu'à 3.6 MPa (Dirk et al. (2012))) qui doit être appliquée lors du processus de dépose. Ce comportement risque alors de diminuer la précision du procédé. L'utilisation d'un robot parallèle du type plateforme de Stewart a donc été choisie, la rigidité due à sa structure parallèle pouvant en effet apporter une solution à ce problème de précision.

Pour réaliser la planification de trajectoire, il est nécessaire de connaître la forme du moule sur lequel on souhaite effectuer la dépose. Pour ce travail, il a donc été décidé que cette géométrie serait représentée par un maillage triangulé provenant d'un fichier au format .STL. Celui-ci a été choisi car la plupart des logiciels de conception assistée par ordinateur (CAO) permettent l'export sous ce format. Certaines hypothèses ont également été réalisées sur les pièces à traiter. Puisque l'on s'intéresse principalement aux pièces de types tubes en « Y », il a été choisi de se restreindre au cas où leurs différentes branches sont des cylindres de section circulaire, de même rayon et dont les axes ont tous un point d'intersection commun. Pour finir considérant la taille des têtes de placement existantes dans l'industrie qui sont souvent de grandes dimensions, l'hypothèse a été faite qu'il était possible de concevoir une tête de placement suffisamment petite et compacte pour être adaptée à nos besoins.

CHAPITRE 2 REVUE DE LITTÉRATURE

Comme évoqué en introduction, les matériaux composites sont de nos jours largement utilisés par les industries cherchant à produire des structures combinant des propriétés de légèreté à de hautes performances mécaniques. Afin d'augmenter leur productivité et leur compétitivité, ces industries ont rapidement cherché à développer des processus automatisés permettant de remplacer la fabrication à la main de ces structures en matériaux composites. Les deux technologies (enroulement filamentaire et placement de bande automatisé) ayant conduit à l'invention du placement de fibre automatisé (*AFP*) seront introduites dans ce chapitre. Nous présenterons ensuite plus en détails le procédé d'*AFP*, étudié dans ce travail. Dans un second temps, une synthèse des différents travaux de planification de trajectoire adaptés à l'*AFP* sera effectuée, permettant de définir les bases du travail présenté au travers de ce mémoire. Nous aborderons également dans cette partie les concepts de base de programmation en robotique.

2.1 Enroulement filamentaire (*FW*)

L'enroulement filamentaire est la première technique de fabrication automatisée permettant de produire des stratifiés en matériaux composites (Shen (1995)). Elle est principalement utilisée pour la production de pièces de révolution comme des tuyaux, des réservoirs de fluide sous pression, des enveloppes de moteur de fusée ou même des clubs de golf ou des cannes à pêche. Dans ce procédé, des rubans de fibres (verre, carbone) imprégnées de résine, sont enroulés au moyen d'une machine commandée numériquement, autour d'un moule possédant la forme de la pièce désirée, avec une tension et une forme d'enroulement prédéfinie. Selon les applications le moule peut être entièrement détruit ou retiré de la pièce finale voir laissé à l'intérieur de la structure en composite. En effet dans le cas de la production de réservoirs sous pression, une doublure intérieure est en général nécessaire pour garantir l'étanchéité du réservoir et celle-ci, si elle est assez rigide pour supporter la tension d'enroulement, peut être utilisée directement comme moule.

De nos jours, différents types de machines d'enroulement filamentaire existent, allant de la plus simple ne possédant que 2 DDL (rotation du moule et déplacement linéaire de la tête de distribution des fibres) permettant la fabrication de cylindres, à des machines plus sophistiquées pouvant posséder jusqu'à 6 DDL¹. Ces dernières permettent ainsi par exemple la production de tubes en coude et autorisent de plus grandes possibilités de conception dans

1. Un exemple de machine d'enroulement filamentaire à 6 DDL peut être trouvé à : https://www.youtube.com/watch?v=R_y3KgOdPxs

les formes d'enroulement. Certaines sociétés (Cadfil®) proposent par exemple la possibilité de faire varier la largeur de bande en tordant la bande de fibre sur elle-même.

La forme de l'enroulement est un paramètre important lors du processus de fabrication. En effet les caractéristiques mécaniques du produit fini vont fortement dépendre du motif réalisé par les fibres. Trois formes particulières d'enroulement sont en général utilisées. L'enroulement polaire, correspondant à des fibres ayant une orientation de 0° par rapport à l'axe de rotation du moule (figure 2.1a), l'enroulement hélicoïdal pour des angles allant de 5° à 80° (figure 2.1b) et l'enroulement en arceau qui permet de déposer les fibres à un angle proche de 90° (figure 2.1c). Le motif formé par ces enroulements peut alors être optimisé afin de résister à certains types de chargements.

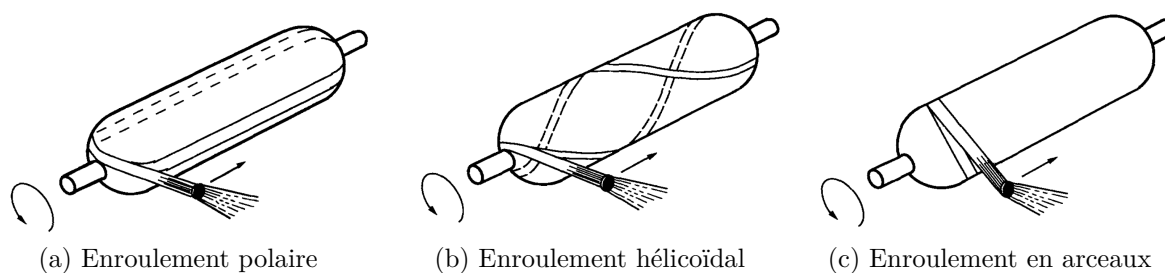


Figure 2.1 Différents motifs d'enroulements utilisés en enroulement filamentaire (Shen (1995))

Cette technique est particulièrement adaptée à la production de pièces de révolution. Elle possède de plus l'avantage de réaliser des pièces avec des fibres continues ce qui permet d'obtenir de très bonnes performances mécaniques. Cependant elle souffre de plusieurs inconvénients qui la rendent inappropriée pour la production de structures de forme plus complexe. Un des désavantages majeurs est l'impossibilité de produire des pièces présentant des concavités. En effet, les rubans étant enroulés avec une certaine tension, ils ne pourront pas adhérer sur des surfaces concaves. Cette tension peut également causer le glissement des fibres déposées sur le moule, qui risque de conduire à l'apparition de défauts tels que des régions vides de fibres (*gaps*), des micro-flambements de fibres ou encore une variation d'épaisseur (Hahn et al. (1994)). Un autre problème important de cette méthode peut être la difficulté, voir l'impossibilité de retirer le moule une fois l'enroulement terminé. Les techniques de production présentées dans les paragraphes suivants ont été mises au point dans l'objectif de s'affranchir des défauts présentés par l'enroulement filamentaire.

2.2 Placement de bande automatisé (*ATL*)

Le placement de bande automatisé (*Automated Tape Laying, ATL*) est un processus de fabrication de stratifiés en matériaux composites. Il a été développé à partir de la fin des années 1960 dans le but d'améliorer la productivité et la fiabilité des pièces par rapport à la dépose à la main. Ce procédé consiste à déposer, de larges bandes (généralement de 70 mm à 300 mm de large) de pré-imprégné sur un moule, afin de recouvrir sa surface d'une ou plusieurs couches. Trois grands types de machines existent, permettant la dépose de différents matériaux : les pré-imprégnés thermodurcissables, les pré-imprégnés thermoplastiques ou les fibres sèches (Blom (2010)).

L'organe principal d'une machine d'*ATL* est sa tête de placement. Celle-ci permet, d'acheminer le pré-imprégné de sa bobine de stockage, jusqu'à la surface du moule grâce à différents rouleaux de guidage. Dans celle-ci, le pré-imprégné y est également chauffé afin d'augmenter son adhésivité, nécessaire à sa tenue en place sur le moule. Il est alors guidé jusqu'au rouleau de compaction permettant de le plaquer sur le moule avec une force contrôlée.

Les machines d'*ATL* ayant été mises au point pour fabriquer des pièces de grandes dimensions en déposant de larges bandes de pré-imprégné, les dimensions et la masse des têtes de placement utilisées dans l'industrie sont souvent importantes. Les limitations en vitesse et en précision des bras robotisés au milieu des années 1980 (Dirk et al. (2012)) ont conduit à préférer l'utilisation de grands portiques robotiques pour déplacer les massives têtes de placement le long des trajectoires de dépose (figure 2.2). Ces portiques, généralement programmés hors ligne à la manière d'une machine à commande numérique classique (CNC), sont dans la plupart des cas conçus pour fabriquer des pièces de surface plane. De nos jours certaines machines d'une plus grande complexité permettent la dépose sur des surfaces courbées, mais leur prix à l'achat est très élevé (plusieurs millions de dollars). Cet inconvénient a poussé certains constructeurs comme *Automated Dynamics* à développer des machines moins chères, spécialisées dans la fabrication d'un seul type de pièces (Black (2003), Sloan (2008)).

De nos jours, les machines d'*ATL* sont principalement utilisées par les industries aéronautique et éolienne pour la production des fuselages et des ailes d'aéronefs ainsi que des pales d'éoliennes. Les principaux avantages de cette technique sont sa capacité à stopper et redémarrer la dépose aux endroits voulus grâce à des couteaux permettant de découper le pré-imprégné, et son important taux de dépose, qui la place parmi les plus productives pour la dépose de pré-imprégné. Ces avantages sont cependant contrebalancés par les forts investissements nécessaires à l'achat d'une telle machine et par le fait que les pertes de matière dues aux coupes sont plus importantes et les géométries réalisables moins complexes qu'avec



Figure 2.2 Portique de placement automatisé de bande (MTorres®)

le procédé d'*AFP*.

2.3 Placement de fibre automatisé (*AFP*)

Afin de s'affranchir des limitations des deux procédés décrits précédemment, une nouvelle technique appelée placement de fibre automatisé a été développée dès la fin des années 1980. Une des idées principales de ce processus est d'ailleurs déjà présente dans le brevet de Goldsworthy (Goldsworthy et al. (1974)) en 1974. Celui-ci décrit une tête de placement pour *ATL* permettant de découper la bande de pré-imprégné en fins rubans de 3.2 mm et de les déposer chacun à une vitesse différente, ceci dans le but d'être capable d'effectuer la dépose sur des surfaces courbées. Le principe de séparer la bande de pré-imprégné en multiples rubans permet de combiner les avantages de l'enroulement filamentaire qui autorise une grande variété de motifs de dépose de fibre, et ceux de l'*ATL* à savoir la capacité de couper et redémarrer le processus et la possibilité d'appliquer une pression de compaction lors de la dépose.

Les premiers développements de la technologie d'*AFP* ont été menés majoritairement par les grands noms de l'aéronautique, comme la *NASA* et *Boeing*, afin de produire plus rapidement et à plus faible coût des pièces ou certaines parties des fuselages d'avions militaire (Blom (2010)). Si cette technologie est restée pendant une vingtaine d'années principalement utilisée dans le domaine de la défense ou de l'aviation d'affaire ces dernières années l'ont vue s'étendre à l'aviation civile avec le lancement du Boeing 787 Dreamliner (présenté figure 2.3) et de son concurrent européen, l'Airbus A350-XWB qui contiennent jusqu'à 50% en masse de matériaux composites (McCarville (2009); Marouene (2015)).

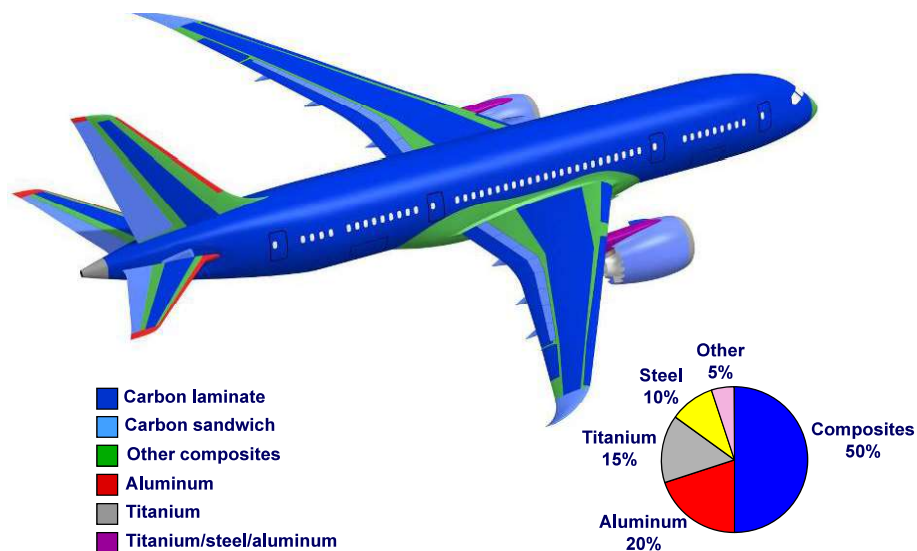


Figure 2.3 Composition du Boeing 787 Dreamliner (Roeseler et al. (2007))

De nos jours d'autres secteurs, comme l'industrie éolienne, s'intéressent également à cette technique de fabrication. Même si ce procédé était encore considéré comme peu rentable il y a quelques années car trop coûteux et pas assez rapide (Scott (2011)), des études montrent que les performances mécaniques obtenues en utilisant l'*AFP* pour fabriquer des pales d'éoliennes peuvent être significativement améliorées, surtout en terme de rigidité, par rapport aux techniques de fabrication traditionnelles utilisées dans ce milieu (Serrano et Vennam (2012)). Les sociétés spécialisées dans la conception de système d'*AFP*, comme *Coriolis*, cherchent également à transférer les technologies et le savoir-faire acquis par l'aéronautique à d'autres secteurs (principalement éolien et automobile) en développant des machines utilisant des matériaux de production à faible coût et qui permettent des cadences de productions mieux adaptées à ces secteurs.

Les développements menés depuis l'invention de cette technologie par les industries aéronautiques et les fabricants de machines-outils comme, *Cincinnati Milacron*, *Ingersoll* ou *MTorres* ont permis d'en augmenter la productivité et la fiabilité, tout en réduisant le prix d'achat des machines. Ce procédé s'impose donc peu à peu auprès des constructeurs aéronautiques alors qu'au début de cette technologie, le prix d'achat des machines était un facteur limitant (jusqu'à 6 millions de dollars d'après Dirk et al. (2012)).

2.3.1 Principe de fonctionnement

Comme décrit pour l'*ATL*, les cellules de placement de fibre automatisé se composent en général d'un robot sur lequel est montée la tête de placement (figure 2.4). Ce robot peut être, en fonction des pièces à fabriquer et des mouvements à effectuer, un manipulateur ou un portique automatisé. Ces cellules possèdent généralement 7 DDL (Blom (2010)), six permettant le positionnement et l'orientation de la tête de placement dans l'espace tandis que le dernier est utilisé pour faire tourner le moule sur lui-même. La redondance introduite par l'utilisation de 7 DDL offre la possibilité d'effectuer des tâches auxiliaires, comme de l'évitement d'obstacle, permettant ainsi la fabrication de pièces complexes. Cela peut également permettre d'optimiser les déplacements du robot afin par exemple d'augmenter la productivité. Les trajectoires sont en général programmées hors ligne (voir section 2.4.1) afin de pouvoir bénéficier des avantages de la simulation, comme la détection de collisions, la simulation dynamique, la planification de trajectoire, ou la re-programmation durant les phases de production (Olsen et Craig (1993); Marouene (2015)).

Comme dans le cas de l'*ATL*, l'organe principal de l'*AFP* est la tête de placement. La principale différence entre ces deux technologies provient de la tête d'*AFP* qui transporte plusieurs rubans de pré-imprégnés, stockés sur différentes bobines. Ces rubans ont des largeurs standards de 3.175 mm, 6.35 mm ou 12.7 mm. Une tête de placement, comme la Cincinnati VIPER®, peut déposer jusqu'à 32 rubans simultanément sur la surface du moule générant une largeur maximale de bande de 406 mm. Ces rubans peuvent être démarrés, coupés et arrêtés individuellement. Cela permet de réaliser des pièces proches de leur état final. En effet, il est par exemple possible de gérer les ouvertures présentes dans un fuselage d'avion en utilisant la capacité d'arrêt et redémarrage des rubans individuels, plutôt que de venir percer une

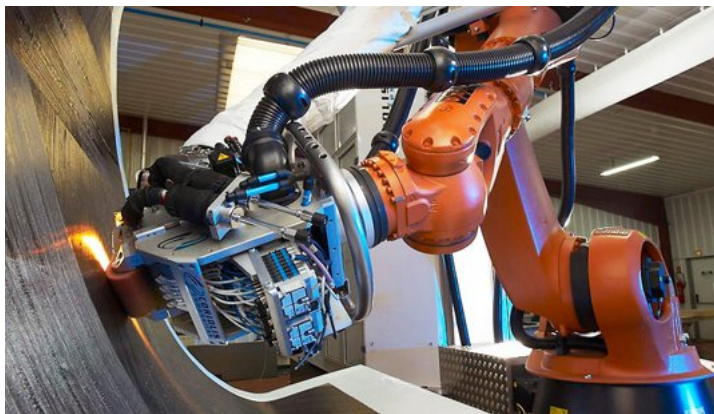


Figure 2.4 Cellule d'*AFP* (Composites)

pièce qui aurait été construite d'un seul bloc. Cela permet aussi, en faisant varier la largeur de la bande déposée, de limiter les défauts tels que les recouvrements et les manques de matière, apparaissant notamment lors de l'utilisation de trajectoires de dépose non parallèles (Blom (2010)). La tension des rubans et leur vitesse de distribution est également contrôlée de manière individuelle. Cela autorise la dépose à orientation constante des bandes de fibres sur des surfaces courbes et permet la réalisation de trajectoires courbes. Finalement comme pour l'ATL (section 2.2), les rubans de pré-imprégnés sont chauffés avant d'être pressés sur le moule par le rouleau de compaction (figure 2.5).

Pour finir, une fois la dépose terminée la pièce est mise sous vide et chauffée dans un autoclave (figure 2.6) afin de terminer le processus de polymérisation de la résine tout en chassant de cette dernière les potentielles bulles d'air qui auraient pu créer des faiblesses mécaniques. Cette étape est commune aux trois procédés présentés dans ce chapitre.

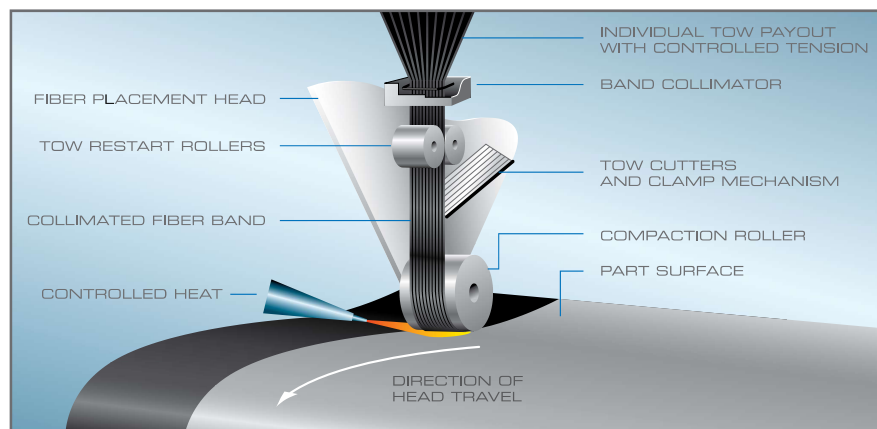


Figure 2.5 Principe de Fonctionnement d'une tête de placement (Brochure Cincinnati VIPER®, Fives)

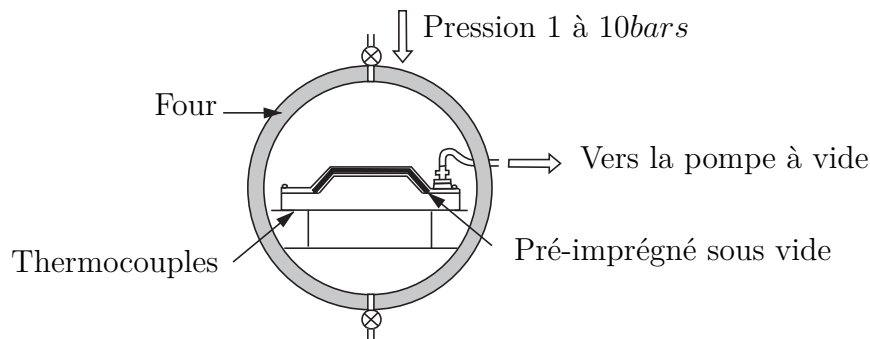


Figure 2.6 Polymérisation sous autoclave (d'après brochure HEXCEL®)

2.3.2 Limites du procédé

Si l'*AFP* possède de nombreux avantages pour la fabrication de structures complexes en trois dimensions, il existe tout de même quelques limites importantes à citer qui auront un impact sur la planification de trajectoires. Sans s'étendre sur celles traditionnellement présentées par les systèmes robotiques mécaniques comme les limites articulaires ou les singularités, on peut citer des contraintes de fabrication intrinsèquement liées au procédé de placement de fibre.

L'une des contraintes principales est le rayon de courbure minimum que peut supporter une trajectoire (Blom (2010)). En effet, lorsqu'une trajectoire est courbée dans le plan, les rubans situés à l'intérieur de la courbe sont en compression. Du fait de leur faible largeur par rapport à leur longueur ceux-ci sont très propices au flambement (Marouene (2015)). Dans sa thèse, Blom (2010) donne des valeurs typiques de rayons de courbures en dessous desquelles le flambement des fibres risque fortement d'apparaître (par ex. : 635 mm pour une bande de 102 mm formée de 32 rubans de 3.175 mm). Ces valeurs dépendent cependant des paramètres de fabrication utilisés et peuvent être améliorées. Le flambement conduit à l'apparition d'ondulations sur la surface de la couche (figure 2.7) pouvant entraîner la diminution des propriétés mécaniques du stratifié.

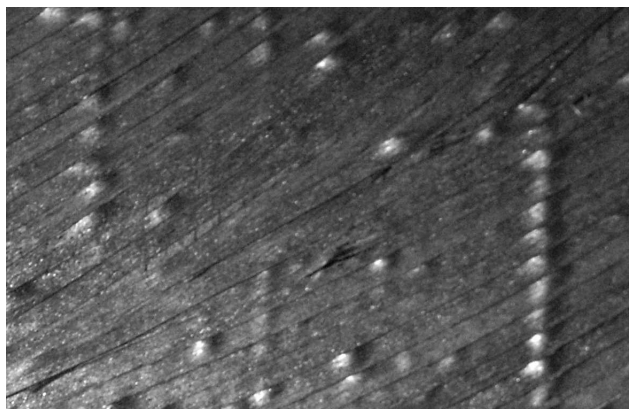


Figure 2.7 Flambement des fibres (Tirée de Blom (2010))

La seconde limite de ce procédé concerne la longueur minimale de coupe des rubans. En effet, en fonction de la conception de la tête de placement les lames permettant la coupe des rubans de composites sont situées à plus de 50 mm du rouleau de compaction (Dirk et al. (2012)). Il sera donc impossible de placer des bandes de pré-imprégnés d'une longueur inférieure sur le moule et si cela est nécessaire (figure 2.8), elles devront être découpées par la suite ou placées à la main à la fin du procédé automatisé ce qui diminue considérablement la productivité. La planification de trajectoire doit donc dans la mesure du possible éviter l'apparition de ces cas.

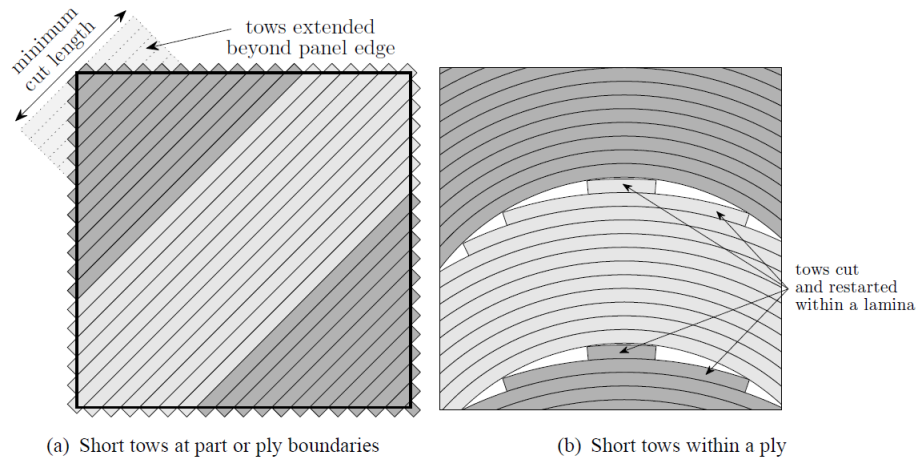


Figure 2.8 Défauts dus à la longueur de coupe minimale (Blom (2010))

2.3.3 Défauts de fabrication

Différents types de défauts de fabrication peuvent apparaître lors de l'utilisation de l'*AFP* pour produire des stratifiés en matériaux composites. Une description de ces défauts peut être trouvée dans la thèse de Marouene (Marouene (2015)). Dans ce mémoire nous ne nous intéresserons qu'à deux types de défauts, les manques (*gaps*) et recouvrements (*overlaps*) (figure 2.9). En effet, ceux-ci sont généralement causés par la méthode de génération de trajectoire utilisée (voir section 2.4.2).

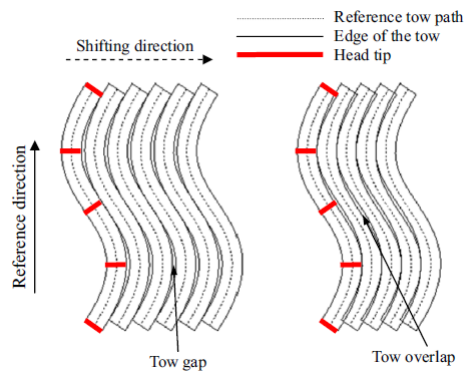


Figure 2.9 *Gaps* et *Overlaps* (D'après Kim et al. (2015))

Si ces défauts sont bien connus dans la littérature, leurs effets sur le comportement mécanique des stratifiés en matériaux composites demeurent encore à l'heure actuelle relativement incertains. Certaines études ont cependant été menées pour essayer de mieux caractériser l'influence de ces défauts sur les modes de rupture des pièces produites par *AFP*. Sawicki et

Minguet (1998) se sont intéressés à l'influence de la largeur des manques pour des pièces soumises à des contraintes de compression. Il ressort de leur étude que, d'une manière générale, peu importe cette largeur, une zone riche en résine entrainera une diminution de la résistance de la pièce soumise à des contraintes de compression. Cependant, si l'augmentation de ce taux de diminution est important pour des largeurs de manque inférieures à 0.76 mm, la diminution de résistance reste ensuite relativement constante pour des largeurs supérieures de défauts.

Dans une autre étude, Turoski (2000) a montré de manière numérique et expérimentale que les concentrations de contraintes autour d'une entaille pratiquée dans un stratifié quasi-isotrope ont une influence beaucoup plus importante sur le comportement mécanique d'une pièce soumise à de la traction ou de la compression que les défauts induits par l'*AFP*. On peut donc penser que l'influence des défauts intrinsèques au procédé de fabrication peut être négligée pour des pièces qui seraient soumises à des environnements rudes. En effet, les causes de rupture seraient alors plutôt dues à d'autres défauts structurels (comme des entailles ou des fissures) venant de facteurs extérieurs.

Pour finir, Croft et al. (2011) soulignent également le peu de connaissances que nous possédons sur le comportement des pièces produites par *AFP*. Ils proposent une étude expérimentale de différents types de défauts (manques, recouvrement, demi manque/recouvrement et *twisted tows*) pratiqués sur des échantillons soumis à différents types de chargements (traction, compression, cisaillement plan, traction et compression en présence d'un trou circulaire). Leur but étant de caractériser individuellement les effets de chacun de ces défauts pour en ressortir les paramètres critiques afin de mieux comprendre par la suite le comportement d'une pièce soumise à plusieurs défauts et à un chargement complexe. Le tableau présenté à la figure 2.10 synthétise les résultats de leur étude. On y remarque que l'influence de ces défauts sur les performances de la pièce est parfois négligeable et ne dépasse jamais les 13% de variation par rapport à une pièce sans défaut. Certains entraînent d'ailleurs pour un type de contrainte, une amélioration du comportement de la pièce. Cependant il est à noter que si le comportement est amélioré pour un type de contrainte, le même défaut entrainera une détérioration du comportement mécanique pour un autre chargement. Les auteurs soulignent donc que pour une pièce réelle soumise à un chargement multiaxial, il est toujours pour le moment difficile de proposer une quantification des effets des défauts sur son comportement.

En conclusion on peut voir que même s'il demeure difficile de quantifier l'impact des défauts, il ressort que dans la plupart des cas on observe une détérioration des caractéristiques mécaniques lors de la présence de manques de fibres dans les couches du stratifié. De plus certains auteurs (Sawicki et Minguet (1998)) ont montré que la rupture en compression était due à

		Gap	Overlap	Half Gap/Overlap	Twisted Tow
Tension		—	—	↘	↗
Compression		—	↗	—	—
In-Plane Shear	Length	—	—	↘	↘
	Width	↗	↘	↘	↘
OHT		—	—	—	
OHC	Length	↗	↗	↗	
	Width	↘	↘	↘	

↗ $\geq 3\%$ increase (up to 13%)
 — $\pm 3\%$ variation
 ↘ $\geq 3\%$ decrease (up to 12%)

Figure 2.10 Influence des défauts pour différents cas de chargement (Croft et al. (2011))

des ondulations hors plan présentes dans les couches du stratifié qui peuvent être induites aussi bien par les manques que par les recouvrements. Il est donc raisonnable de penser que lors du processus de génération de trajectoire, il est essentiel d'essayer d'éviter la production de ce type de défauts.

2.4 Principes de planification de trajectoire

2.4.1 Méthodes de programmation des robots industriels

De nos jours, deux grandes méthodes de programmation des robots industriels sont utilisées : la programmation « en ligne » ou « hors ligne ». Chacune de ces deux méthodes possède des avantages et inconvénients en fonction de la situation dans laquelle elles sont utilisées. La principale différence entre ces deux méthodes réside dans le fait que la programmation hors ligne s'effectue sur un ordinateur hôte, le plus souvent grâce à un logiciel de simulation permettant la visualisation 3D de la cellule de travail. Bien que la première méthode nécessite l'arrêt de la production, elle est tout de même toujours répandue dans l'industrie car elle permet d'obtenir rapidement une trajectoire adaptée à l'environnement réel du robot (Nilsson (1996)). En effet, un inconvénient de la programmation hors ligne est d'assurer la correspondance entre les cellules virtuelles et réelles, ce qui nécessite une étape préalable d'étalonnage absente en programmation en ligne. Toutefois de nombreux autres avantages font de la programmation hors ligne un outil puissant qui, avec l'essor des logiciels de simulations tels que ROBOGUIDE, MotoSim, RobotMaster ou RoboDK, est de plus en plus répandue. Outre le fait que cette technique de programmation ne nécessite pas l'arrêt de la production, on peut également souligner le fait qu'elle permet, à travers des simulations, d'optimiser le cycle de

travail du robot et même de concevoir des cellules de travail optimisées pour une certaine tâche, avant même l'investissement dans de l'équipement (Olivieri (2013)). Cette technique est également beaucoup plus adaptée à la génération de trajectoires complexes, notamment lorsque celles-ci sont décrites mathématiquement ou à partir de fichier CAO (Nilsson (1996)). Ce type de trajectoire étant en général extrêmement difficile à reproduire à l'aide du pupitre de commande en programmation en ligne. Pour finir la programmation hors ligne permet la simulation des mouvements du robot sans risque d'endommagement pour ce dernier. Cette simulation permet de vérifier le comportement des machines et de vérifier entre autres, qu'il n'y ait pas de collision avec l'environnement extérieur, ce qui est une étape importante avant le lancement d'une production.

Le but de ce travail de recherche étant de générer des trajectoires adaptées à l'*AFP* sur la surface de moules décrits à l'aide de fichiers CAO, il a naturellement été choisi de développer une interface utilisateur permettant la définition de ces trajectoires et la programmation hors ligne des robots inclus dans la cellule de travail.

2.4.2 Méthodes de génération de trajectoires

L'objectif de l'*AFP* étant de produire des structures stratifiées en matériaux composites, il est essentiel que le processus de planification de trajectoire s'attèle à générer des chemins permettant la couverture de la surface complète du moule. Dès lors que les processus de production ont été automatisés, il a été nécessaire de développer des méthodes de planification de trajectoires afin de permettre à l'outil porté par la machine de réaliser les tâches voulues.

Plusieurs méthodes de génération de chemins existent à présent et elles présentent chacune des avantages et des inconvénients en fonction de l'application que l'on souhaite en faire. Une des méthodes de génération de trajectoires utilisée dans l'industrie est la génération de chemins de couverture. Ceux-ci sont définis comme étant les chemins que doit suivre un outil pour couvrir la totalité d'une surface de travail (Olivieri (2013)). Ce type de chemin est utilisé pour différentes applications allant du polissage automatisé à l'usinage en passant par le dépôt de matière en spray (Tam (1999); Lasemi et al. (2010); Sheng et al. (2005)). Le grand nombre de chemins possibles pour couvrir toute une surface laisse la possibilité de rechercher un chemin optimal pour l'application souhaitée, par exemple en minimisant le temps de parcours (Current et Schilling (1989)) ou en cherchant à obtenir la meilleure qualité possible pour le produit final (Lasemi et al. (2010)). Cependant, ces problèmes d'optimisation peuvent se révéler très complexes et difficilement résolubles. Une technique de simplification utilisée est alors de choisir a priori un type de motif pour le chemin, et d'en chercher les paramètres qui conduisent à l'optimalité pour les critères choisis. Olivieri (2013) recense

quatre types de motifs qui peuvent être utilisés pour générer des chemins de couverture. On y retrouve les chemins en zig-zag, en spirale, les courbes de Lissajous (figure 2.11a) et les courbes de remplissages (comme les courbes de Péano, figure 2.11b, de Hilbert ou de Moore). Olivieri (2013) propose dans son travail, d'utiliser les chemins en zig-zag pour des applications d'automatisation de contrôle non destructif par courant de Foucault. En effet, ce type de chemin présente l'avantage de faire en sorte que chaque point de la surface est inspecté selon la même direction de balayage. Cela permet alors d'effectuer facilement deux inspections perpendiculaires, maximisant les chances de détecter d'éventuelles fissures dans la pièce. Sheng et al. (2005) ont montré que, pour des procédés de moulage par projection, les courbes en spirales conduisent à une meilleure uniformité du dépôt de matière par rapport à un chemin en zig-zag. Pour des applications de polissage sur des surfaces relativement planes, Tam (1999), a montré que l'utilisation de courbes de Lissajous présentait de bons résultats.

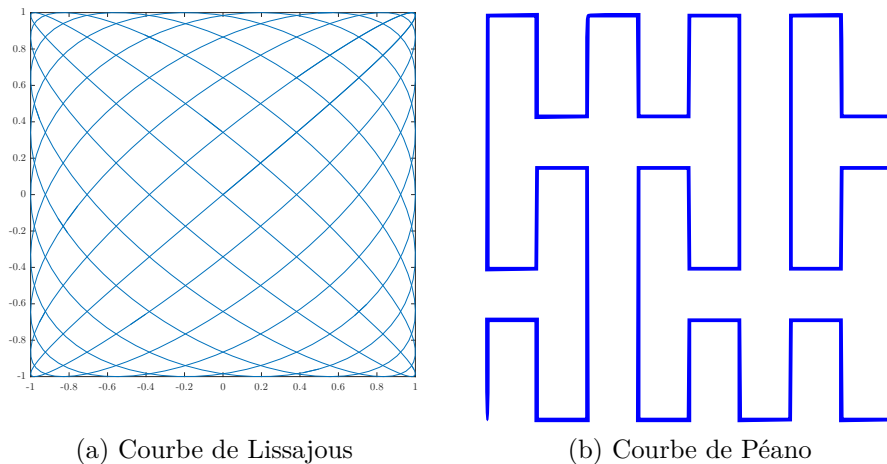


Figure 2.11 Différents motifs de chemins de couverture

Dans leur revue de l'état de l'art, Lasemi et al. (2010) rappellent les techniques traditionnelles utilisées pour la génération de trajectoires en usinage par machines à commandes numériques. La première technique, nommée trajectoire à direction parallèle, se rapproche de la technique de génération de chemins en zig-zag. Il y est proposé de générer des trajectoires parallèles à une trajectoire de référence (figure 2.12a). Le choix de cette trajectoire de référence va alors influencer la longueur de la trajectoire et par conséquent le temps de cycle. Une seconde technique appelée trajectoire parallèle au contour, propose d'utiliser le contour de la surface de travail comme référence et de calculer des trajectoires parallèles à ce contour pour couvrir toute la surface voulue (figure 2.12b). Il est souligné dans cet article que le choix entre ces deux méthodes (dans le but d'obtenir un temps de cycle le plus court possible) est fortement

relié à la géométrie de la pièce, il n'est donc pas possible de savoir a priori laquelle de ces deux méthodes sera la plus rapide. Les techniques de générations de trajectoires iso-paramétrique et iso-planaire y sont également décrite. La première nécessite de connaître la représentation paramétrique ($S(u, v)$) de la surface de travail. La génération de courbes iso-paramétriques consiste ensuite à calculer des courbes équidistantes dans le domaines paramétrique. Chaque courbe est calculée en fixant un des deux paramètres (par ex. u) et en faisant varier continuellement le second. Des incréments constants sur le premier paramètre permettent ensuite de générer des courbes équidistantes dans le domaine paramétrique. L'inconvénient de cette méthode est qu'elle ne permet pas en général de produire des courbes équidistantes sur la surface réelle de la pièce. Pour des applications d'usinage, cela se traduit alors par des hauteurs de crêtes différentes entre les passes de l'outil. La méthode iso-planaire consiste quant-à-elle à générer des courbes en calculant l'intersection entre des plans parallèles et la surface de la pièce. Dans le cas de l'usinage, la distance entre ces plans peut être ajustée en fonction des contraintes sur les hauteurs de crêtes entre les passes de l'outil. Cette technique présente également l'avantage de pouvoir être utilisée sur des pièces représentées par des maillages triangulés. Cela en fait alors une des méthodes largement utilisée dans les logiciels commerciaux de fabrication assistée par ordinateur.

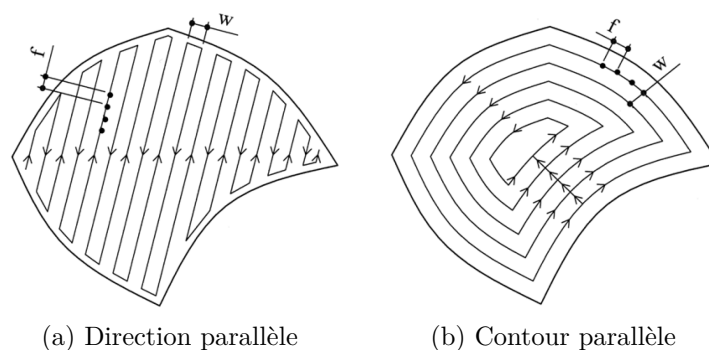


Figure 2.12 Motif de couverture traditionnels en usinage (Lasemi et al. (2010))

2.4.3 Trajectoires spécifiques à l'AFP

Contrairement à d'autres applications comme le dépôt de peinture, le polissage ou l'usinage, où différents types de chemins de couverture sont envisageables, la liberté de choix est plus restreinte dans le cas de l'AFP. En effet, la fibre étant continue, il n'est pas possible d'effectuer des changements brusques de direction (angles vifs) sans l'endommager. De plus, on cherche en général à faire en sorte que les différentes couches possèdent des orientations de fibres

choisies en fonction des contraintes que va subir la pièce.

Les premiers développements de techniques de génération de trajectoire pour l'*AFP* se sont concentrés sur la fabrication de larges panneaux plats (Shirinzadeh et al. (2004)) avant de s'orienter vers des objets de formes plus complexes en trois dimensions comme des surfaces courbées ouvertes ou des cônes (Shirinzadeh et al. (2004); Blom (2010)). La méthodologie globale utilisée dans ces travaux est de générer une courbe guide, possédant certaines propriétés en fonction de l'application, puis de la décaler afin de couvrir toute la surface du moule. Cela permet par exemple de définir des stratifiés dont chaque couche présente une orientation de fibre constante par rapport à une référence. Une séquence d'empilement de ces couches peut alors être définie afin de répartir au mieux les orientations des fibres en fonctions des contraintes que va subir la pièce ou de fabriquer des stratifiés quasi-isotropes (figure 2.13).

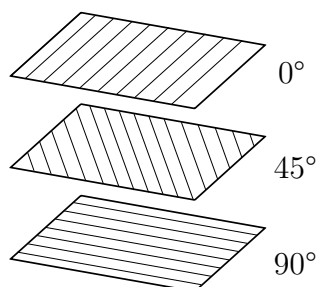


Figure 2.13 Séquence d'empilement possible de couches à orientation de fibre constante

Plusieurs techniques ont été mises au point pour générer la courbe guide ainsi que les trajectoires décalées, et seront décrites dans les prochains paragraphes. Une des principales différences observées pour la réalisation de ce calcul réside dans le choix de la représentation de la surface à couvrir. Si certains auteurs préfèrent utiliser une représentation paramétrique permettant de profiter d'un cadre mathématique pour travailler, d'autres ont choisi d'utiliser des représentations par maillage triangulé, qui vont nécessiter l'utilisation de techniques numériques. Ces deux représentations possèdent chacune leurs avantages en fonction de l'application que l'on souhaite en faire. Ceux-ci ont été listés par Chen et al. (2009) pour le cas de l'étude de l'application de peinture automatisée, mais certains restent tout à fait cohérents pour la génération de trajectoire pour *AFP*. Si la représentation par maillage permet d'avoir accès directement à l'aire des facettes et à leurs normales, utiles pour définir la pose de l'outil, la précision de la trajectoire générée dépendra du raffinement du maillage qui, rappelons-le, reste une approximation de la surface réelle du moule. Cette représentation souffre également d'un manque d'informations sur les propriétés géométriques de la surface, comme par exemple la localisation des arêtes vives. Celle-ci peut en revanche être connue dans une représentation paramétrique. Cette dernière sera également plus précise mais aura

le désavantage de ne pas permettre le traitement de surfaces complexes dans leur globalité, contrairement à une représentation discrète. En effet il n'est pas possible dans le cas général de trouver une surface paramétrée décrivant complètement un moule. Celui-ci devra donc en général être découpé en sous-régions de géométrie simple avant le calcul de génération de trajectoire.

Parmi les auteurs ayant préféré une représentation paramétrique de la surface du moule ($S(u, v)$), Shirinzadeh et al. (2007) proposent d'utiliser une méthode d'intersection d'un plan avec la surface du moule pour générer la courbe guide (figure 2.14). Cela permet de facilement définir l'orientation des fibres par rapport à un axe de référence en choisissant le plan initial. Pour être adapté à l'*AFP*, ce plan doit être choisi de manière à former, lors de son intersection avec la surface du moule, une courbe ouverte. Les auteurs proposent ensuite de résoudre itérativement l'équation d'intersection du plan et de la surface du moule. A partir d'un point initial, deux étapes sont nécessaires pour calculer le point suivant. Dans un premier temps, un pas intermédiaire est effectué dans une des deux directions paramétrique (u ou v), l'autre restant constante, de manière à décaler le point initial dans le sens de parcours de la courbe de référence que l'on recherche. Un pas de convergence est ensuite effectué durant lequel la variation à appliquer au second paramètre pour que le nouveau point se trouve à l'intersection des deux surfaces est recherché par la méthode de Newton-Raphson. Une fois la courbe de référence totalement définie, l'utilisation de la géométrie différentielle permet de calculer des trajectoires parallèles à la courbe initiale se trouvant toujours à la même distance géodésique de celle-ci (c'est-à-dire la distance calculée sur la surface). Celle-ci correspond à la largeur de la bande déposée et permet ainsi d'assurer une couverture uniforme sans *gaps* ni *overlaps*. Cette méthode a été améliorée par Yan et al. (2014) qui proposent d'utiliser une distance de décalage variable en fonction de la courbure de la surface afin d'assurer que tous les rubans de pré-imprégnés soient bien pressés contre la surface du moule. Pour chaque point de la courbe de référence, la distance de décalage autorisée est calculée. La plus petite distance trouvée est ensuite choisie et convertie en nombre de rubans à déposer, pour définir la distance de décalage servant au calcul de la nouvelle courbe. Il est également souligné dans cet article que si les courbes sont représentées de manière discrète, les trajectoires associées aux mouvements du rouleau de compaction seront alors des segments de droite. Cette approximation peut causer des interférences entre le rouleau et le moule dans le cas d'une surface convexe, ou un mauvais dépôt des bandes de fibres dans le cas d'une surface concave. Pour remédier à ce problème, il est proposé d'utiliser un critère définissant l'erreur entre la trajectoire formée de segments et celle qui serait réellement sur le moule. En fonction de ce critère, la discrétisation de la trajectoire peut être raffinée afin d'éviter les inconvénients sus-cités. Une approche totalement différente a été utilisée par Parnas et al. (2003) pour définir les directions de fibres dans un

stratifié. Ces auteurs proposent d'utiliser des surfaces et courbes de Bézier pour représenter respectivement le moule et les trajectoires. La position des points de contrôle des courbes sur la surface est optimisée de manière à minimiser la masse d'une plaque soumise à un chargement particulier.

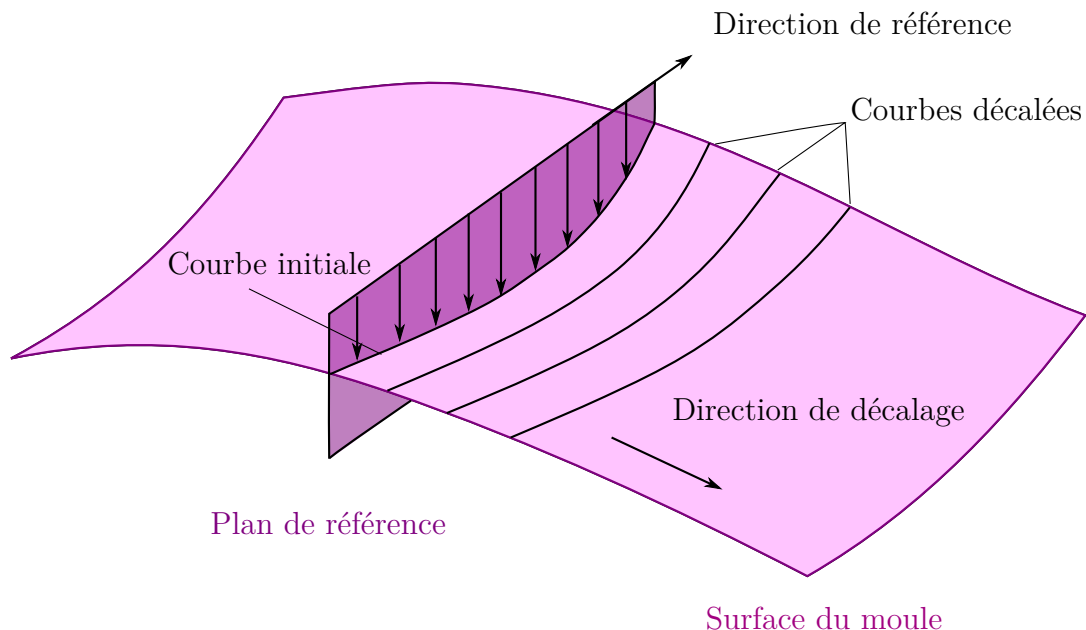


Figure 2.14 Génération de la courbe guide par intersection de la surface du moule avec un plan de référence

De nombreux travaux sont également actuellement menés sur les composites à raideur variable. Dans ceux-ci, les fibres ont des trajectoires courbes, permettant d'obtenir des propriétés mécaniques variables le long de la pièce et d'améliorer leur comportement structural (Blom (2010)). En effet, Gürdal et al. (2008) ont montré qu'il était possible, avec ce type de trajectoire, d'augmenter la charge critique de flambement et d'obtenir un découplage entre cette dernière et la raideur globale de la pièce. La courbe de référence choisie pour ce type de trajectoires est en général également représentée par des équations paramétriques. Celles-ci vont dépendre des propriétés géométriques de la surface du moule et des propriétés mécaniques que l'on souhaite obtenir. Les trajectoires suivantes, permettant de couvrir le restant de la surface du moule, sont calculées en décalant la courbe de référence dans la direction d'un axe fixe (Olmedo et Gurdal (1993)), cela permet de garder le contrôle sur la forme de la trajectoire mais entraîne la formation de manques et recouvrements puisque les trajectoires ne restent pas parallèles si la surface du moule n'est pas plane.

En ce qui concerne la génération de trajectoires sur des maillages triangulés, un important travail a été réalisé par Bruyneel et Zein (2013) dans lequel la méthode de *Fast Marching*

est adaptée afin de produire des trajectoires équidistantes sur des surfaces de géométries complexes (figure 2.15). Cette méthode, mise au point par Sethian (1996) consiste à résoudre l'équation Eikonal (equation aux dérivées partielles non linéaire) qui décrit le temps de trajet d'une interface mobile sur un domaine. Ici, en considérant une trajectoire de référence définie sur le domaine (la surface à couvrir), un ensemble de trajectoires équidistantes peut être calculé de la manière suivante : chaque point de la trajectoire possède une vitesse orthogonale à la courbe et celle-ci est la même pour tous les points. Il est alors possible de définir pour chaque point x de la surface le temps $T(x)$ nécessaire pour se déplacer de la courbe de référence à ce point x . On peut alors dire que n'importe quelle courbe parallèle à la courbe de référence est une iso-valeur de $T(x)$. Cela permet alors de définir un ensemble de courbes équidistantes, parallèles à la courbe de référence. Cette courbe de référence est ici générée dans un espace 2D à l'aide de fonctions paramétriques avant d'être transportée sur la surface réelle du moule. Cette méthode permet de traiter des pièces de géométries complexes telles que l'on peut en trouver dans l'industrie. La générations de trajectoires parallèles équidistantes évite l'apparition, dans la définition des couches, de manques ou de recouvrement, mais cette technique ne permet pas de faire varier la largeur des bandes déposées. Il n'est donc pas possible de s'adapter à la courbure de la surface (comme proposé par Yan et al. (2014)) pour éviter les brins mal pressés contre le moule et optimiser le taux de dépose. Cette technique permet finalement de définir facilement l'orientation des fibres dans chaque élément du maillage et ainsi d'exporter ces données pour réaliser une étude de comportement structurel par éléments finis. Ceci a été mis en œuvre par Lemaire et al. (2015) pour calculer des trajectoires parallèles optimisées afin de maximiser la raideur de structures soumises à des chargements. Ce travail a permis de mettre en évidence à travers des simulations numérique, qu'il était possible d'augmenter les performances mécaniques des structures construites par *AFP* en utilisant des trajectoires courbées.

Pour finir, une autre approche a été développée par Han et al. (2012) pour couvrir des surfaces fermées tout en respectant la contrainte de courbure maximum des rubans. La technique d'intersection d'un plan avec la surface du moule pour générer la courbe de référence n'étant pas adaptée aux surfaces fermées (plusieurs solutions peuvent apparaitre pour cette intersection), ces auteurs proposent d'utiliser un algorithme itératif pour définir les trajectoires. Cet algorithme sera décrit plus en détail au chapitre 4.4.1. Une fois la trajectoire de référence obtenue, les points formant cette courbe sont décalés perpendiculairement afin de former des trajectoires parallèles. Une vérification de la courbure des trajectoires est également effectuée afin d'assurer que celle-ci ne dépasse pas la limite autorisée en *AFP*.

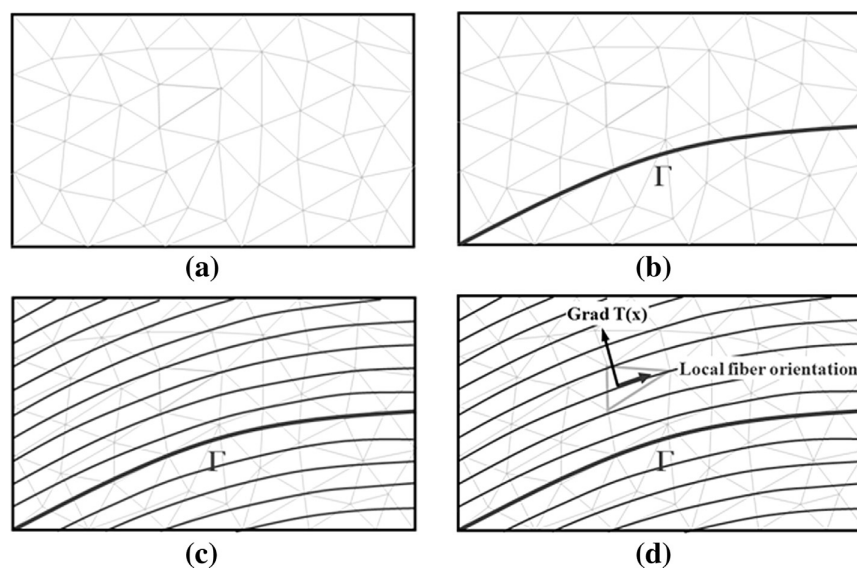


Figure 2.15 Utilisation de la méthode *Fast Marching* pour la définition de trajectoires équidistantes sur un maillage triangulé (Bruyneel et Zein (2013)). (a) Définition du maillage. (b) Définition de la courbe de référence. (c) Solution de l'équation Eikonal. (d) Définition de l'orientation des fibres pour chaque éléments

CHAPITRE 3 DÉMARCHE DE TRAVAIL

La mise en place d'une cellule de placement de fibre automatisée utilisant des robots collaboratifs, qui est l'objectif à long terme du projet dans lequel s'inscrit ce travail de recherche, nécessite de travailler sur plusieurs aspects. Celui qui nous intéresse dans ce mémoire est la génération de trajectoires, adaptées à l'*AFP*, permettant de couvrir un moule avec des bandes de pré-imprégnés à l'aide d'un système robotique. Pour mettre en place ce processus, plusieurs étapes ont été nécessaires. La démarche développée pour aboutir à un programme de planification de trajectoires fonctionnel va être décrite dans les paragraphes suivants.

En premier lieu, il a été choisi de travailler sur des moules représentés à partir de fichiers CAO au format stéréolithographie (.STL). Ceux-ci décrivent les objets sous forme de maillages triangulés (formés à partir d'une liste de points reliés par des arêtes et formant des faces triangulaires). Ce format d'enregistrement ne conservant qu'un minimum d'informations, il est alors nécessaire de re-calculer certaines propriétés géométriques des pièces à traiter. À cette fin, et afin de pouvoir plus tard définir les régions de travail sur lesquelles le placement de fibre doit être effectué, une première étape de segmentation sera implémentée. Celle-ci va permettre de découper la surface de la pièce en régions continues et lisses afin d'obtenir, pour chacune de ces régions, des propriétés géométriques également continues et lisses. Une fois ce premier traitement réalisé, l'étude de la génération de chemins en elle-même pourra être effectuée dans de bonnes conditions.

Dans un premier temps, les différentes méthodes permettant de générer des trajectoires sur des surfaces ont été étudiées et il a été choisi d'adapter la méthode itérative proposée par Han et al. (2012) qui semblait la mieux appropriée pour notre travail. Cette stratégie permet de générer sur des surfaces quelconques, des courbes présentant une direction constante par rapport à une référence. Une fois l'efficacité de cette méthode confirmée dans des cas simples (cylindres, cônes, combinaison de ces deux géométries, etc.), nos travaux se sont concentrés sur la génération de trajectoires pour *AFP* sur une forme particulière de moules. Il a été choisi dans ce travail, de se focaliser sur des surfaces de révolution multi-axiales, plus particulièrement des tubes, de section circulaire constante, en forme de « Y ». Deux démarches différentes seront présentées pour réaliser la couverture en pré-imprégné de ces pièces. Elles ont conduit au développement des quatre algorithmes, tous implémentés dans le logiciel de planification de trajectoires développé. Dans le cas des deux premières méthodes, présentées au chapitre 5, la stratégie est de couvrir chaque branche du « Y » individuellement avec des trajectoires dérivant d'hélices, puis de joindre ces différentes branches en calculant une

trajectoire de dépose de fibres sur l'intersection des cylindres formant les branches du moule. Une piste différente a ensuite été explorée, présentée au chapitre 4 en faisant l'hypothèse que les pièces produites en *AFP* auraient de plus grandes raideur et résistance mécanique, si chaque trajectoire, c'est-à-dire chaque bande de pré-imprégné déposée, parcourait deux branches du « Y » sans être coupée. Cela permet également d'éviter de devoir ajouter des trajectoires permettant de lier les branches entre elles. Suivant ces hypothèses, un premier travail a été réalisé généralisant les méthodes de placement de fibre à orientation constante qui sont utilisées dans l'industrie. Dans un second temps, notre travail s'est concentré sur l'objectif de générer des trajectoires dont la courbure géodésique sur la surface du moule n'excédait pas une certaine limite, au-delà de laquelle le processus d'*AFP* est connu pour entraîner l'apparition de défauts, détériorant les caractéristiques mécaniques des pièces produites.

Une analyse structurelle par méthode des éléments finis sera ensuite menée sur un modèle simplifié des pièces fabriquées avec les trajectoires produites dans ce travail. Celle-ci permettra de valider ou d'infirmer l'hypothèse faite que les pièces fabriquées avec des trajectoires sans coupe ont une plus grande rigidité mécanique. Elle permettra également d'identifier quelles sont les meilleures formes de trajectoires pour certains cas de chargement particuliers.

Pour finir, afin de démontrer la faisabilité des trajectoires générées, des simulations seront réalisées dans le cas d'une cellule de travail simple à l'aide du logiciel RoboDK. Cette cellule comportera un robot manipulateur FANUC couplé à une table rotative. Un algorithme permettant de séparer la trajectoire cartésienne en deux trajectoires dans les espaces articulaires des robots sera proposé et des tests seront conduits pour vérifier la faisabilité de telles trajectoires.

CHAPITRE 4 ARTICLE 1 : FEASIBILITY STUDY OF ROBOTIC FIBRE PLACEMENT ON INTERSECTING MULTI-AXIAL REVOLUTION SURFACES

Clément Hély, Lionel Birglen and Wenfang Xie

Cet article a été soumis pour publication au Journal *Robotics and Computer-Integrated Manufacturing* le 13 mai 2016.

4.1 Abstract

In this paper, the first steps towards using a robotic workcell for the automated fibre placement (AFP) manufacturing of Y-shaped tubes are proposed. The proposed workcell is constituted of a standard serial manipulator holding the fibre placement toolhead combined to a rotary table on which the part where the fibres must be laid out is attached. The investigations carried out in this work explore the feasibility of this setup and more precisely the path planning aspect. To this aim two novel path planning algorithms are presented generalizing the techniques proposed in the literature for open-contoured and cylindrical surfaces. In the first, the maximal geodesic curvature typically allowed in AFP is disregarded to generate continuous paths with a constant placement angle on the branches without any gaps or overlaps. Subsequently, a second algorithm, taking into account this curvature constraint, is presented. The different shapes of the generated toolhead trajectories are then analyzed using a finite element software to give some insights about their usefulness. Finally, an algorithm to optimize the motion of the robotic system is presented and simulations are discussed.

4.2 Introduction

Automated Tape Laying (ATL) and Automated Fibre Placement (AFP) are two of the most common automated processes used to produce laminate composite structures. They both rely on the application of continuous carbon or glass fibres, pre-impregnated with a polymer resin (prepreg) onto the surface of a mould. Once the lay-down is complete, the part is placed in a vacuum bag and cured in an autoclave to complete the polymerization of the resin. A comprehensive review of the history of these technologies and the ongoing trends can be found in Dirk et al. (2012). While ATL is capable of bringing accuracy and repeatability to the lay-down process and of reaching high deposition rates by delivering wide prepreg tapes

(70 mm to 300 mm), it can barely deal with curved surfaces. AFP on the other hand has been developed to overcome this weakness. Indeed, instead of laying down wide prepreg tapes, the placement head of an AFP system is capable of delivering up to 32 narrow tows side by side simultaneously. The tow tensions, feeding speeds, and cut locations are controlled independently, allowing for the delivery of the material along curved trajectories and therefore, production of contoured surfaces.

Because of the large range of applications and sensitivity on manufacturing parameters of composites structures, path planning of the AFP toolhead is a critical aspect to produce a reliable and performant final product. Many efforts were reported in the literature dealing with the path planning on open contoured surfaces, e.g. Shirinzadeh et al. (2007, 2004); Blom et al. (2010); Li et al. (2015). A commonly found methodology consists in the definition of a reference curve on the mould surface which is subsequently shifted to produce the different courses composing the ply. The computation of this reference course is usually done by intersecting a reference plane with the mould surface Shirinzadeh et al. (2007); Xiaoping et al. (2008); Yan et al. (2014), mapping a 2D curve onto it Bruyneel et Zein (2013); Lemaire et al. (2015), or by using parametric functions Parnas et al. (2003). This last method seems to be the preferred technique to produce variable stiffness composites in which the courses are intentionally curved to obtain anisotropic mechanical properties Blom (2010).

Once this first curve is defined, it can be shifted along the direction of a defined axis to generate the remaining courses Olmedo et Gurdal (1993). While this method allows to control the shape of each course quite well, it cannot produce perfectly parallel trajectories in all cases, which may yield gaps and overlaps in the ply. These defects are sadly often unacceptable. To solve this issue, it has been proposed to define the courses such that they remain parallel to each other, see for instance Waldhart (1996). To this aim, once the reference curve is known, the next course is computed by shifting the original one of a constant magnitude in a direction locally perpendicular to the curve. It was shown in Shirinzadeh et al. (2007) that this method allows to produce a uniform lay-up on open contoured surfaces without any gaps or overlaps.

While the authors Shirinzadeh et al. (2007) used parametric surfaces and differential geometry to compute the parallel offsets, another approach proposed in Bruyneel et Zein (2013) detailed a solution based on an adaptation of the fast marching method (see Sethian (1999)) and was shown to be able to work with triangulated meshes and thus, handle a large variety of complex shapes as found in practice. For instance, in Lemaire et al. (2015) this method was used to produce trajectories maximizing the stiffness of a structure for a given load.

A common denominator of all the aforementioned works is the fact that they deal with open-

contoured or simple closed surfaces such as cylinders and cones. However, many practical parts with more complex geometries would also benefit from being built using AFP. The purpose of this work is to demonstrate the possibility of manufacturing complex structures such as Y shapes using AFP. To achieve this goal it is proposed to use a 7 degree-of-freedom (DOF) robotic workcell including a 6 DOF serial manipulator holding the toolhead synchronized with a 1 DOF rotary table. This system is selected to have a better dexterity and reach than existing AFP machines thereby allowing to produce complex multi-axial composite structures.

To the best of the authors' knowledge, the generation of trajectories on multi-axis closed surfaces for AFP was never reported in the literature. The analysis proposed here begins in Section 4.3 with a definition of the parts considered in this work and the presentation of a segmentation method. Then, in Section 4.4, the different path generation algorithms considered are described and compared. Subsequently, a finite element analysis (FEA) is presented in Section 4.5 to investigate the advantages of the different shapes of these generated paths. Finally, simulations of the workcell are discussed in Section 4.6.

4.3 Segmentation

While defining tool paths on plates can be done by relatively simple techniques, it can become a much more intricate problem when working on complex 3D surfaces. To handle this complexity, a segmentation strategy is generally used to divide the whole part into simpler regions on which basic path planning algorithms can be applied. The parts considered in this work are assumed to be multi-axial closed surfaces or, more precisely, Y-shaped tubes with circular cross sections. Since they cannot be represented by a single parametric equation, it is assumed that a triangulated mesh of these parts with sufficient resolution is available. This assumption is not restrictive since a great number of CAD software can generate such files. They can even be obtained by 3D scanning. Furthermore, two additional hypotheses are considered, namely : all the branches of the part have the same radius, and all the axes of these branches intersect in a common point (cf. Figures 4.1 and 4.2.)

To ensure proper adherence of the fibres to the part, the orientation of the compaction force exerted by the roller used in the toolhead should be kept normal to the mould surface at all time. Moreover, the axis of rotation of the roller has to be inside a plane perpendicular to the trajectory. Based on these constraints, a local reference frame can be constructed for each position along the path of the tool (see Figure 4.3.) This frame can be decomposed into three perpendicular unit vectors : \mathbf{a}_i , the approach inward-pointing vector normal to the surface at the contact point \mathbf{p}_i , \mathbf{o}_i the orientation vector tangent to the path and belonging to the

tangent plane of the surface, and finally, \mathbf{n}_i the normal vector obtained by : $\mathbf{n}_i = \mathbf{o}_i \times \mathbf{a}_i$. The frame defined as such can be used to obtain the pose of the roller with a 4×4 homogeneous transformation matrix (HTM) : $\mathbf{P}_i = \begin{bmatrix} \mathbf{n}_i & \mathbf{o}_i & \mathbf{a}_i & \mathbf{p}_i \\ \mathbf{0} & & & 1 \end{bmatrix}$.

4.3.1 Segmentation in smooth regions

To define the homogeneous matrix characterizing the pose of the roller, the normals of the surface at any point of the trajectory must be obtained. However, in a triangulated approximation of a real geometry, only the normals of the triangular facets are available. It is therefore necessary to interpolate some of these vectors. To this aim, several methods exist. For instance, in Thürmer et Wüthrich (1998) a method was proposed to accurately compute the normal vectors at any vertices of a triangulated mesh regardless of the tessellation. The algorithm proposed in this reference relies on a weighted average of the neighbouring faces' normal vectors. The weight of each of these vectors is computed as the angle between the edges of the faces incident to the considered vertex. This method has been demonstrated to produce robust estimates by many in the literature including the authors Olivieri et al. (2014a). Once the normal vectors at any vertices of the mesh are known, it is also possible to similarly interpolate them to find the normal vector at any point of a facet.

To proceed with the actual segmentation of the part into smooth regions, several algorithms have also been proposed in the literature, e.g. Olivieri et al. (2014b); Atkar et al. (2009). However, considering the assumptions made here, the surfaces dealt with in this work can be handled much more simply. Indeed, since our parts are constituted by intersecting cylinders, segmentation can be done by clustering neighbouring faces separated by an angle lower than a chosen threshold (see Figure 4.1 for an example.) The computation of the normal vectors is then carried out for each individual region.

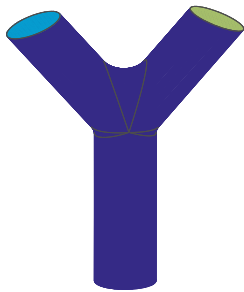


Figure 4.1 Smooth regions

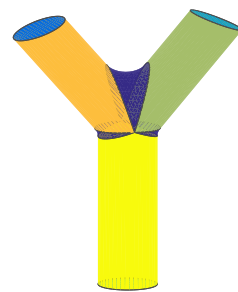


Figure 4.2 Cylinder extraction

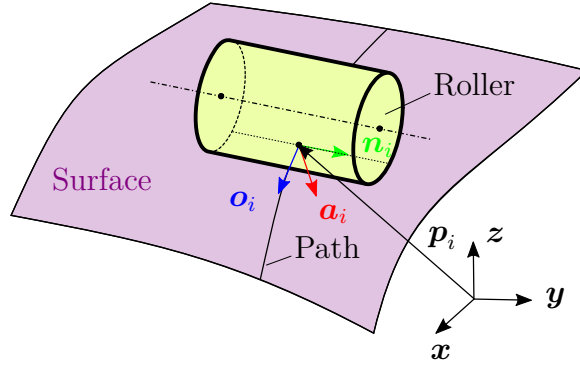


Figure 4.3 Local reference frame

4.3.2 Cylinder extraction

The actual segmentation of the parts considered here mainly consists in identifying cylinders from the list of faces defining the mesh. An efficient solution to do this task was proposed in Tran et al. (2015) from which our work derives. The algorithm proposed in the latter reference allows to extract cylinders from point clouds by selecting an initial set of points potentially belonging to a cylinder using curvature data. The best cylinder fitting these points is then computed and iteratively propagated to neighbouring points. This propagation stops when a set of validation criteria are violated (see Tran et al. (2015).) Following this approach, the centroids of the facets are used in this work as the input points. If a cylinder can be extracted from these data according to the validation criteria, the corresponding facets are clustered into a region associated to the cylinder properties (cf. Figure 4.2 for an illustration.) This identification process allows for the definition of the cylinders radii, axes, lengths, and intersection point. All of which will be useful in the following steps. A local frame is also associated with each identified cylinder with its z -axis parallel to the axis of revolution.

4.4 Path Planning

Once the geometry of the part is properly defined through a series of intersecting cylinders, computation of the covering paths per se must be dealt with. Initially, paths were generated on individual branches of the part. However, a variety of problems arises from this technique, such as gaps and overlaps, numerous cuts, high geodesic curvature, and a difficult connection between the three branches. Clearly, another approach was required using the possible covering of more than one branch with a single course. As will be shown later, this also leads to an improved strength of the final part compared with a single branch coverage. In the following sections, two path generation algorithms are proposed. The first one (Section 4.4.1) aims at

generalizing the constant angle plies generally used for open contoured surfaces by generating paths of constant angle with respect to the revolution axis of the associated cylinder. The second method (Section 4.4.2) allows for the definition of a maximum geodesic curvature for the path in order to avoid fibre buckling during manufacturing.

4.4.1 Constant Placement Angle (CPA)

Despite the fact that a wealth of work is available in the literature dealing with curved trajectories for variable stiffness composite parts, e.g. Lemaire et al. (2015); Blom (2010); Kim et al. (2015), the industry mostly continues to rely on and use standard orientations to define the different plies of the laminate. To follow a similar philosophy, the ply definition is extended in this section to multiple-axis bodies. Similarly to planar surfaces, the ply orientation can be defined with respect to a reference direction. Considering the geometry of the parts discussed in this paper, a natural choice for this reference is the axes of the cylinders. Again, following the standard practices of the industry, three ply orientations were chosen, namely 0° (i.e. parallel to the cylinder axis), 45° and 90° . In order to obtain a continuous course, the 90° ply is approximated by a helix with a pitch equal to the width of the course, i.e. the actual placement angle is slightly smaller than 90° .

For each ply, a path is defined as a constant angle helix and is constructed using the iterative algorithm presented in Han et al. (2012). Briefly, this algorithm is based on a user-defined reference direction (here the cylinder axis) and then, a vector \mathbf{d} tangent to the surface at the current point P_i of the path and pointing towards the reference direction is estimated. A second vector \mathbf{t} is then computed by rotating \mathbf{d} around the surface normal vector \mathbf{n} by the selected placement angle ψ . An intermediate point P_F is thereby computed by translating P_i along \mathbf{t} of a small distance. Finally, P_F is projected onto the mould surface along $-\mathbf{n}$ to obtain the next point of the path, P_{i+1} . See Figure 4.4 for an illustration. This process stops when the boundary of the region to cover is reached, and the final point of the path is then computed on this boundary.

Once the reference curve is completed, it is offset in the orthogonal direction by a geodesic distance equal to the course width by using the method presented in Khakpour et al. (2015). Figure 4.5 illustrates the result of this method for the 0° ply. This leads to a ply without gaps or overlaps as in Blom (2010); Waldhart (1996). However, one of the drawbacks of this method is that the geodesic curvature of the parallel courses can possibly become quite large, depending on the reference curve and the geometry of the part. To illustrate this problem, let us consider the 45° ply illustrated in Figure 4.6. One can see that the further the path is from the reference curve the greater its geodesic curvature. Another issue can also be seen

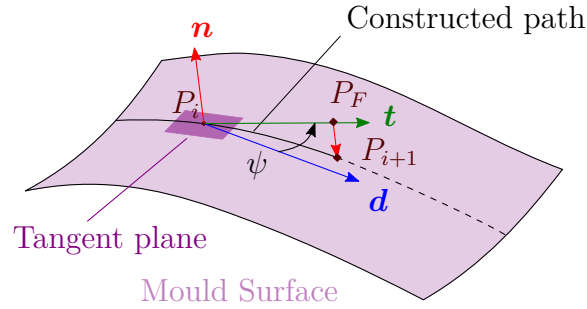


Figure 4.4 Iterative algorithm for path generation

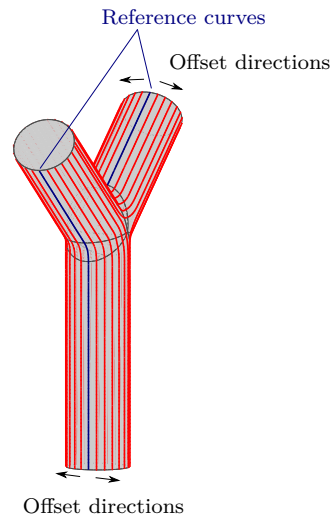
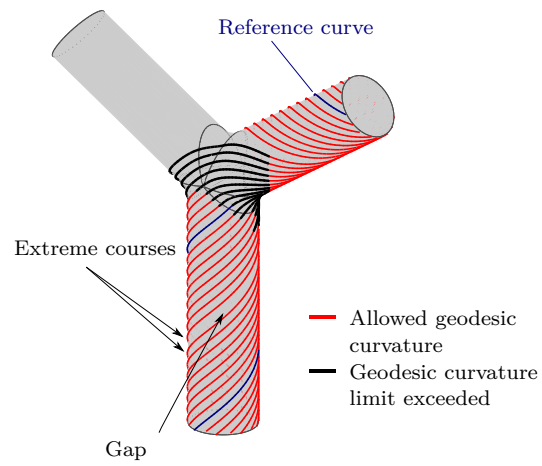
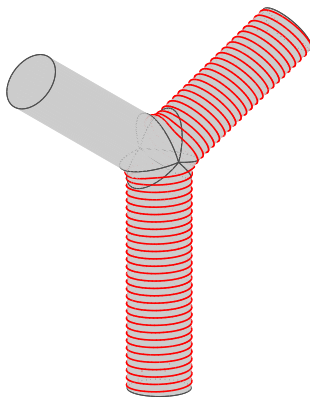
in this Figure, namely that a gap (or in other cases overlap) appears between the extreme courses. It is unavoidable if the perimeter of the base circle of the cylinder is not a multiple of the offset distance. However, it is possible to deal with this problem in practise by using the capacity of the placement head to deliver a variable number of fibre tows and thus, to produce a course with the required width. Moreover, the illustrated placement angle value leads to a partial coverage of the third branch near the junction region. The result for a 90° ply is also illustrated in Figure 4.7 and shows a uniform coverage of the two considered branches but a lack of fibre in the junction region.

Using this technique, it is possible to cover two branches with a continuous course while minimizing gaps and overlaps, and obtain a uniform coverage. Nevertheless, when using the 45° and 90° placement angle, a combination of plies starting from the different branches is required to obtain the same number of layers on each branch. The main drawback of this method is its tendency to create paths with high geodesic curvature around the junction region (cf. Figure 4.6.) This can lead the fibre to buckle during manufacturing and produce parts with severely weakened mechanical properties.

4.4.2 Maximal Curvature Constraint (MCC)

In this section another path planning technique is introduced and investigated where the generated trajectories are covering at least two branches continuously while avoiding geodesic curvatures greater than a selected threshold. Based on results from the literature, e.g. Blom (2010), the typical minimum turning radius feasible in AFP seems to be approximately 635 mm for 3.175 mm wide tows. This value was used for the examples shown in this section. However, it should be noted that this number can be adjusted depending on the tow width and specifications.

To ensure that this maximal curvature is not exceeded, the proposed algorithm first computes a helix path on the starting branch in order to generate a full ply. The rationale of using again

Figure 4.5 0° ply trajectoriesFigure 4.6 45° ply trajectoriesFigure 4.7 90° ply trajectory

a helix is that this curve is a geodesic for cylinders regardless of its pitch (i.e. it has a zero geodesic curvature.) Therefore, the path on the first branch of the part will necessarily satisfy the curvature constraint, namely to have a turning radius greater than 635 mm. However, once the helix reaches the junction region, this property is not necessarily satisfied anymore. Trying to pursue the iterative construction of the path, if the algorithm detects a geodesic curvature greater than the set threshold, it stops there and an optimization is run to compute the next points of the path. Three options for the continuation of the path were selectable by the user in the algorithm : (1) going in the direction of the minimal curvature path, (2) trying to reach a path generating a helix on the second branch with a pitch equal to the course width (within the allowable curvature limit) or (3) trying to reach a path generating a straight line on the second branch still while respecting the curvature constraint.

This algorithm is similar to the one presented in the previous section but this time the angle ψ is variable. Indeed, point P_{i+1} is computed from point P_i in order to satisfy one of the three options listed above. For each point P_{i+1} , a golden section search algorithm is used to compute the placement angle ψ which minimizes a fitness function established from the desired options. For instance in the case of option (1), this function is simply the opposite of the radius of the circle tangent to the path computed at point P_i and belonging to the local tangent plane. For options (2) and (3) the difference between the minimal allowed curvature radius for the path and the obtained curvature radius with P_{i+1} is minimized. This technique is able to produce a path with an acceptable curvature, slowly reaching the targeted placement angle ϕ (cf. Figure 4.8.)

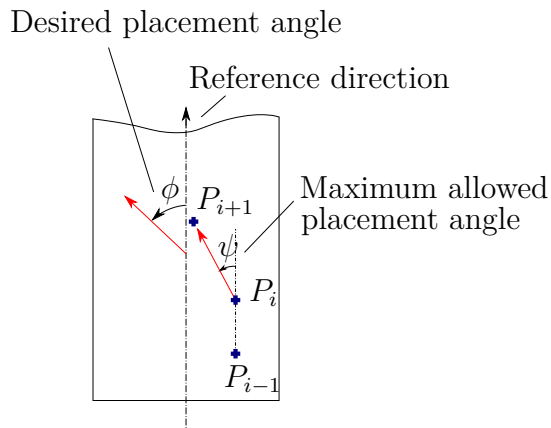


Figure 4.8 Optimization algorithm

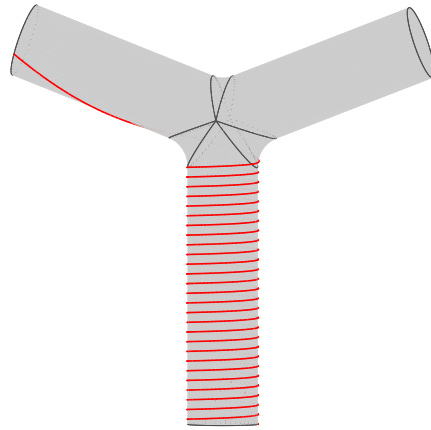


Figure 4.9 Minimum curvature path ending in a straight line

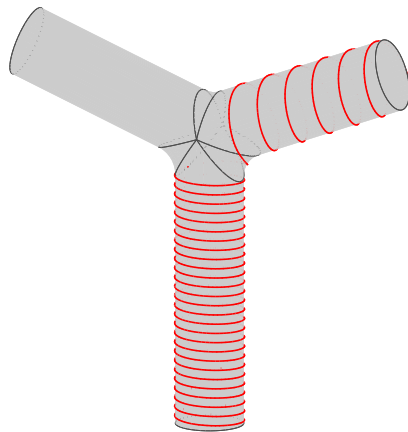


Figure 4.10 Minimum curvature path ending in a helix

4.4.3 Selection Algorithm

Using the previous technique, the second part of the path from the beginning of the junction to the end of the second branch is not predictable since it is computed through an optimization. Different starting points on the initial branch will lead to completely different paths and therefore, mechanical properties for the part. To alleviate this uncertainty an exhaustive set of possible trajectories for each part must be obtained in order to be able to select those adapted for a specific load on the part.

With this in mind, the trajectories were separated in two categories producing the most different mechanical properties (see Section 4.5) : paths finishing in a straight line (cf. Figure 4.9) or in a helix (cf. Figure 4.10.) A selection algorithm is then used with the set of available trajectories from the constituted database. The criteria used in this selection are typically the desired number of layers on each starting branch and the ratio between paths ending in straight lines and helices. The selection algorithm sequentially deletes the unwanted paths from the available set while aiming for the most uniform distribution of the second part of the courses on the branches. Indeed, straight line paths will only cover a small fraction of the second branch, see Figure 4.11 for an example.

4.5 Finite Element Analysis

In this section a finite element analysis is conducted aiming at discussing the effect of different shapes of paths on the structural stiffness of the part. Because the development of a dedicated finite element software was well beyond the scope of this preliminary study, a simplified model was adopted. Indeed, the realistic modelling of each fibre in an AFP produced part is a very challenging issue beyond even most commercial FEA software due to the sheer number of elements required by an accurate simulation. Therefore, an equivalent fibre path was defined to represent the prepreg course. This equivalent fibre has the same mechanical properties as a carbon fibre but with a rectangular cross section whose area is equal to the sum of the cross

Tableau 4.1 FEA results

<i>Case</i>	<i>Part</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
1	Maximum displacement (mm)	1.2836	1.2689	1.2747	1.2172	1.2436
	Displacement difference	0%	-1.15%	-0.69%	-5.17%	-3.12%
2	Maximum displacement (mm)	0.2787	0.2585	0.2629	0.2552	0.2605
	Displacement difference	0%	-7.24%	-5.68%	-8.40%	-6.53%
	Volume of fibre (mm ³)	0	1089	1045	1276	1246

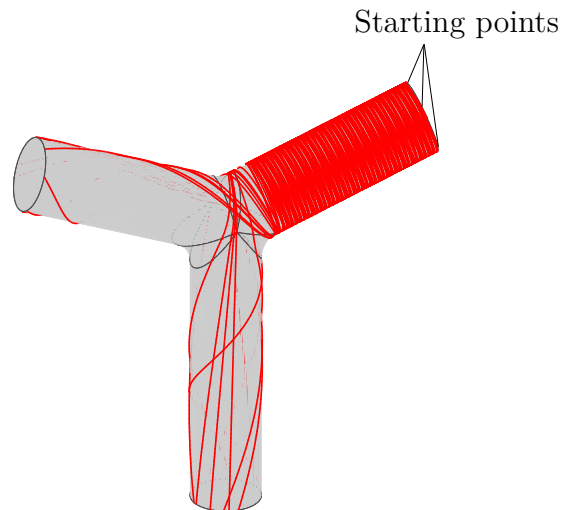


Figure 4.11 Minimum curvature selected paths

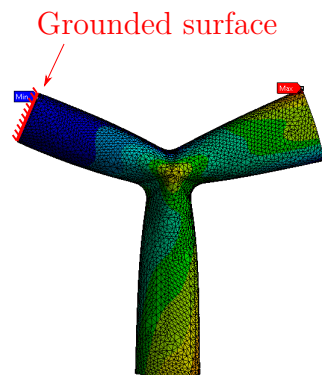


Figure 4.12 Case 1 : internal pressure of 0.1MPa

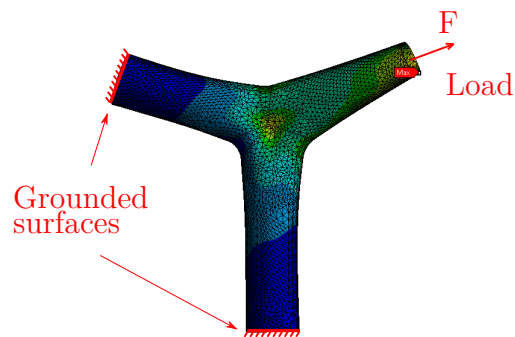


Figure 4.13 Case 2 : traction stresses of 100N

section areas of all the microscopic fibres contained in the prepreg. This equivalent fibre, also defined to have approximately the same thickness than the cured prepreg tape, is assumed to stay tangent to the part surface.

The proposed procedure to characterize the effect of the path trajectory is to compare the behaviour of a part entirely made of epoxy resin with one part having the same geometry but including one equivalent fibre with a particular shape. In this paper, five different parts are considered. The first one is the reference part made of epoxy only. The second and third parts both contain an equivalent fibre ending in a straight line (cf. Figure 4.9.) The latter are identical except for one important distinction : the equivalent fibre of the third part is cut at and removed from the junction region. Parts four and five both have an equivalent fibre shaped as a helix (cf. Figure 4.10.) Similarly to the previous case, this fibre is cut and removed in the junction region of the fifth part.

Two load cases are considered for these five parts. The first one corresponds to an internal pressure (cf. Figure 4.12) while the second corresponds to exerting traction stresses in the branch where the equivalent fibre ends (see Figure 4.13.) For each case, the maximal displacement of the extremity of the end branch of the fibre is measured. Results are presented in Table 4.1.

These results corroborate that indeed, a continuous fibre running through two branches will lead to a greater stiffness. More specifically, parts 3 and 5 have significantly greater displacements than respectively parts 2 and 4. The two cases considered here also show that, for an equivalent volume of fibre, helix shaped paths are best suited to resist an internal pressure while straight lines are the best choice when traction stresses are anticipated. These results reinforce the assumption made before that two separate sets of trajectories should be considered. The last insight uncovered by this analysis is that for each load case, the maximal deformation generally occurs in the junction region and therefore, special attention must be paid at the path planning stage to make sure that this region is appropriately covered.

4.6 Simulations

As illustrated in Figure 4.11, the paths generated by the method proposed here are quite complex. Therefore, they require a high dexterity which is the reason why a 7 DOF robotic workcell was proposed. This workcell is constituted by a 6-DOF serial manipulator completed by a 1-DOF rotary table. Since the task of laying down fibres requires 6-DOF, this workcell is kinematically redundant. In this section, a method to solve this redundancy, i.e. to translate the path from the Cartesian space to the joint space of the workcell, is presented based on Huo et Baron (2008). The redundancy is helpful in our case and can be used to handle auxiliary tasks such as avoiding joint limits and singularities or minimizing displacements

between two consecutive poses. The simulation of the workcell¹ is illustrated in Figure 4.14. Solving the inverse kinematics of a general n -DOF redundant serial manipulator can be realized by considering the linear relationship between the twist \mathbf{t} of the end effector (EE) of the manipulator and the joint velocity vector $\dot{\boldsymbol{\theta}}$. These vectors can be defined as $\mathbf{t} = [\boldsymbol{\omega}^T \ \dot{\mathbf{p}}^T]^T$ and $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1 \ \dots \ \dot{\theta}_n]^T$ where $\boldsymbol{\omega}$ is the angular velocity of the EE, $\dot{\mathbf{p}}$ the velocity of a point of the EE and $\dot{\theta}_1$ to $\dot{\theta}_n$ are the angular velocities at the joints.

Considering the small displacements $\Delta\mathbf{t}$ and $\Delta\boldsymbol{\theta}$ instead of velocities, the relationship between these vectors becomes :

$$\Delta\mathbf{t} = \mathbf{J}\Delta\boldsymbol{\theta} \quad (4.1)$$

where \mathbf{J} is the Jacobian matrix of the manipulator. Since this matrix is not square with redundant manipulators, the solution introduced by Liégeois (Liegeois (1977)) to solve the underdetermined linear algebraic system of Equation (4.1) can be used, namely :

$$\Delta\boldsymbol{\theta} = \mathbf{J}^\dagger\Delta\mathbf{t} + (\mathbf{I}_7 - \mathbf{J}^\dagger\mathbf{J})\mathbf{h} \quad (4.2)$$

where $\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ is the generalized inverse of the Jacobian matrix, \mathbf{I}_7 is the 7×7 identity matrix, and \mathbf{h} is a vector defining the auxiliary tasks to handle. The first term on the right-hand side of Equation (4.2) is the minimal norm solution of (4.1) while the second term is a projection of \mathbf{h} into the null space of \mathbf{J} . This vector has therefore no influence on the velocity at the EE of the manipulator but has an effect on joint velocities and can be used to perform the auxiliary tasks.

In Huo et Baron (2008), Equation (4.1) was derived for redundant serial manipulators. However, the problem here is slightly different since the redundancy is due to the introduction of the rotary table. It is nevertheless possible to show that Equation (4.1) holds true even in this case by defining a Jacobian matrix of the whole workcell.

To this aim, let $\Delta\mathbf{t}_m$ be a small displacement from the current pose of the manipulator, described by the HTM \mathbf{P}_i , to the desired pose of HTM \mathbf{P}_{i+1} , defined on the part attached to the spindle of the rotary table. It is possible to define the opposite small displacement $\Delta\mathbf{t}_s$ that this spindle must produce to make this desired pose correspond to the pose of the EE of the manipulator (cf. Figure 4.15), i.e. $\Delta\mathbf{t}_m = -\Delta\mathbf{t}_s = \Delta\mathbf{t}$. Two equations are then obtained :

$$\Delta\mathbf{t} = \mathbf{J}_m\Delta\boldsymbol{\theta}_m \quad (4.3)$$

$$\text{and} \quad -\Delta\mathbf{t} = \mathbf{J}_s\Delta\boldsymbol{\theta}_s. \quad (4.4)$$

1. All simulations were done with RoboDK : <https://www.robodk.com/>

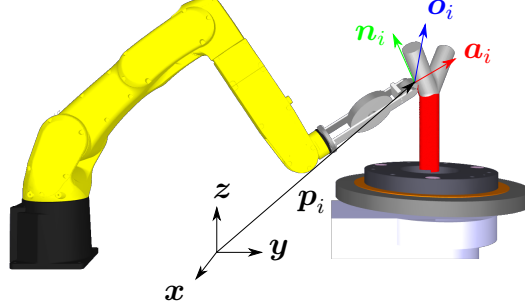


Figure 4.14 Workcell definition.

Where \mathbf{J}_m and $\boldsymbol{\theta}_m$ are respectively the Jacobian matrix and the joint angles of the manipulator, while \mathbf{J}_s and θ_s are the ones of the spindle. Equations (4.3) and (4.4) can be arranged into :

$$\Delta \mathbf{t} = \frac{1}{2} \begin{bmatrix} \mathbf{J}_m & -\mathbf{J}_s \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\theta}_m \\ \Delta \theta_s \end{bmatrix} = \mathbf{J} \Delta \boldsymbol{\theta}. \quad (4.5)$$

Since vector \mathbf{h} can be used to control the position of the links of the robot without affecting the pose of the end effector, this vector can be defined in order to minimize a performance function z , chosen here as a quadratic function of $\boldsymbol{\theta}$. This quadratic function is firstly used to avoid joint limits and singularities of the serial manipulator and, since the main part of a path generated by our algorithms is a circular helix, to take advantage of the infinite range of the rotary table to minimize the displacements of the manipulator. Three corresponding performance indexes, respectively z_{joint} , z_{sing} and z_{disp} , are then defined as :

$$z_{joint} = \frac{1}{2} (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T \mathbf{W}_1^T \mathbf{W}_1 (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) \quad (4.6)$$

$$z_{sing} = \frac{\omega_{ps}^2}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts})^T \mathbf{W}_2^T \mathbf{W}_2 (\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts}) \quad (4.7)$$

$$z_{disp} = \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_{old})^T \mathbf{W}_3^T \mathbf{W}_3 (\boldsymbol{\theta} - \boldsymbol{\theta}_{old}) \quad (4.8)$$

where $\bar{\boldsymbol{\theta}}$ are the middle positions of the joints' ranges, $\boldsymbol{\theta}_{Ts}$ the recorded values of $\boldsymbol{\theta}$ when ω_{ps} the manipulability of the workcell exceeds a fixed threshold (see Huo et Baron (2008)), and $\boldsymbol{\theta}_{old}$ the joint positions of the system at the previous point of the trajectory. The matrices $\mathbf{W}_i = \text{diag}(\mathbf{w}_i)$ are diagonal weight matrices. The vectors \mathbf{w}_1 and \mathbf{w}_3 are set such that their last component, associated with the rotation of the rotary table spindle is zero. Therefore, the optimization favours a wide range of motion of this axis. The other components of \mathbf{w}_i were arbitrarily set to obtain a proper motion of the manipulator. The performance index ω_{ps} represents the manipulability of the system (Angeles (2002)). It is here defined using the

singular values σ_i of \mathbf{J} :

$$\omega_{ps} = \sqrt{\frac{1}{\sigma_2 \sigma_3 \dots \sigma_7^2}}. \quad (4.9)$$

High values of ω_{ps} mean a low manipulability, i.e. a poor condition number of \mathbf{J} . This corresponds to singular configurations where the ability of the robot to move along a particular direction is lost, a situation that must be avoided. Finally, \mathbf{h} is defined as the opposite of the gradient of z .

$$\mathbf{h} = -\nabla z = -\nabla(z_{joint} + z_{sing} + z_{disp}). \quad (4.10)$$

A video of the simulated AFP worcell using the proposed algorithms can be found at : <https://www.youtube.com/watch?v=rW8igFNMHVE>.

4.7 Discussion and future works

As it was underlined in Section 4.4.1, the principal weakness of the CPA method was its tendency to produce paths with high geodesic curvatures. With conventional AFP techniques such trajectories generally lead to fibre buckling. However, new techniques are being developed that would allow to produce these paths without inducing defects. One of these promising methods is continuous tow shearing (CTS), introduced in Kim et al. (2012). The authors assert that it is possible to produce trajectories with a minimal curvature radius less than 50 mm with this technique, without any fibre buckling. Thus, it appears that upcoming AFP techniques might be able to use the trajectories produced by the proposed CPA algorithm.

The MCC algorithm presented in this work provides a tool to generate feasible paths on Y-shaped parts for current AFP manufacturing. While this algorithm has been designed to avoid fibre buckling, it also suffers from a few disadvantages, first of which is that it does not lead to a uniform coverage. Moreover, the conducted simulations revealed that this algorithm frequently leads to a lack of fibre in the junction region which is the area showing the greatest deformations in the FEA. This problem could be solved by manually adding prepreg reinforcement in this region before curing the part.

The selection algorithm also presented here allows for the determination of a set of desired shapes for these paths and it was noticed that some of them are infrequent amongst the available choices. To circumvent this issue requires the computation of a large amount of initial trajectories which in practice demands long computation times. For instance, the average computation time of a unique path being approximately 45 seconds with a standard computer, the generation of a set of 360 trajectories on a unique branch (i.e. with starting points distributed every degree on the base circle of this branch) would last up to 4.5 hours.

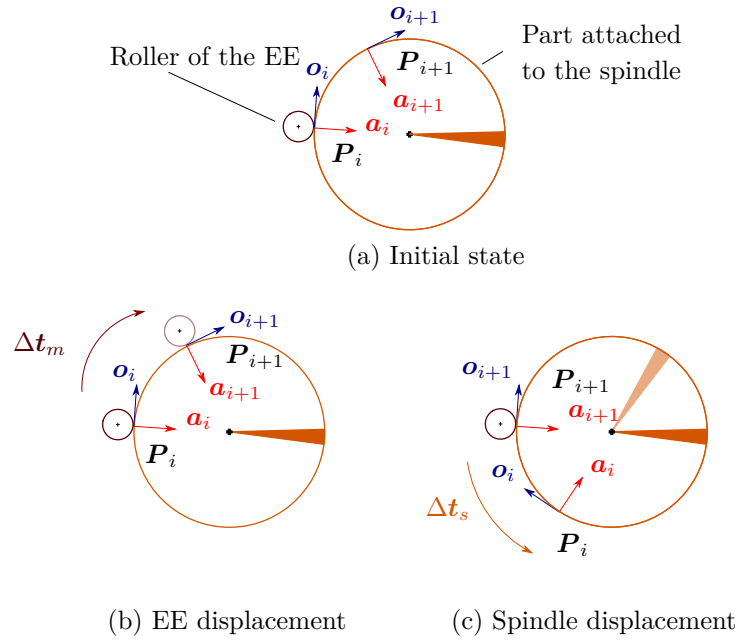


Figure 4.15 Definition of Δt_m and Δt_s

Moreover, the selection algorithm is not automated at this point and the input from the user is required. A major improvement and useful feature would be to embed the FEA simulations in the selection process to produce a part optimized for a load case. However, this seems impractical due to the complexity of the FEA unless a simplified model could be found.

Another important practical issue neglected in this work lies in manufacturing constraints. More specifically, it was assumed that a placement head small enough to move around the part existed. However, most placement heads presented in the literature and used in the industry are quite large and it is easy to see that such design would often collide with the part, the mandrel or the robot. While collisions with the robot could be partially avoided using the redundancy of the system, collision with the mandrel are more difficult to avoid since they are intrinsically linked by geometry. A solution to minimize this issue is to design a toolhead specifically suited for the production of parts with a Y-shaped geometry.

Finally, the results obtained by simulation indicate that the proposed method can be used to compute the paths in the joint space of the workcell. It also revealed that the relative localization of the rotary table with respect to the robot is an important step to obtain feasible paths. While this has been done by trial and error in this work, a deeper investigation to find an optimized layout of the workcell would be useful. Indeed, even with the setup proposed here, some paths remain impossible to obtain because the robot poses are out of reach due to physical joint limits. A possible solution would be to add more DOFs to the workcell by

using a second robot holding and tilting the part when necessary.

4.8 Conclusion

In this paper, the path planning on the surface of multi-axial closed surfaces was investigated. Two algorithms were proposed to generate trajectories on Y shaped tubes by covering two branches with a single course. While the first one produces a uniform coverage, it suffers from high geodesic curvature in the branches intersection region, possibly leading to defects in the final part due to fibre buckling. The second was designed to avoid this problem and is able to produce paths of different shapes, always starting from a helix on the first branch. The selection of these shapes is realized based on adjustable criteria, that can be chosen depending on the results of a FEA. Finally, simulations were carried out to underline the challenges still remaining before implementation on a real workcell.

Acknowledgements

The financial support of the Fonds de Recherche du Québec – Nature et Technologies (project 182702) is gratefully acknowledged.

CHAPITRE 5 ASPECTS MÉTHODOLOGIQUES ET RÉSULTATS COMPLÉMENTAIRES

Nous avons choisi de ne présenter dans l'article (chapitre 4) que deux algorithmes permettant la génération de trajectoires sur des surfaces fermées multi-axiales de type tubes en « Y », ainsi que les différentes étapes permettant leur définition. Cependant, avant de se concentrer sur les surfaces fermées multi-axiales, les travaux effectués dans cette recherche ont permis de mettre au point des méthodes de planification de trajectoires sur des surfaces fermées de formes plus générales. Ce chapitre apporte plus de précisions sur les travaux complémentaires menés lors de cette étude et qui n'ont pas été traités dans l'article. Il permet également de mettre en lumière les réflexions qui ont menés jusqu'aux travaux présentés dans l'article.

5.1 Directions principales de courbure

Comme nous l'avons vu au chapitre 2, il existe plusieurs méthodes pour générer une courbe sur une surface, qui dépendent aussi bien de la surface traitée que de la manière dont cette dernière est représentée. Le but de ce travail étant de générer des trajectoires sur des pièces présentant des surfaces fermées, il a été décidé d'utiliser l'algorithme itératif présenté par Han et al. (2012) pour générer ces courbes. Comme expliqué au paragraphe 4.4.1, cet algorithme, utilisé pour générer la courbe de référence d'une couche, nécessite la définition d'une direction de référence. L'idée de ce projet étant de travailler sur des pièces de formes complexes pouvant présenter différentes orientations dans l'espace (comme un cadre de vélo, des tubes en « S », voir figure 5.1, ou les pièces en « Y » discutées dans le chapitre précédent), il a semblé judicieux d'utiliser les propriétés intrinsèques des pièces traitées afin de définir cette direction principale. Pour cette raison, les directions principales de courbure de la surface des pièces ont donc été choisies pour définir cette direction principale. En effet, ces propriétés donnent les deux directions dans lesquelles la surface de la pièce présente les courbures extrêmes. De plus ces deux directions sont par définition toujours orthogonales. La méthode utilisée pour calculer ces directions est celle présentée par Goldfeather et Interrante (2004). Cette technique utilise des notions de géométrie différentielle que nous rappellerons dans le paragraphe qui suit.

Lorsque qu'une surface paramétrée $\mathbf{X}(u, v)$ est localement lisse, il est possible de l'approximer par le développement limité $z = L\frac{x^2}{2} + Mxy + N\frac{y^2}{2} + \epsilon$, avec $L = \frac{\partial^2 \mathbf{X}^T}{\partial u^2} \mathbf{n}$, $M = \frac{\partial^2 \mathbf{X}^T}{\partial u \partial v} \mathbf{n}$ et $N = \frac{\partial^2 \mathbf{X}^T}{\partial v^2} \mathbf{n}$, \mathbf{n} étant la normale à la surface au point considéré. Dans ce cas on appelle l'application $\Pi = Ldx^2 + 2Mdx dy + Ndy^2$ la seconde forme fondamentale. Cette forme quadratique peut

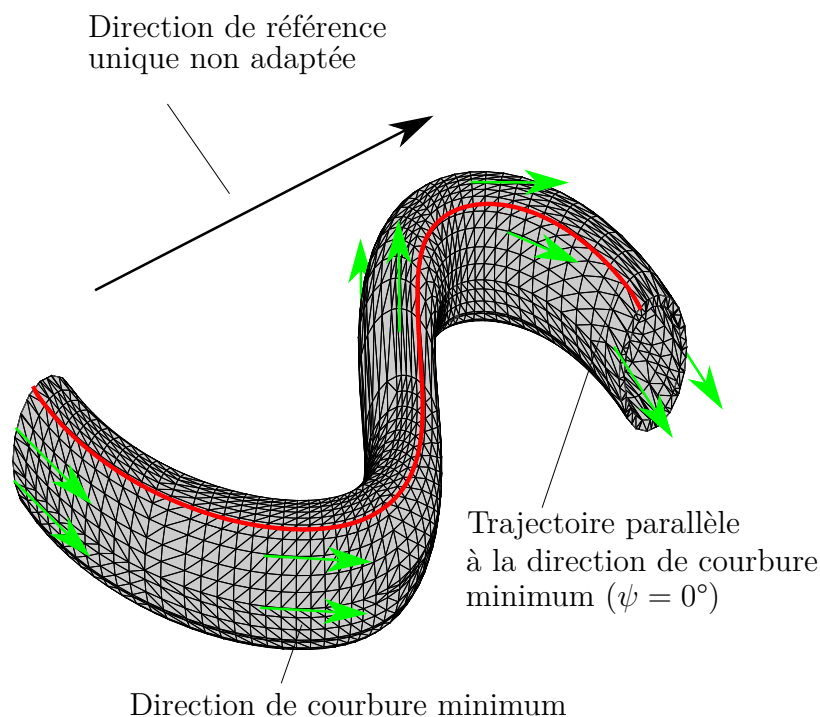


Figure 5.1 Définition D'une trajectoire grâce à la direction de courbure minimale de la surface

également s'écrire sous forme matricielle (parfois appelée matrice de Weingarten) :

$$\Pi = \begin{bmatrix} L & M \\ M & N \end{bmatrix} \quad (5.1)$$

Il est possible de montrer (Abbena et al. (2006)) que les courbures principales de la surface sont les valeurs propres de cette matrice et que les directions principales de courbure correspondantes en sont les vecteurs propres. Dans le cas de l'utilisation de maillage triangulé le paramétrage de la surface est inconnu. Goldfeather et Interrante proposent alors une méthode utilisant une approximation locale cubique pour calculer les composantes de la matrice de Weingarten. Cette méthode utilise les données supposées connues des normales aux sommets du maillage. Cela permet alors de définir les directions principales de courbure de la surface étudiée d'une manière plus robuste et moins bruitée que les méthodes utilisant des approximations quadratiques. Pour plus de détails, le lecteur se référera à l'article sus-cité. La figure 5.2 illustre la direction de courbure minimale calculée sur un tube courbé de section circulaire. On comprend ici bien en quoi l'utilisation d'une telle propriété est intéressante. En effet on remarque que la direction de courbure minimale est toujours colinéaire à l'axe équivalent de la pièce. Cet axe équivalent est ici défini comme étant localement la normale au plan coupant la pièce selon une section circulaire, passant par le centre du cercle formé par

l'extérieur de la pièce. Cela permet alors de généraliser facilement à des pièces de géométries plus complexes les types de trajectoires présentées en section 2.1 telles que des trajectoires colinéaires à cet axe équivalent ou des trajectoires de type enroulement hélicoïdal, présentant toujours le même angle par rapport à l'axe équivalent (figure 5.3). Le calcul de ces propriétés géométriques a pu servir de base pour définir les différents algorithmes présentés au chapitre précédent.

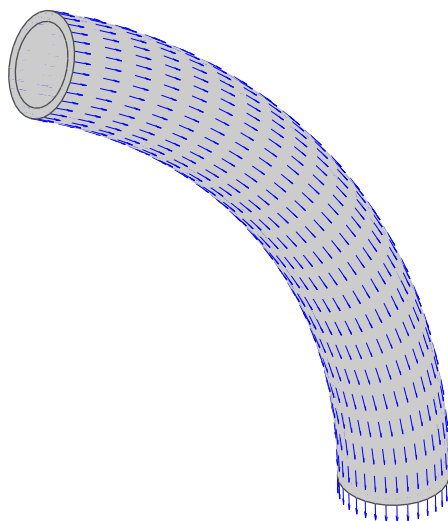


Figure 5.2 Direction de courbure minimale

Certains types de pièces peuvent cependant poser un problème lors de l'utilisation de cette propriété pour définir la direction de référence pour le placement de fibre. En effet, dépendamment de la géométrie de la pièce, les directions de courbures minimales et maximales d'une surface peuvent s'inverser (figure 5.4). Lorsque l'on définit une de ces deux directions comme référence pour une telle pièce, il est alors nécessaire d'ajouter un algorithme vérifiant si une inversion ne se produit pas sur l'élément du maillage sur lequel est calculé le dernier point de la trajectoire. Si c'est le cas, il faut donc changer la direction de référence utilisée afin de garder une cohérence avec le début de la trajectoire.

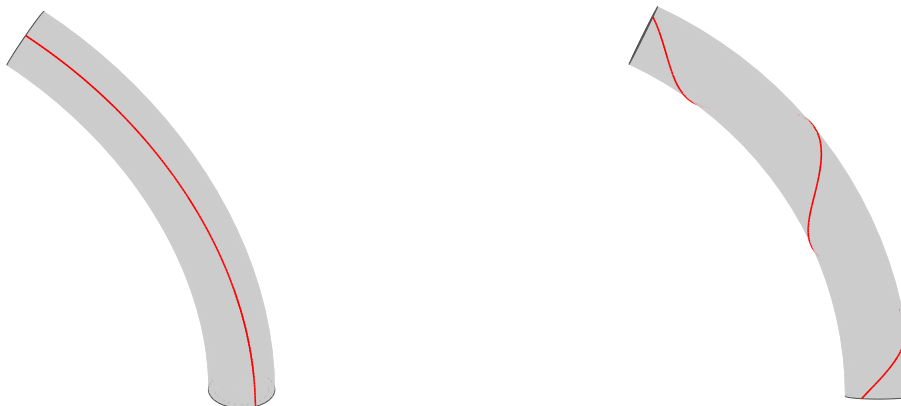


Figure 5.3 Trajectoires ayant un angle constant par rapport à l'axe équivalent du moule (gauche : 0° , droite : 45°)

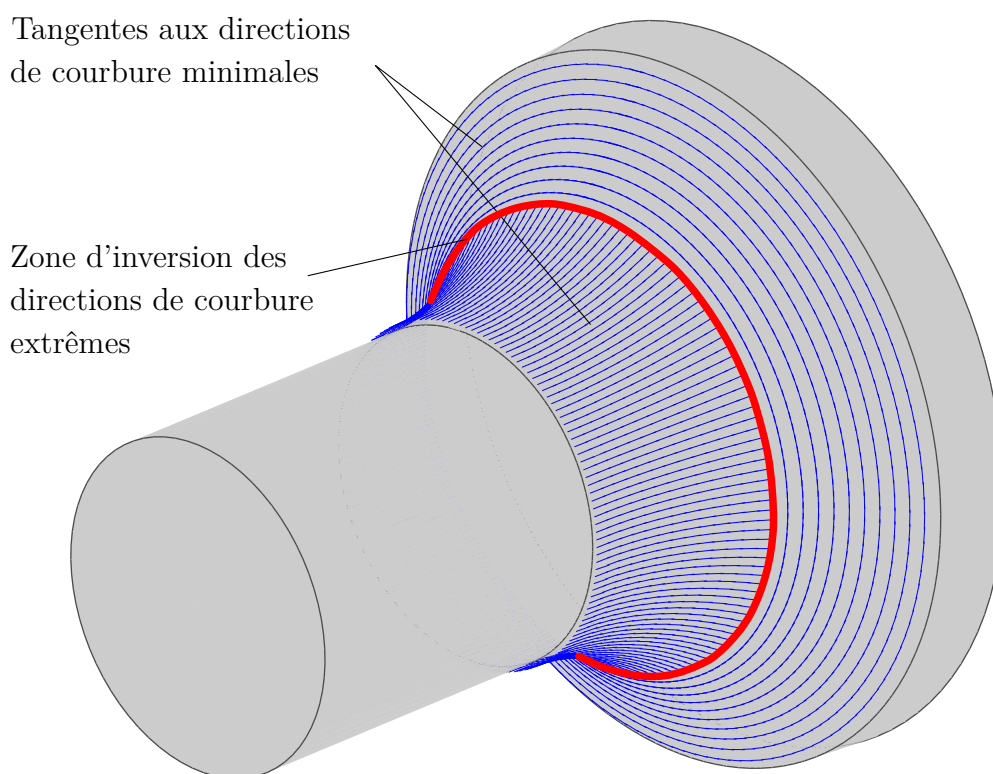


Figure 5.4 Inversion des directions de courbure principales

5.2 Discontinuités aux frontières entre régions

Comme expliqué précédemment, la définition de trajectoires sur des surfaces fermées nécessite la connaissance de certaines propriétés géométriques de la surface de travail, en particulier les directions principales de courbures qui servent de référence à l'algorithme itératif et les

normales à la surface qui permettent la définition de la pose de l'outil. Cependant, il a été montré dans l'article (chapitre 4.3.1) que pour obtenir des résultats précis lors du calcul de ces propriétés, il était nécessaire de travailler sur des régions lisses. C'est pourquoi une méthode de segmentation en régions lisses a été introduite. Si cette division permet le calcul sur chaque région de propriétés géométriques qui sont effectivement continues et lisses, ces deux propriétés ne sont plus assurées sur la frontière entre deux régions. Il est donc possible d'observer, pour deux points proches sur la surface, mais appartenant à deux régions différentes, une variation importante de leur normale à la surface. Les poses définissant la trajectoire de l'extrémité de l'outil étant définies à partir de ces propriétés, il apparaîtra donc également une importante variation dans leur définition.

Afin d'éviter les problèmes liés à ces discontinuités lors de l'implémentation sur un robot réel (comme une mauvaise interpolation entre deux poses appartenant à deux régions différentes), il a été choisi de définir, à chaque fois que la trajectoire traverse une frontière, deux poses situées à la même position sur celle-ci. La première étant calculée avec les propriétés géométriques de la première région (\mathbf{n}_{R_1} , \mathbf{o}_{R_1} et \mathbf{a}_{R_1}) et la seconde avec celles de la région suivante (\mathbf{n}_{R_2} , \mathbf{o}_{R_2} et \mathbf{a}_{R_2}), voir figure 5.5. Cela va donc permettre au robot de changer d'orientation de l'outil dès l'entrée dans la nouvelle région et ainsi de maintenir une pression du rouleau de compaction toujours normale à la surface du moule.

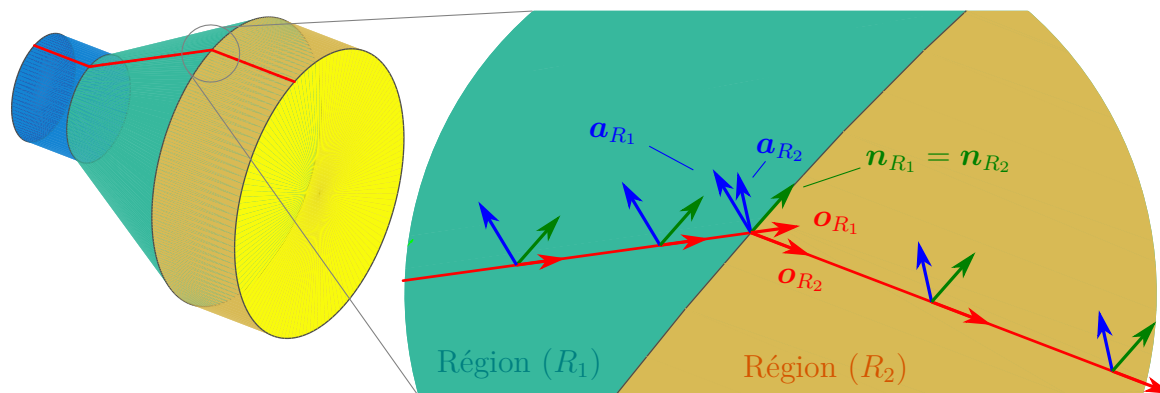


Figure 5.5 Définition de deux poses sur une frontière

5.3 Méthode de l'hélice discontinue (HD)

Après avoir mis au point les méthodes permettant de générer des courbes de référence sur des surfaces fermées de formes quelconques, nous avons décidé de nous concentrer sur les surfaces fermées multi-axiales telles que des tubes en forme de « Y ». Pour couvrir de telles géométries,

deux options ont été identifiées. La première, présentée dans ce chapitre, consiste à couvrir en intégralité chaque branche du « Y » de manière individuelle, puis de joindre ces branches en définissant une trajectoire sur la région jonction. La seconde, qui a été présentée au chapitre 4, consiste à couvrir à l'aide d'une trajectoire continue (sans coupe) deux branches du « Y ». Cela permet alors, à la fois de supprimer l'étape d'association des branches entre elles et d'apporter plus de rigidité à la structure en évitant de couper la fibre. Nous présenterons dans les paragraphes suivants les méthodes mises au point pour couvrir les « Y » en utilisant la première option.

La méthode présentée dans ce paragraphe est celle de l'hélice discontinue. Dans cette méthode le but est de couvrir entièrement chaque branche du tube en « Y » individuellement, sans pénétrer dans la région de la jonction ou sur une autre branche. La méthodologie globale de cette méthode est de calculer sur le cylindre une trajectoire géodésique (c'est-à-dire de courbure géodésique nulle) en forme d'hélice ayant un pas égal à la largeur de la bande de préimprégné déposée. Cette technique est connue dans la littérature pour produire des cylindres stratifiés en composite et possède l'avantage de permettre d'éviter les manques et les recouvrements. Cependant il est clair que dans le cas des pièces en forme de « Y », des problèmes vont être rencontrés au niveau de la jonction, particulièrement lorsque deux des branches du « Y » forment un angle de faible valeur.

Comme abordé au chapitre 4, plusieurs méthodes existent pour générer des courbes sur des surfaces. Une des plus commune en *AFP* est de calculer l'intersection d'un plan de référence avec la surface du moule. Si cette méthode est particulièrement adaptée pour travailler sur des surfaces courbes ouvertes car elle permet de générer des trajectoires orientées par rapport à une direction de référence, il n'est pas possible de l'utiliser pour travailler sur des surfaces fermées comme les tubes en « Y » étudiées ici. L'algorithme itératif présenté en section 4.4.1 sera donc utilisé dans ce cas pour générer la trajectoire. La direction de courbure minimale du cylindre (équivalente à la direction de son axe de révolution) est ici choisie comme direction de référence. L'angle de placement ψ est calculé de manière à obtenir un pas égal à la largeur de bande déposée, pour assurer une couverture complète et uniforme du moule.

L'hélice est alors générée de manière itérative à partir d'un point de départ situé sur le cercle de base du cylindre concerné (côté opposé à celui où se situe l'intersection des cylindres). Ce calcul est stoppé lorsque le dernier point créé est référencé comme appartenant à la région de la jonction. Si la surface de la branche n'a pas encore été totalement couverte, il est nécessaire de redémarrer l'hélice depuis l'autre côté du congé. Pour cela, une hélice virtuelle est calculée, poursuivant celle déjà créée sur la pièce, et les points appartenant à cette hélice virtuelle sont successivement projetés sur la surface de la pièce (figure 5.6, courbe bleue) jusqu'à arriver à

nouveau sur la surface de la branche qui est en train d'être couverte. Le point appartenant à l'hélice et situé sur la frontière entre les deux régions est ensuite recherché et l'algorithme itératif redémarre pour générer la suite de la trajectoire sur la branche à couvrir. Ce processus est répété jusqu'à ce que la surface totale de la branche soit couverte.

Bien que cette méthode soit efficace pour couvrir toute la surface d'une branche (figure 5.7), le grand nombre de coupes et de redémarrages à opérer dans le pré-imprégné va considérablement ralentir le procédé. De plus il est envisageable que le produit final présente des faiblesses mécaniques au niveau de ces coupes. Dans le but de s'affranchir de ces problèmes, une autre méthode de couverture individuelle est présentée dans le paragraphe suivant.

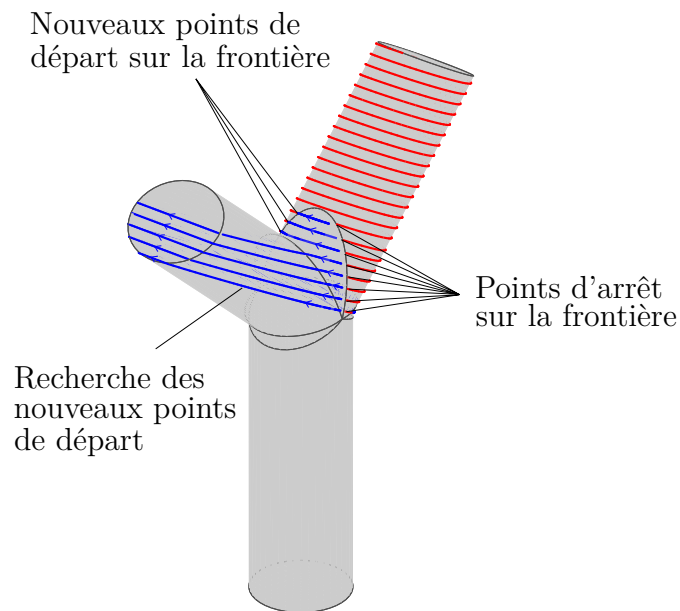


Figure 5.6 Recherche des nouveaux points de départ pour le calcul de l'hélice. Les lignes bleues représentent le processus de recherche des nouveaux points de départ situés sur la frontière entre la jonction et la branche couverte.

5.4 Méthode de l'hélice ajustable (HA)

De manière similaire à la méthode précédente, des hélices vont être utilisées pour couvrir chaque branche individuellement. Cependant l'équation de l'hélice a été modifiée de manière à ajouter la possibilité de déformer la trajectoire générée. Cela va alors permettre d'éliminer les coupes et redémarrages et permettre de couvrir le maximum de surface d'une branche en entrant le moins possible dans la région jonction. L'algorithme itératif n'est plus utilisé ici et la trajectoire sera générée directement grâce à l'équation paramétrique exprimée dans le repère local du cylindre concerné :

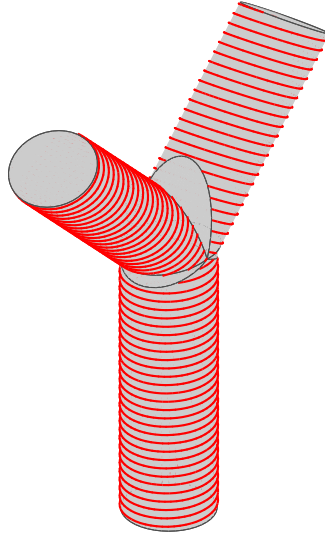


Figure 5.7 Méthode hélice discontinue : couverture des trois branches

$$\left\{ \begin{array}{l} x(t) = R \cos(t) \\ y(t) = R \sin(t) \\ z(t) = ht + \underbrace{A \times f(t) \cos(P(t + \phi))}_{\text{terme d'ajustement}} \end{array} \right. \quad (5.2)$$

Une fois la forme désirée obtenue pour la trajectoire, les points sont projetés sur la surface réelle du modèle CAO pour plus de précision. Dans l'équation (5.2), on retrouve l'équation paramétrique d'une hélice circulaire, additionnée d'un terme permettant son ajustement. Dans ce terme, $f(t)$ est une fonction utilisée pour contrôler l'amplitude de la déformation de l'hélice le long de la branche. Comme il est ici souhaité avoir le maximum de déformation au niveau de la jonction afin de permettre à la trajectoire de s'adapter à la géométrie du congé, $f(t)$ est définie comme étant une fonction continue, équation (5.3), variant entre 0 au départ de l'hélice et 1 à la fin de la trajectoire dans la zone proche de la jonction :

$$f(t) = at^3 + bt^2 \quad (5.3)$$

La forme de cette fonction est illustrée à la figure 5.8. Dans l'équation précédente, a et b sont calculés en fonction du nombre de points composant l'hélice de manière à obtenir la forme voulue pour la fonction f (figure 5.8).

Dans le terme d'ajustement, le paramètre A permet le contrôle de l'amplitude de la défor-

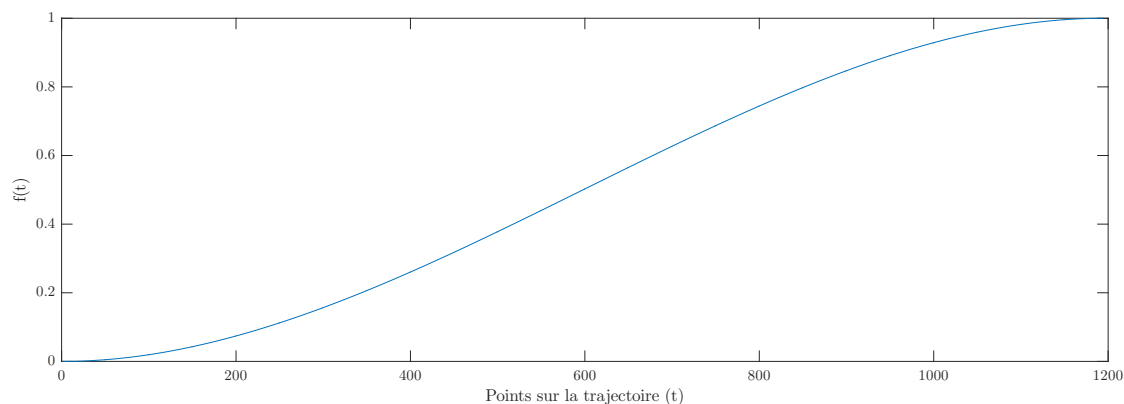


Figure 5.8 Forme de la fonction $f(t)$

mation et autorise une adaptation à différents types de géométries, c'est-à-dire différentes valeurs d'angles entre les branches. La figure 5.9 présente une trajectoire déformée sur deux pièces différentes dont les branches supérieures forment des angles de 110° (gauche) et 60° (droite).

P est utilisé pour contrôler la période de la déformation. Cela permet de traiter des pièces comprenant plus de trois branches si celles-ci sont réparties de manière périodique autour de la branche sur laquelle est calculée la trajectoire (figure 5.10).

Finalement, ϕ permet d'ajuster la phase de la déformation. Cela peut être utilisé pour faire « tourner » la déformation autour de l'axe du cylindre. La figure 5.11 illustre l'effet de ce paramètre. Le point de départ choisi n'étant pas le point recommandé, il est nécessaire d'ajuster ϕ pour placer la déformation à l'endroit voulu.

Cette méthode possède l'avantage de ne pas nécessiter de coupes et de redémarrages dans la bande de pré-imprégné déposée. Cependant, elle s'est révélée posséder d'autres inconvénients. En premier lieu, comme l'hélice est déformée, son pas n'est pas conservé, ce qui conduit à l'apparition de manque de matière et de recouvrement dans une couche. De plus, il est évident que plus l'angle entre deux branches est faible, plus il est difficile de couvrir toute la surface de la branche considérée (figure 5.9). Finalement, comme illustré dans les figures 5.12 et 5.13, la trajectoire ainsi obtenue conduit à l'apparition de zones à forte courbure géodésique dont les conséquences ont déjà été discutées dans l'article (chapitre 4). On notera que la figure 5.13 permet d'observer que l'évolution de l'enveloppe de l'amplitude de la courbure géodésique suit la forme de la fonction $f(t)$. Ceci est tout à fait logique puisque cette fonction permet de contrôler l'amplitude de la déformation de l'hélice et donc son éloignement par rapport à une géodésique du cylindre.

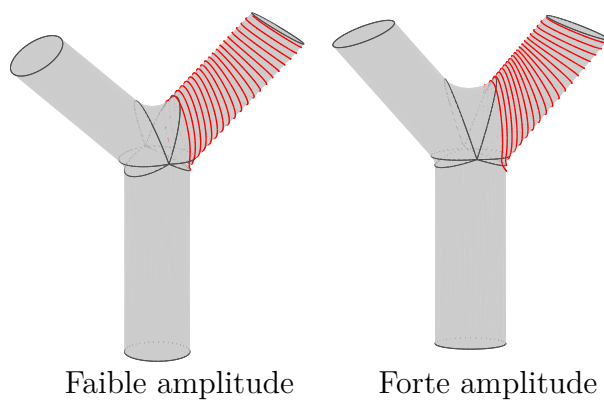


Figure 5.9 Hélice ajustable, variation de l'amplitude

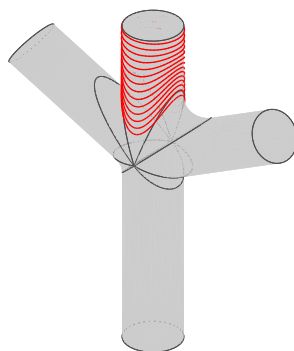


Figure 5.10 Hélice ajustable, $P = 2$

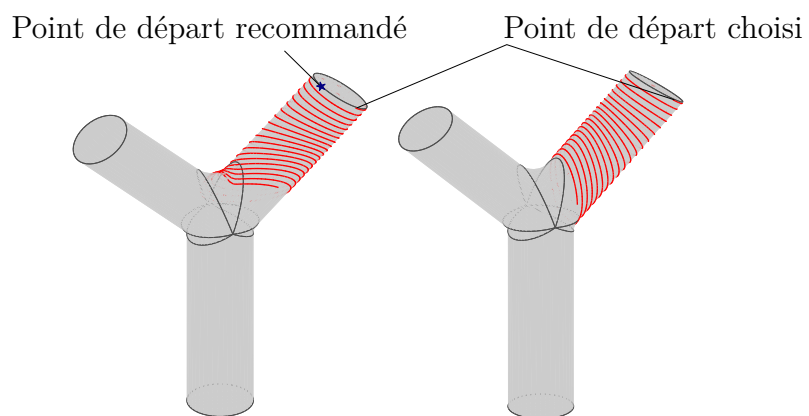


Figure 5.11 Hélice ajustable, variation de la phase (gauche $\phi = 0$, droite $\phi = 1.7$)

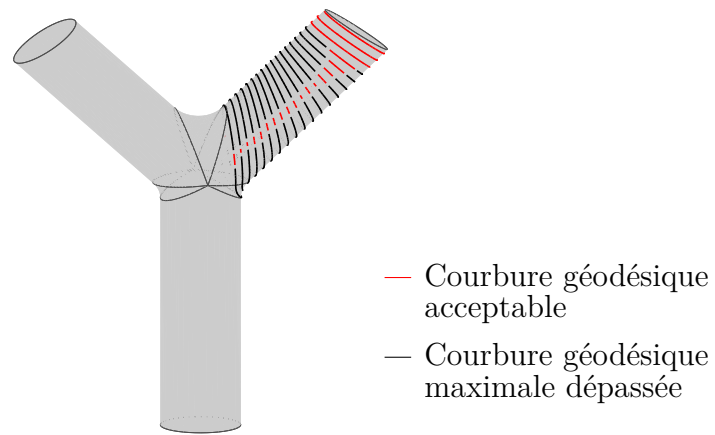


Figure 5.12 Courbure géodésique de la trajectoire

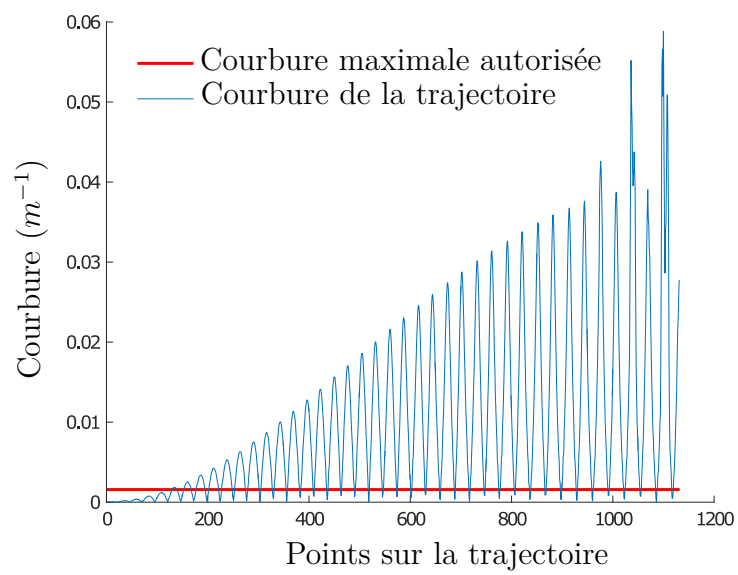


Figure 5.13 Évolution de la courbure géodésique le long de la trajectoire illustrée à la figure 5.12

5.5 Trajectoire sur l'intersection des cylindres

Comme les deux techniques présentées précédemment (méthodes HD et HA) permettent la couverture des branches de manière individuelle, une étape additionnelle permettant de lier ces branches entre elles est nécessaire pour obtenir une pièce en un seul tenant. Pour cela, une trajectoire parcourant la région jonction va être calculée à partir de la courbe formée par l'intersection des différents cylindres formant les branches de la pièce en « Y ». La méthodologie utilisée est de calculer un ensemble de points appartenant à la courbe d'intersection des surfaces cylindriques. Cet ensemble est généré en calculant les courbes d'intersection des cylindres deux à deux. Les points sont ensuite triés afin d'obtenir une trajectoire continue, avant d'être projetés sur la surface réelle du congé formant la région jonction. Cette méthode a été généralisée au cas de pièces formées d'un nombre arbitraire de cylindres de même rayon et dont les axes se coupent en un même point. La technique utilisée pour calculer les points situés sur l'intersection de deux cylindres est détaillée ci-dessous.

Dans un premier temps, un repère local $\mathcal{R}_{ci} = (O_{ci}, \mathbf{x}_{ci}, \mathbf{y}_{ci}, \mathbf{z}_{ci})$ est attribué à chaque cylindre ci . Le vecteur \mathbf{z}_{ci} pointant dans la direction de l'axe du cylindre. On définit également r_i , le rayon de ce cylindre. Les points situés sur l'intersection des cylindres i et j sont ensuite recherchés. Ce calcul étant effectué uniquement entre deux cylindres, nous choisirons ici $i = 1$ et $j = 2$ sans perte de généralité. Un ensemble de n points appartenant à la surface du cylindre 1 sont alors répartis sur un de ses cercles générateurs par :

$$\mathbf{Cercle}_1 = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{y}_1^T \\ \mathbf{z}_1^T \end{bmatrix} = \begin{bmatrix} r_1 & \dots & r_1 \cos(\frac{2p\pi}{n}) & \dots & r_1 \cos(\frac{2(n-1)\pi}{n}) \\ 0 & \dots & r_1 \sin(\frac{2p\pi}{n}) & \dots & r_1 \sin(\frac{2(n-1)\pi}{n}) \\ z_{1_0} & \dots & z_{1_p} & \dots & z_{1_{n-1}} \end{bmatrix}, \quad p = 0, \dots, n-1 \quad (5.4)$$

Cela permet de fixer les coordonnées \mathbf{x} et \mathbf{y} de ces points, exprimées dans le repère local du premier cylindre (\mathcal{R}_{c1}) alors que les coordonnées \mathbf{z} restent à déterminer de manière à ce que les points appartiennent également au cylindre 2. Pour cela, les coordonnées de ces points vont à présent être exprimées dans le repère local du deuxième cylindre. Pour cela, on définit une matrice de transformation homogène \mathbf{T} telle que :

$$\mathbf{T} = {}^o\mathbf{T}_{c2}^{-1} {}^o\mathbf{T}_{c1} \quad (5.5)$$

qui permet de transformer les coordonnées d'un point exprimées dans le repère associé au premier cylindre (\mathcal{R}_{c1}) en leur expression dans le repère du deuxième cylindre (\mathcal{R}_{c2}).

On définit :

$${}^o\mathbf{T}_{ci} = \begin{bmatrix} \mathbf{R}_i & \mathbf{k} \\ \mathbf{0} & 1 \end{bmatrix} \quad (5.6)$$

Avec \mathbf{R}_i la matrice de rotation qui décrit l'orientation du repère local \mathcal{R}_{ci} associé au cylindre ci et \mathbf{k} les coordonnées du point d'intersection des axes des deux cylindres (définissant le centre de ce repère). Ces deux objets sont exprimés dans le repère global du problème, \mathcal{R}_0 . En définissant $\mathbf{P}_1 = [\mathbf{x}_1 \ \mathbf{y}_1 \ z_1 \ \mathbf{1}]^T$, où $\mathbf{1}$ est un vecteur de mêmes dimensions que \mathbf{x}_1 et dont tous les éléments sont égaux à 1, il est alors possible d'exprimer ces points dans \mathcal{R}_{c2} :

$$\mathbf{P}_2 = \mathbf{T}\mathbf{P}_1 = [\mathbf{x}_2 \ \mathbf{y}_2 \ z_2 \ \mathbf{1}]^T \quad (5.7)$$

Ceux-ci doivent alors vérifier l'équation (5.8) pour assurer qu'ils appartiennent bien également au deuxième cylindre :

$$\mathbf{diag}(\mathbf{x}_2)\mathbf{x}_2 + \mathbf{diag}(\mathbf{y}_2)\mathbf{y}_2 - \mathbf{diag}(\mathbf{r}_2)\mathbf{r}_2 = \mathbf{0} \quad (5.8)$$

Avec $\mathbf{diag}(\mathbf{x})$ la matrice diagonale dont les éléments sont les composantes de \mathbf{x} et \mathbf{r}_2 un vecteur contenant n fois la valeur du rayon du deuxième cylindre.

En substituant \mathbf{x}_2 et \mathbf{y}_2 par leur expression en fonction des coordonnées dans \mathcal{R}_{c1} , on obtient une fonction quadratique dont les racines sont les coordonnées z_1 désirées :

$$a \mathbf{diag}(z_1)z_1 + \mathbf{diag}(\mathbf{b}(\mathbf{x}_1, \mathbf{y}_1))z_1 + \mathbf{c}(\mathbf{x}_1, \mathbf{y}_1) = \mathbf{0} \quad (5.9)$$

Avec :

$$a = T_{13}^2 + T_{23}^2$$

$$\begin{aligned} \mathbf{b} = & 2T_{11}T_{13}\mathbf{x}_1 + 2T_{12}T_{13}\mathbf{y}_1 + 2T_{13}T_{14} \\ & + 2T_{21}T_{23}\mathbf{x}_1 + 2T_{22}T_{23}\mathbf{y}_1 + 2T_{23}T_{24} \end{aligned}$$

$$\begin{aligned} \mathbf{c} = & -\mathbf{diag}(\mathbf{r}_2)\mathbf{r}_2 + T_{11}^2\mathbf{diag}(\mathbf{x}_1)\mathbf{x}_1 + 2T_{11}T_{12}\mathbf{diag}(\mathbf{x}_1)\mathbf{y}_1 \\ & + 2T_{11}T_{14}\mathbf{x}_1 + T_{12}^2\mathbf{diag}(\mathbf{y}_1)\mathbf{y}_1 + 2T_{12}T_{14}\mathbf{y}_1 + T_{14}^2 \\ & + T_{21}^2\mathbf{diag}(\mathbf{x}_1)\mathbf{x}_1 + 2T_{21}T_{22}\mathbf{diag}(\mathbf{x}_1)\mathbf{y}_1 + 2T_{21}T_{24}\mathbf{x}_1 \\ & + T_{22}^2\mathbf{diag}(\mathbf{y}_1)\mathbf{y}_1 + 2T_{22}T_{24}\mathbf{y}_1 + T_{24}^2 \end{aligned}$$

et T_{ij} l'élément situé à la $i^{\text{ème}}$ ligne et $j^{\text{ème}}$ colonne de \mathbf{T} .

Pour chaque point initial, deux solutions existent pour \mathbf{z}_1 , correspondant aux deux côtés de l'intersection (figure 5.14). Un tri est donc effectué pour supprimer les points situés sur le mauvais côté de l'intersection ainsi que ceux qui se trouvent à « l'intérieur » d'une des autres branches.

Une fois que tous les points appartenant aux portions d'intersection désirées sont calculés, ils sont projetés sur la surface de la région jonction. Le procédé est répété pour chaque paire de cylindres et les points obtenus sont finalement triés pour produire une trajectoire d'outil qui minimise le nombre de coupes à opérer dans les fibres placées. La trajectoire est également lissée, de manière à éviter les angles vifs (figures 5.15a et 5.15b).

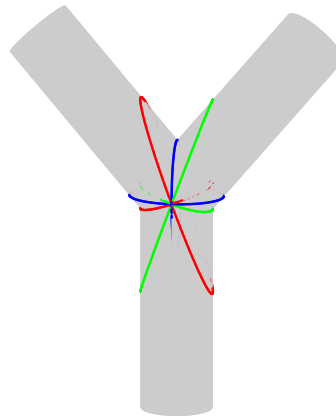


Figure 5.14 Points situés sur les intersections mathématiques des cylindres

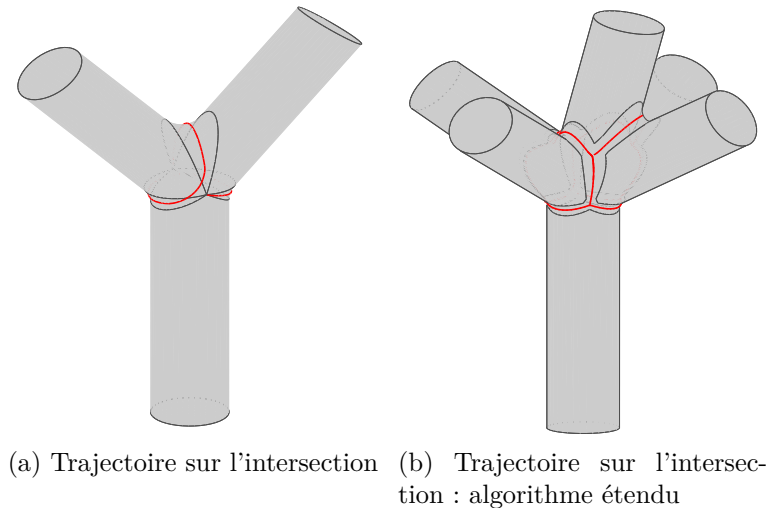


Figure 5.15 Trajectoires dans la région jonction

5.6 Décalage de la courbe de référence

Après un premier travail qui a consisté à couvrir individuellement chaque branche du « Y », nos recherches se sont concentrés sur les moyens permettant de couvrir deux branches à l'aide d'une trajectoire continue non coupée. Ces travaux sont présentés au chapitre 4, et quelques précisions vont à présent y être apportées.

Comme il a été mentionné dans l'article, la définition des différentes couches grâce à la méthode *CPA* nécessite de décaler la courbe de référence pour obtenir des trajectoires parallèles. Ce décalage doit s'effectuer dans une direction perpendiculaire à la courbe guide et appartenant au plan tangent à la surface. La distance géodésique entre les deux courbes (c'est-à-dire la distance la plus courte, mesurée sur la surface) doit être égale à la largeur de la bande déposée afin d'éviter les manques et recouvrements. Plusieurs méthodes pour générer de telles trajectoires parallèles existent, dont certaines ont été présentées en section 2. Dans ce travail une méthode simple inspirée de Khakpour et al. (2015) a été implémentée pour translater la courbe de référence.

Les trajectoires sont représentées par une succession de poses définissant les positions des points de la trajectoire ainsi que les vecteurs tangents \mathbf{o} et normaux \mathbf{a} (dans le plan normal à la surface) et \mathbf{n} (dans le plan tangent à la surface) à la trajectoire (voir figure 4.3). Le vecteur \mathbf{n} permet donc de définir la direction dans laquelle le point appartenant à la nouvelle courbe doit être recherché. Il est ensuite nécessaire d'assurer que la distance géodésique de décalage soit bien celle désirée. La surface de la pièce étant représentée par des facettes

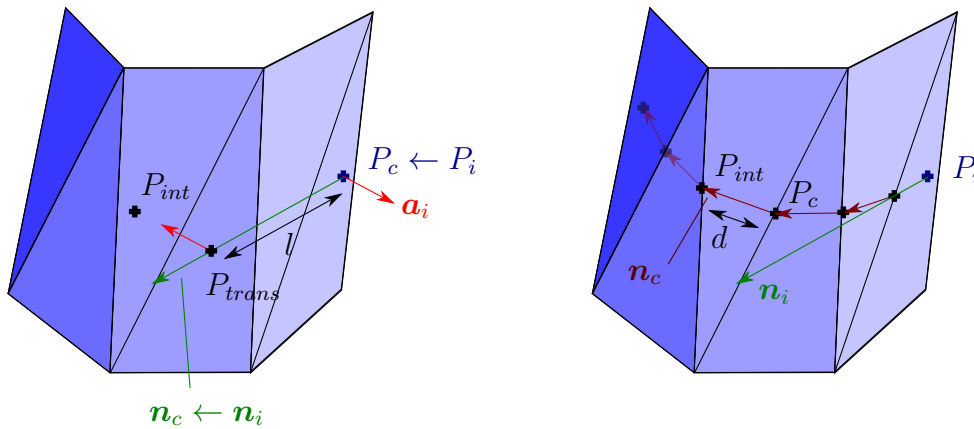
triangulaires, cette distance peut donc être mesurée grâce à une succession de segments de droites appartenant à des faces voisines. L'algorithme de calcul des points de la nouvelle trajectoire est alors le suivant :

Pour $i = 1$ à n , nombre de points de la courbe de référence, faire :

1. P_i et son vecteur normal associé \mathbf{n}_i , appartenant à la courbe de référence, sont définis comme objets courant P_c et \mathbf{n}_c . l est défini comme la distance géodésique à parcourir.
2. P_c est translaté de la distance l le long de la direction \mathbf{n}_c (P_{trans}) puis projeté sur la surface de la pièce selon \mathbf{a}_i . P_{int} est obtenu.
3. Si P_{int} appartient à la même facette que P_c alors la distance entre les deux est bien l , $i = i + 1$ et retour à l'étape 1. Sinon P_{int} est recalculé comme étant l'intersection de \mathbf{n}_c et de l'arête de la face contenant P_c . d est défini comme étant la distance entre P_c et P_{int} .
4. $l = l - d$.
5. \mathbf{n}_c est projeté sur la nouvelle face.
6. $P_c = P_{int}$ et retour à l'étape 2.

Une illustration de cet algorithme peut être trouvée à la figure 5.16. Comme expliqué par Khakpour et al. (2015), cette stratégie peut conduire à l'apparition de défauts sur la courbe décalée. Il est cependant possible de les éliminer pour obtenir une courbe lisse. Plusieurs sous-algorithmes ont donc été implémentés dans ce travail pour réaliser ce lissage. Cependant, dans notre cas, les courbes de référence utilisées étant déjà lisses il est peu probable que ces algorithmes soient utiles. Le lecteur se référera donc à la référence sus-citée pour plus de détails sur ces techniques de lissage.

Pour finir, un dernier détail doit être réglé pour obtenir une trajectoire complète. En effet, comme l'illustre la figure 5.17, les extrémités de la courbe décalée, ont peu de chance de se trouver sur les frontières de la surface à couvrir. Il est donc nécessaire d'extrapoler la nouvelle courbe pour générer les points manquants. Pour cela, l'angle entre la direction tangente aux extrémités de la courbe et la direction de référence (axe du cylindre dans le cas des pièces en « Y ») est calculé. L'algorithme itératif présenté en section 4.4.1 est ensuite utilisé pour calculer les points manquants de la trajectoire tout en assurant que celle-ci se termine sur une arête marquant la frontière de la surface à couvrir.



Etape 2

Etape 3 : recalcul de P_{int} , 4^{ieme} itération

Figure 5.16 Décalage d'un point de la courbe de référence

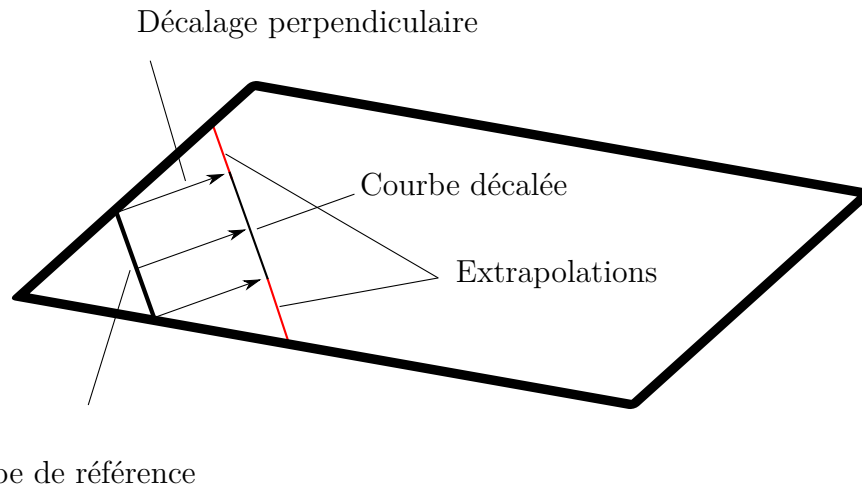


Figure 5.17 Extrapolation des extrémités de la courbe pour atteindre les frontières de la région de travail

5.7 Calcul de la courbure géodésique

Le concept de courbure géodésique a été abordé tout au long de ce mémoire comme un aspect important dans la génération de trajectoire pour *AFP*. La définition de cette propriété, ainsi qu'une méthode pour mesurer sa valeur dans le cas d'une courbe discrète tracée sur une surface représentée par un maillage triangulé, sont présentées dans ce paragraphe.

On désigne par le terme *géodésique*, la généralisation d'une ligne droite pour une courbe qui serait tracée sur une surface courbée quelconque. La courbure géodésique est alors une mesure de la différence entre une courbe quelconque et une géodésique de la surface en un

point donné. Cela peut également être vu localement comme étant l'inverse du rayon du cercle osculateur de la courbe, qui appartient au plan tangent à la surface au point considéré. Si les méthodes pour calculer la courbure géodésique d'une courbe paramétrée tracée sur une surface ayant également une représentation paramétrique sont bien connues (Abbena et al. (2006)), son calcul dans le cas d'une courbe discrète tracée sur un maillage triangulé n'est pas aussi bien documenté. Une approximation est ici proposée pour calculer cette propriété en chaque point de la trajectoire générée. L'hypothèse faite est que les points précédant et suivant le point de calcul sont suffisamment proches pour être considérés comme appartenant au plan tangent à la surface au point de calcul. En pratique pour s'en assurer, on projette ces points sur ce plan tangent. Le cercle passant par ces trois points (et donc appartenant au plan tangent à la surface au point de calcul) est alors calculé, et son rayon est pris comme étant l'inverse de la courbure géodésique de la trajectoire (figure 5.18). La validité de cette technique a pu être vérifiée en calculant la courbure géodésique de courbes connues. Elle a été utilisée dans l'implémentation de l'algorithme *MCC*, présenté en section 4.4.2.

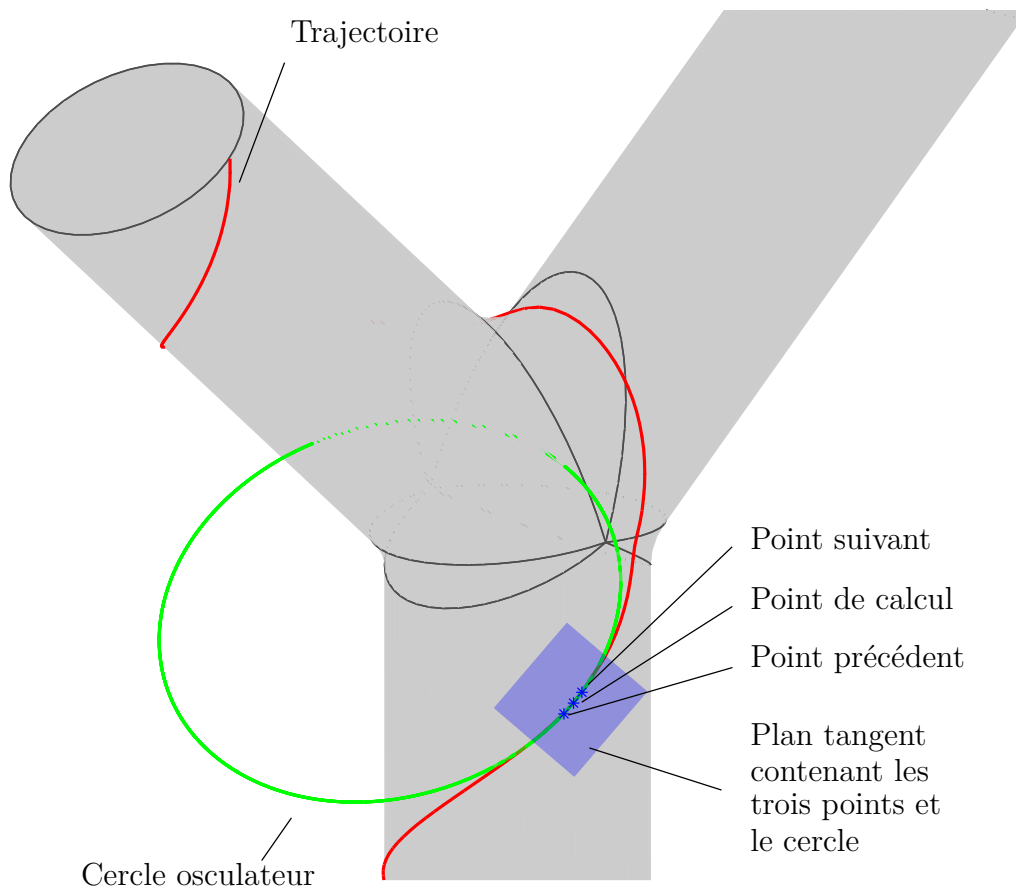


Figure 5.18 Calcul de la courbure géodésique en un point d'une trajectoire sur une pièce en « Y »

5.8 Détection des collisions

Afin d'identifier les trajectoires qui pourraient poser des problèmes et ne pas être réalisables en pratique, une investigation préliminaire a été réalisée dans cette étude au niveau de la détection de collisions. Celle-ci se concentre uniquement sur les collisions outil-moule qui sont, comme abordé dans l'article, impossible à corriger si l'on souhaite garder une orientation d'outil correspondante aux spécificités de l'*AFP* (pression du rouleau de compaction normale à la surface du moule). Un outil fictif a été défini (figure 5.19) pour être utilisé dans les tests de collisions. Celui-ci se présente sous forme d'un stylo au bout duquel un cylindre représente le rouleau de compaction qui vient plaquer le pré-imprégné sur le moule dans une tête de placement réelle. Ce type de forme a été choisi car il est évident que pour fabriquer des pièces telles que des tubes en « Y », il sera nécessaire de développer de nouvelles têtes de placement se terminant de cette manière pour pouvoir accéder au moule, particulièrement au niveau de la jonction entre les branches.

Une fonctionnalité a donc été ajoutée au logiciel de planification de trajectoire, permettant d'importer le fichier CAO de l'outil, sous format STL et de lui faire parcourir la trajectoire générée afin de vérifier si certaines zones de cette trajectoire conduisent à des collisions. Pour cela l'algorithme développé par Liu et al. (2010) et fourni sous forme de fonction MATLAB a été utilisé. Celui-ci permet de déterminer efficacement et rapidement si un point est à l'intérieur ou à l'extérieur d'un polyèdre. On cherche donc ici à savoir si, pour chaque pose de l'outil sur la trajectoire, il existe des points de sa surface se situant à l'intérieur du polyèdre définissant le moule (pièce en « Y »). Pour assurer que toutes les collisions soient bien détectées, il est également vérifié s'il existe des points de la surface du moule situés dans le polyèdre définissant l'outil. Chaque point de la trajectoire présentant une collision est alors enregistré et un rapport décrivant les zones de la trajectoire où se situent ces collisions est envoyé à l'utilisateur à la fin de l'analyse.

Bien qu'en *AFP* la force de compaction doit toujours agir dans une direction normale à la surface du moule afin d'assurer une bonne dépose des rubans de fibres, une méthode d'évitement des collisions outil-moule a tout de même été développée. On fait ici l'hypothèse de l'existence d'un système mécanique permettant d'incliner, dans une certaine mesure, le corps de l'outil par rapport au rouleau de compaction. L'angle maximal d'inclinaison de l'outil a été fixé arbitrairement à 20° pour les tests effectués. Le principe de cette méthode (illustré figure 5.20) est alors d'incliner l'outil, autour du point de contact (\mathbf{p}) entre le rouleau et le moule, dans une direction normale à la surface du moule au niveau de la région où est détectée la collision. Pour cela, les points de la surface du moule situés à l'intérieur de l'outil lors de la collision sont récupérés, ainsi que leurs vecteurs normaux. Le barycentre de la collision

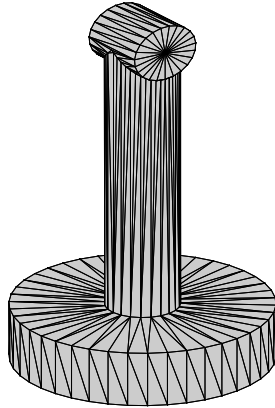


Figure 5.19 Outil utilisé pour tester l'algorithme de détection de collisions

(\mathbf{c}) et la moyenne de ces vecteurs normaux (\mathbf{n}_{moy}) sont ensuite calculés. Ces données sont utilisées pour définir l'axe de rotation $\mathbf{r} = \mathbf{n}_{moy} \times \mathbf{h}$ autour duquel l'outil doit être incliné (avec $\mathbf{h} = \mathbf{c} - \mathbf{p}$). L'inclinaison est ensuite réalisée par itérations successives en effectuant une rotation d'un angle faible (typiquement autour de 1°) autour de l'axe de rotation \mathbf{r} , jusqu'à ce que l'angle total d'inclinaison ψ permette d'éviter la collision ou soit supérieur à l'angle limite d'inclinaison de l'outil par rapport à la pose initiale (figure 5.21). Cette méthode se révèle efficace pour éviter les collisions dans le cas de l'outil défini plus haut. Cependant, elle reste fortement dépendante de la géométrie du moule à couvrir et de l'angle d'inclinaison maximal autorisé pour l'outil.

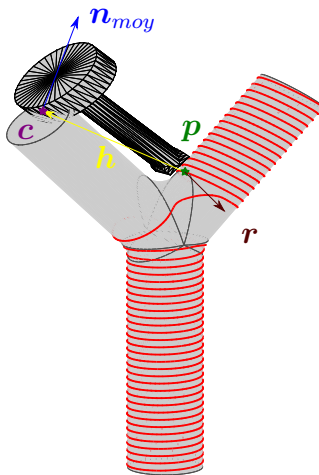


Figure 5.20 Méthode d'inclinaison de l'outil

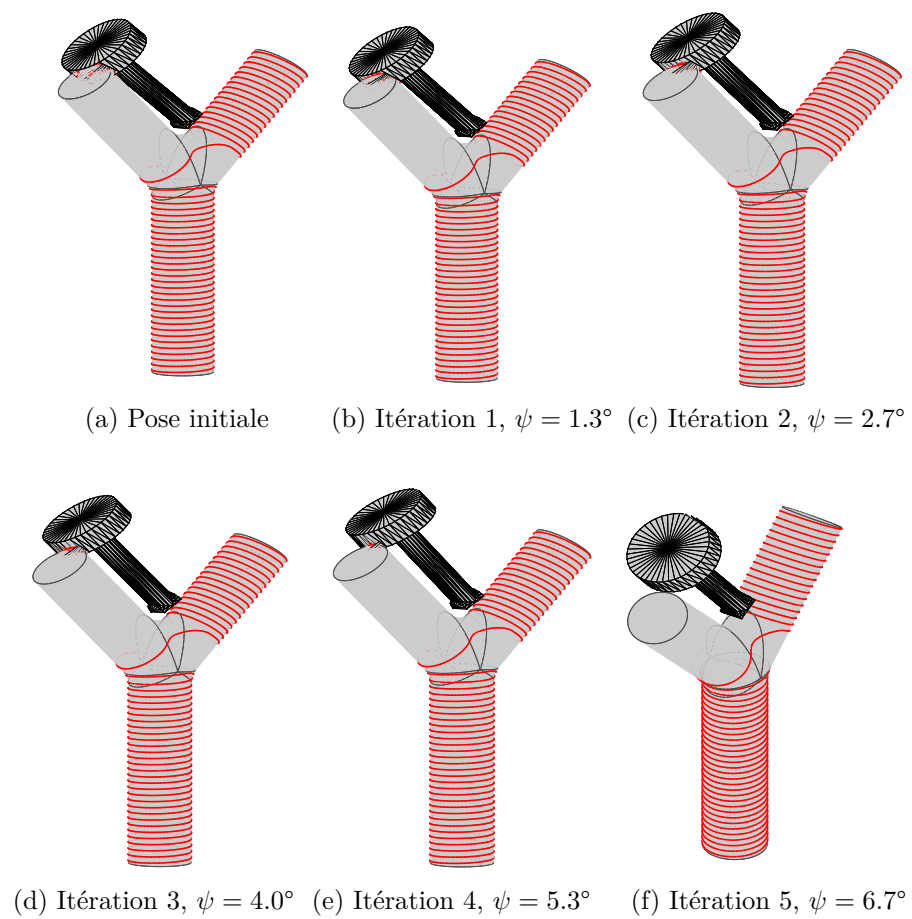


Figure 5.21 Processus itératif d'inclinaison de l'outil

5.9 Simulations

Pour prouver la faisabilité des trajectoires générées à l'aide des algorithmes présentés dans ce mémoire, des simulations ont été effectuées. Le problème principal à résoudre pour réaliser ces simulations a été de transformer la trajectoire cartésienne en deux trajectoires définies dans les espaces articulaires des deux robots (manipulateur FANUC et table rotative). Les bases mathématiques permettant cette séparation ont été présentées dans l'article en section 4.6. Afin de mener à bien cette séparation, un algorithme inspiré du travail de Huo et Baron (2008) a été utilisé. Cet algorithme permet de faire coïncider l'extrémité de l'outil que porte le manipulateur avec la pose qu'il doit atteindre dans la trajectoire définie sur le moule, lui-même fixé sur la table rotative. Il est appliqué pour chaque pose de la trajectoire, en prenant comme posture initiale θ_{pre} qui correspond à la configuration atteinte pour la pose précédente de la trajectoire. Il donne comme résultat le vecteur θ à 7 composantes, contenant les valeurs angulaires des articulations du manipulateur (θ_1 à θ_6) et de la table rotative θ_7 . Schématiquement cet algorithme est :

```

Data:  $\theta_{pre}$ 
Result:  $\theta$ 
Initialisation;
 $P_{courant} = \text{CindirManip}(\theta(1:6));$ 
 $P_{objective} = \text{CindirTable}(\theta(7));$ 
 $\Delta R = R_{courant}^T R_{objective};$ 
 $\Delta p = p_{objective} - p_{courant};$ 
 $e_{position} = \Delta p^T \Delta p;$ 
 $e_{orientation} = \text{Erreur}(\Delta R);$ 
while  $e_{pos} > \epsilon$  or  $e_{orient} > \epsilon$  do
     $\Delta t = \begin{bmatrix} R_{courant} \text{vect}(\Delta R) \\ \Delta p \end{bmatrix};$ 
     $\Delta \theta = J^\dagger \Delta t + (I_7 - J^\dagger J)h;$ 
     $\theta = \theta_{pre} + \eta \Delta \theta;$ 
     $P_{courant} = \text{CindirManip}(\theta(1:6));$ 
     $P_{objective} = \text{CindirTable}(\theta(7));$ 
     $\Delta R = R_{courant}^T R_{obj};$ 
     $\Delta p = p_{objective} - p_{courant};$ 
     $e_{position} = \Delta p^T \Delta p;$ 
     $e_{orientation} = \text{Erreur}(\Delta R);$ 
     $\theta_{pre} = \theta ;$ 
end

```

Dans cet algorithme, les matrices $P = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$ sont des matrices de transformation homo-

gènes qui décrivent des poses. Les deux fonctions `CindirManip` and `CindirTable` calculent respectivement la cinématique directe du manipulateur et de la table rotative. Elles donnent en sortie respectivement, la pose de l'effecteur du manipulateur et la pose associée au point de la trajectoire que doit atteindre l'outil, après rotation de la table rotative. Ces deux éléments sont exprimés dans le repère global du système. La fonction `Erreur` donne quant à elle l'angle de rotation permettant de passer de l'orientation $\mathbf{R}_{courant}$ à l'orientation \mathbf{R}_{obj} . Ces deux matrices représentant respectivement les orientations de l'outil du manipulateur et de la pose à atteindre sur la trajectoire, après rotation de la table rotative. Un facteur d'amortissement η est utilisé pour éviter les instabilités numériques. Celles-ci peuvent apparaître lorsque les déplacements à effectuer sont trop importants. En effet, l'hypothèse de petits déplacements effectuée pour calculer les nouvelles valeurs articulaires ne serait alors plus respectée dans ce cas.

Comme expliqué dans l'article, chaque pose est optimisée de manière à éviter les limites articulaires et singularités de la cellule de travail et minimiser les déplacements du manipulateur. Toutefois, cette optimisation est locale (effectuée pour chaque pose) et la forme de la trajectoire dans l'espace articulaire des robots va donc en général dépendre de la configuration de départ. Il serait intéressant dans un futur travail, d'effectuer un calcul d'optimisation global de manière à obtenir une trajectoire optimisée pour une trajectoire complète.

Un exemple de séparation de trajectoire est illustré à la figure 5.22. Celui-ci montre l'évolution des valeurs articulaires le long des trajectoires à 45° générées par l'algorithme *CPA* (figure 4.6). On peut y observer deux phases pour la trajectoire : la première correspond à l'hélice sur la branche de base (verticale) et la deuxième à l'hélice sur la branche supérieure. On constate que le comportement désiré (minimisation des déplacements du manipulateur, évitement des singularités et des limites articulaires) est bien réalisé. En effet, dans la première phase c'est la table rotative (θ_7) qui effectue les déplacements les plus importants, alors que le manipulateur (θ_1 à θ_6) effectue une translation verticale de l'outil. Dans la deuxième phase on observe plus de variations au niveau des articulations du manipulateur. En effet celui-ci doit à présent tourner autour de la branche supérieure pour y enrouler les rubans de pré-imprégnés. Une zone proche d'une singularité apparaît également aux alentours de la centième pose de chaque trajectoire. Cela correspond au moment où l'outil se situe dans la région de la jonction. En effet, comme le montre la figure 5.23, l'angle de la cinquième articulation du manipulateur peut devenir proche de 0° dans cette zone. Les axes des articulations quatre et six sont alors presque alignés, conduisant à une perte de mobilité locale, donc une singularité. On peut alors observer que dans cette zone, la trajectoire de table rotative est modifiée pour éviter cette singularité autant que possible.

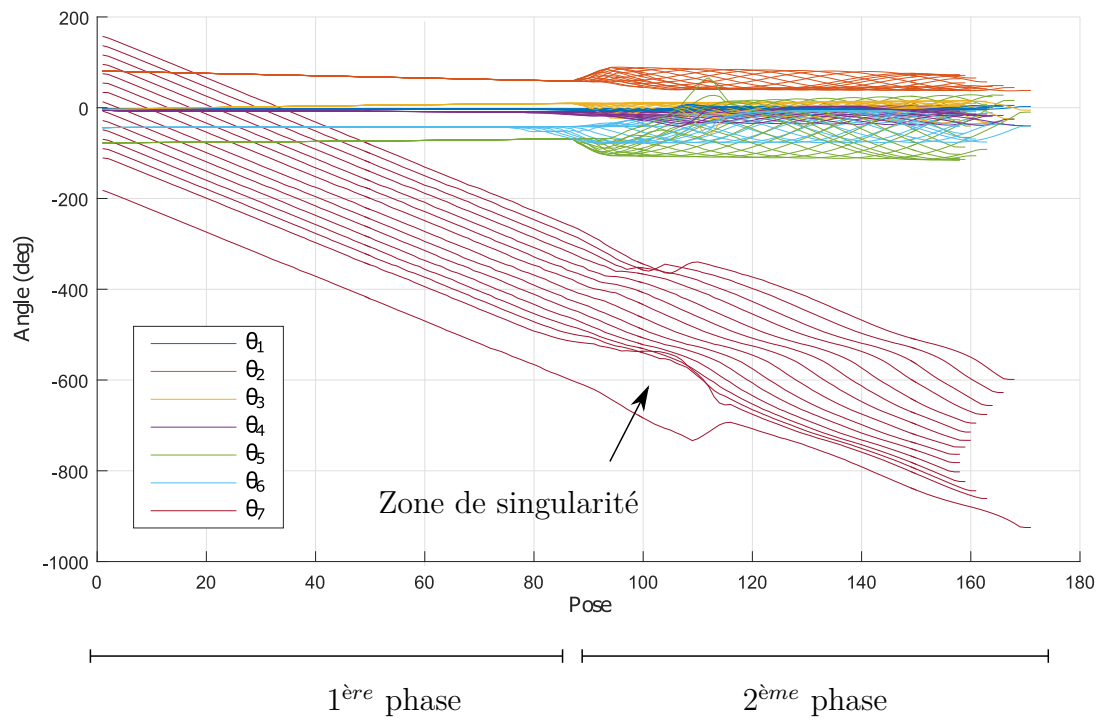


Figure 5.22 Évolution des valeurs articulaires le long des 15 trajectoires permettant la définition d'une couche orientée à 45°

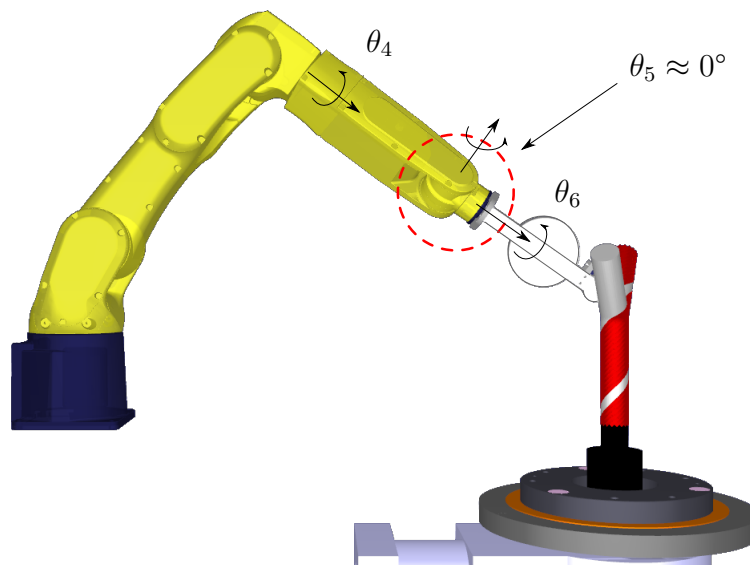


Figure 5.23 Singularité du poignet du manipulateur

Après avoir été testées en simulations, certaines trajectoires obtenues pour le manipulateur ont été expérimentées sur des robots réels. Deux manipulateurs FANUC différents ont été utilisés pour réaliser ces tests, un M20iA (figure 5.24) et un LR Mate 200iD/7L. Pour mener à bien ces expériences, les trajectoires obtenues sous forme de listes de points contenant les valeurs articulaires du manipulateur ont été converties en programme .LS. Ce type de programmes peut ensuite être compilé grâce au logiciel ROBOGUIDE en fichier binaire .TP. Ceux-ci sont alors directement utilisables par le contrôleur FANUC. Ces expériences ont permis de mettre en lumière les travaux encore à réaliser, notamment pour réussir à synchroniser le manipulateur et la table rotative. En effet, la commande du manipulateur dans l'espace articulaire à l'aide de fichiers .LS offre peu de libertés en ce qui concerne la maîtrise de la vitesse de déplacement de l'outil. Ce travail sortant du cadre de cette recherche, les solutions permettant de réaliser cette synchronisation seront investiguées dans la suite du projet.

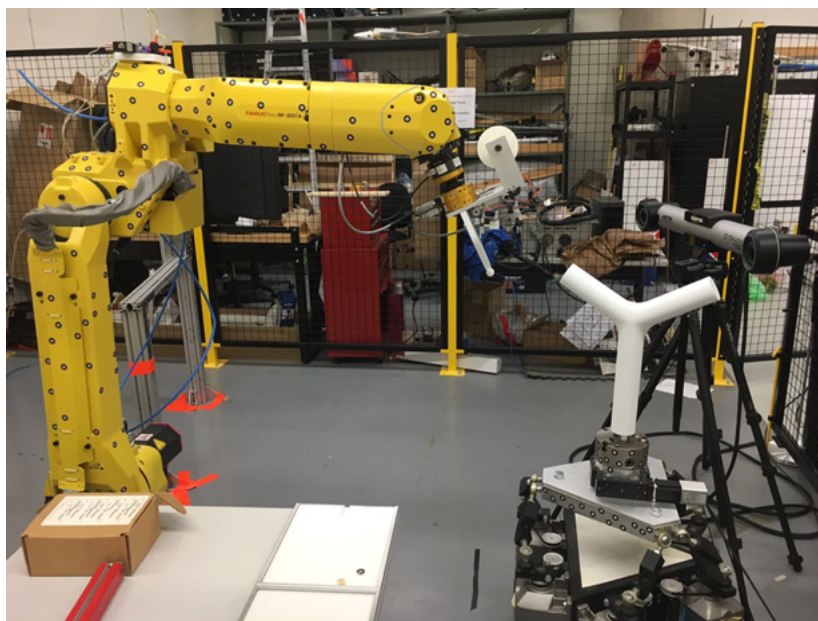


Figure 5.24 Cellule de travail comprenant le manipulateur FANUC M20iA (Université Concordia)

5.10 Description générale du programme de génération de trajectoires

Au cours de ce projet de recherche, une application regroupant les différents algorithmes présentés dans ce mémoire a été développée à l'aide du logiciel MATLAB, sous la forme d'une interface graphique. Celle-ci permet de générer des trajectoires aussi bien sur des pièces de formes quelconques que sur des tubes en « Y ». Toutefois, la plus grande part du travail s'est concentrée sur cette dernière option. Dans tous les cas, les premières opérations à effectuer

sont l'import d'un fichier de format .STL et la segmentation de la pièce en régions lisses. Dans le cas des formes quelconques, l'utilisateur pourra ensuite sélectionner les surfaces de travail (régions sur lesquelles les trajectoires seront générées), le point de départ de la trajectoire, ainsi que l'orientation générale servant de référence à l'algorithme itératif de calcul de la trajectoire. En ce qui concerne les pièces en « Y », les quatre différentes méthodes présentées dans ce mémoire ont été autant que possible automatisées. Cette interface permet également à l'utilisateur de sauvegarder les trajectoires générées sous la forme d'une structure MATLAB. Celle-ci contient aussi les informations géométriques de la pièce traitée afin d'éviter de répéter tout le processus d'import et de segmentation lorsque l'utilisateur souhaite retravailler sur une même pièce. Des détails supplémentaires sur les fonctionnalités et l'utilisation de ce logiciel sont donnés dans le manuel utilisateur présenté en annexe A.

CHAPITRE 6 DISCUSSION GÉNÉRALE

Les algorithmes et méthodes abordés dans ce mémoire présentent certaines limitations et améliorations possibles qui seront discutées dans ce chapitre. Avant de traiter plus spécifiquement ces questions, on peut citer une limitation générale du programme de génération de trajectoires développé dans cette étude qui est le temps de calcul. En effet, ce logiciel ayant été écrit à l'aide du langage MATLAB et peu de travail ayant été réalisé sur l'optimisation du code, certaines opérations peuvent rapidement se révéler longues à effectuer si le volume de données à traiter est important. Il est par exemple peu pratique de travailler avec des pièces dont le maillage a un haut niveau de raffinement et présente un grand nombre d'éléments. Cependant cette limitation pourrait facilement être réglée en retravaillant le code et en le réécrivant dans un langage plus rapide tel que le C++.

6.1 Limitation de l'utilisation des directions de courbures principale

L'utilisation des directions de courbures principales comme direction de référence pour la génération de trajectoires grâce à l'algorithme itératif présenté au chapitre 4.4.1 fonctionne bien pour des pièces dont la surface est suffisamment régulière (comme par exemple des cylindres courbés). Cependant, si la surface présente de nombreuses variations de courbure, ces directions vont varier de manière importante et on n'obtiendra plus une orientation particulière pour définir l'angle de placement de la fibre, ce qui peut conduire à des trajectoires chaotiques (voir figure 6.1). Cette technique n'est donc pas adaptée à tous les types de pièces, c'est pourquoi il a été implémenté dans le logiciel, pour la génération de trajectoire sur des pièces quelconques, la possibilité pour l'utilisateur de choisir une direction de référence ($\mathbf{x}, \mathbf{y}, \mathbf{z}$ ou un vecteur de son choix) qui est ensuite projetée sur la surface de la pièce afin de pouvoir être utilisée dans l'algorithme itératif.

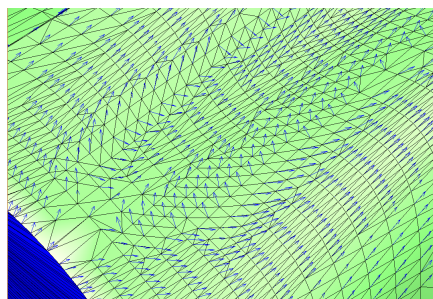


Figure 6.1 Directions principales de courbure non adaptées pour l'algorithme itératif

6.2 Trajectoire sur l'intersection des cylindres

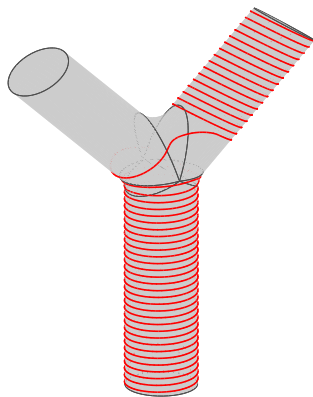
La méthode de génération de trajectoires sur l'intersection des cylindres a été développée pour combler les manques de matière apparaissant dans la région jonction lors de l'utilisation des algorithmes HD et HA. Cependant comme ces deux algorithmes se sont révélés conduire à de nombreux problèmes et que la trajectoire sur l'intersection présentait de fortes courbures géodésiques qui ne peuvent pas être réalisées en *AFP*, la définition des trajectoires sur la région jonction n'a pas été poussée à son terme et il a été décidé de consacrer les efforts aux méthodes *CPA* et *MCC*. On peut en effet voir que la génération d'une seule trajectoire sur l'intersection ne conduit en général pas à la couverture totale de la région jonction. Pour assurer cette couverture dans le cas des pièces en forme de « Y » et, afin que les branches soient bien liées entre elles, il avait été envisagé de décaler cette trajectoire, par exemple en lui faisant subir une rotation autour d'un axe passant par le point d'intersection des cylindres et normal au plan de symétrie de la pièce qui contient les axes des trois branches. Cette méthode pourrait se révéler intéressante pour d'autres applications qui ne présentent pas de contraintes sur la courbure de la trajectoire, comme par exemple des applications de contrôle non destructif.

6.3 Algorithme *CPA*

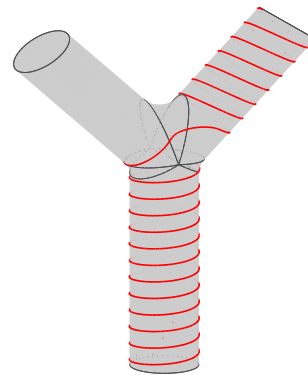
Si on a vu que parmi les algorithmes de génération de trajectoires proposés dans ce mémoire, celui de l'angle de placement constant était celui qui conduisait à la couverture la plus uniforme, il possède tout de même certains défauts qu'il est important de souligner. En premier lieu, il est possible de montrer que certaines situations conduiront à l'apparition de *gaps* et *overlaps* dans une couche. En effet, prenons exemple de la couche placée à l'angle 45° , si le périmètre du cercle de base du cylindre sur lequel est placée la fibre n'est pas un multiple de la largeur de bande déposée, un manque ou un recouvrement sera alors observé au niveau des courses qui viennent « fermer » la couche sur la surface du cylindre (figure 4.6). Ce problème peut cependant être corrigé assez facilement en utilisant la capacité des têtes de placement à délivrer le nombre voulu de brins de composite dans une course. Il est alors possible de faire varier sa largeur pour l'adapter au mieux à la situation.

Le second problème de cette méthode est que la taille de la région jonction, et par conséquent l'angle formé par les branches du « Y » va également influencer sur la qualité des trajectoires produites. Il a en effet été observé que la trajectoire à 90° a tendance à conduire à des manques de fibre aux alentours de la région jonction (voir figure 6.2). En effet, le problème qui a conduit à la définition des deux algorithmes HD et HA se retrouve ici puisque la

trajectoire hélicoïdale ne pourra pas s'enrouler de manière satisfaisante autour des branches si les congés de la région jonction sont trop importants. Contrairement aux couches à 0° et 45° où il est possible de limiter l'impact qu'aura la traversée de cette région sur la forme de la trajectoire en choisissant avec précaution la localisation de cette traversée, il est très difficile d'obtenir un résultat similaire avec l'angle de 90° . Il est donc recommandé, au vu des manques qui peuvent apparaître, d'utiliser la couche à 90° en complément des couches à 0° ou 45° qui, elles, permettent une couverture uniforme.



(a) Largeur de bande de 6.35 mm



(b) Largeur de bande de 15.875 mm

Figure 6.2 Algorithme *CPA*, manque de fibre aux alentours de la région jonction pour la trajectoire à 90°

Pour finir, un des principaux problème de cette méthode, qui a déjà été discuté au chapitre 4, réside dans le fait que cette stratégie produit des trajectoires présentant en général de fortes courbures géodésiques qui peuvent conduire au flambement des fibres lors de la dépose. Dans l'état actuel de l'art, ce problème est difficilement résoluble et, si le placement peut éventuellement être effectué, les pièces produites auront de fortes chances de présenter des faiblesses mécaniques. On peut envisager que des solutions à ce problème soient trouvées dans le futur grâce à l'utilisation de nouveaux matériaux ou de nouvelles technologies de placement. Une des technologie récemment proposée et qui semble prometteuse est celle du « cisaillement continu des brins » (*continuous tow shearing*) proposée par Kim et al. (2012). Cette technique consiste, lors de la dépose sur une trajectoire courbe, non pas de courber les fibres, mais de les cisailer en maintenant l'orientation de la tête de placement constante (figure 6.3). D'après les auteurs, il est possible de réaliser des trajectoires ayant un rayon de courbure pouvant descendre jusqu'à 50 mm tout en garantissant une absence de flambement des fibres. Cette technique laisse donc penser qu'il est réaliste d'envisager de produire des pièces grâce à la méthode *CPA* dans les années à venir. Si les technologies de

placement évoluent dans cette direction et permettent l'utilisation de l'algorithme *CPA*, on peut également envisager de combiner les trajectoires obtenues par cet algorithme avec celles obtenues grâce aux autres méthodes définies dans ce mémoire (*HD*, *HA*, *MCC*). La mise au point d'un modèle d'analyse par éléments finis plus poussé pourrait également permettre de choisir ces combinaisons de manière à apporter plus de résistance mécanique dans les zones les plus sollicitées de la pièce.

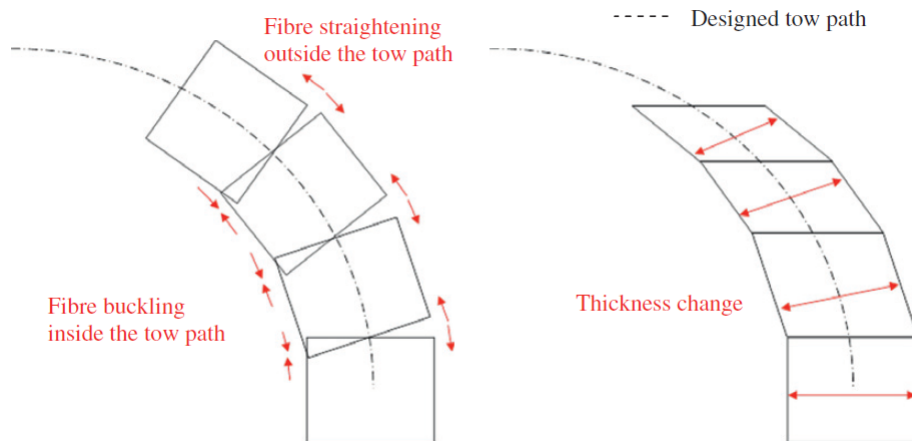


Figure 6.3 Méthode « *Continuous tow shearing* » tiré de Kim et al. (2012)

6.4 Algorithme *MCC*

La méthode de contrainte de courbure maximale (*MCC*) présentée au chapitre 4 permet de générer des trajectoires qui sont réalisables en pratique par les machines d'*AFP* actuelles. Cependant, comme souligné dans l'article, cette méthode ne permet pas d'obtenir une couverture uniforme de la pièce. Bien que l'on puisse assurer la couverture complète des branches grâce aux hélices par lesquelles commencent chaque trajectoire, on observe en général un manque de fibre dans la région de la jonction qui devra éventuellement être comblé à la main par la suite en appliquant des renforts de pré-imprégnés. Cela entraîne en plus un risque de faiblesse mécanique au niveau de cette région qui est celle où, d'après l'analyse par éléments finis, on retrouve les déformations les plus importantes pour les deux cas de chargement étudiés. De plus, les trajectoires terminant en ligne droite ne couvrent qu'une faible partie de leur branche d'arrivée, cela va donc entraîner des surépaisseurs locales. Les ondulations ainsi créées dans les couches supérieures peuvent également affecter les propriétés mécaniques de la pièce, par exemple en diminuant la charge critique de flambement (Croft et al. (2011)).

Un dernier type de trajectoire produit par cet algorithme n'a pas encore été discuté dans ce

mémoire. En effet, en générant les bases de données de trajectoires associées à chaque pièce (comme décrit au chapitre 4.4.3), il a été observé des trajectoires réalisant une boucle pour revenir sur leur branche de départ (voir figure 6.4). Afin de garder une uniformité dans les trajectoires générées et de respecter les objectifs définis pour cette méthode, à savoir obtenir des trajectoires parcourant deux branches différentes, il a été décidé de systématiquement supprimer ces trajectoires. Cependant, il pourrait être intéressant dans un travail futur d'étudier les propriétés mécaniques de telles trajectoires qui pourraient éventuellement permettre de renforcer la jonction entre les branches, puisqu'elles s'enroulent en général autour de cette région. Dans cette optique, une option a donc été implémentée dans le logiciel permettant de choisir si ces trajectoires doivent être systématiquement supprimées ou non. Pour finir, une étude en simulation sera également nécessaire pour savoir si ce type de trajectoire est effectivement réalisable par la cellule de travail proposée. En effet, le fait d'avoir une plus grande partie de la trajectoire autour de la jonction entraîne l'augmentation du risque de collisions entre l'outil et la pièce. Il est également probable que les limites articulaires du manipulateur soient atteintes dans cette région.

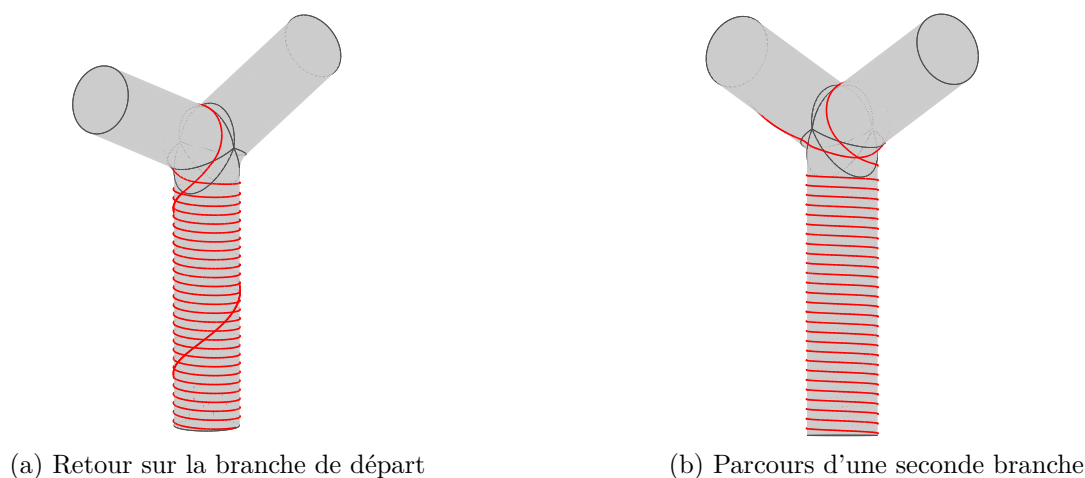


Figure 6.4 Trajectoires effectuant des boucles sur elles-mêmes

6.5 Algorithme de sélection

L'algorithme de sélection proposé dans ce travail permet de choisir la quantité de trajectoires finissant en hélice ou en ligne droite que l'on souhaite appliquer sur la pièce. Or, en utilisant l'algorithme *MCC*, la probabilité d'obtenir une trajectoire ayant effectivement un angle de placement de 0° par rapport à l'axe du cylindre définissant la branche d'arrivée est quasiment

nulle (si on utilise l'option de génération de géodésique naturelle, voir paragraphe 4.4.2). Il a donc été choisi de fixer un seuil pour cet angle de placement final permettant de faire la distinction entre ce que l'on considérera comme des trajectoires terminant en ligne droite et celles terminant en hélice. Celui-ci a été choisi de manière arbitraire. De plus le choix de la proportion de lignes droites ou d'hélices est également remis à l'utilisateur. La qualité des trajectoires sélectionnées en fonction de l'utilisation que l'on souhaite faire de la pièce dépend donc pour le moment de l'expérience de l'utilisateur. Une amélioration majeure envisagée pour cet algorithme serait de développer un code d'analyse par élément finis spécifiquement adapté à l'*AFP*, permettant d'importer et d'analyser les trajectoires générées. Celui-ci serait alors intégré à l'algorithme de sélection et permettrait d'obtenir une pièce optimisée pour un cas de chargement particulier.

6.6 Simulations

La réalisation de simulations du processus de placement de fibre sur des pièces en forme de « Y » à l'aide d'un manipulateur FANUC associé à une table rotative a permis de mettre en évidence plusieurs problèmes encore à résoudre avant l'implémentation sur une cellule de travail réelle. En premier lieu, il est évident qu'il sera nécessaire de concevoir une tête de placement adaptée à la dépose de fibre sur ce type de pièces. En effet, on comprend facilement, au vu des trajectoires générées qu'il va être relativement difficile d'éviter les collisions entre l'outil et le moule. Un premier travail concernant la forme à donner à la tête de placement a été effectué afin de pouvoir réaliser les simulations. Celle-ci est présentée figure 6.5. Si cette dernière conduit à moins de collisions qu'une tête de placement traditionnelle (figure 6.6), les simulations effectuées montrent qu'elle n'est pas encore optimale et qu'un travail de conception plus poussé (qui est au delà du cadre de cette recherche) sera nécessaire.

Comme abordé dans l'article (chapitre 4), les simulations ont également permis de mettre en évidence que la position relative entre les deux robots dans la cellule de travail est essentielle pour réussir à séparer la trajectoire cartésienne définie sur le moule, en deux trajectoires dans les espaces articulaires des robots. En effet, à cause de la complexité des pièces et des trajectoires qui leur sont associées, il est en général nécessaire de rechercher pour chaque trajectoire un positionnement particulier des robots qui permettra de la parcourir sans atteindre les limites physique des robots (limites articulaire et singularités). Nous rappelons que la méthode utilisée ici pour la séparation de la trajectoire de l'espace cartésien vers l'espace articulaire des robots s'apparente à une optimisation locale. En effet, la pose suivante est calculée à partir de la pose courante en optimisant trois critères (minimisation des déplacements du manipulateur, évitement des singularités et évitement des limites articulaires). Cependant,

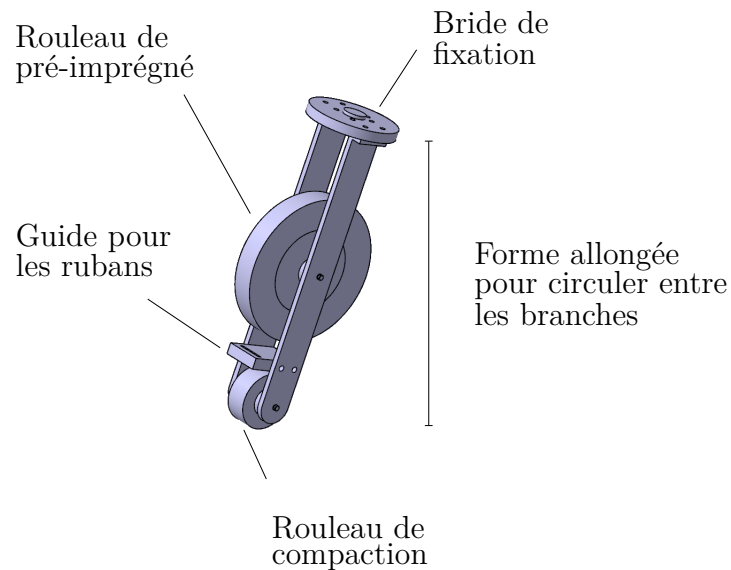


Figure 6.5 Outil utilisé pour les simulations

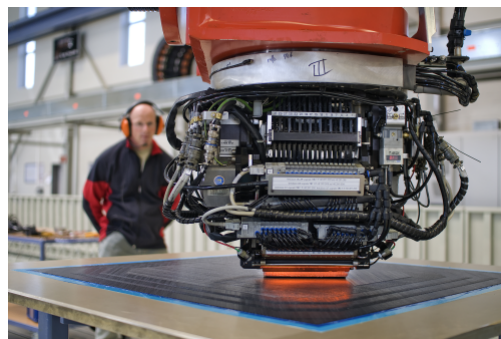


Figure 6.6 Outil traditionnel d'AFP (Rainer (2015))

comme il n'y a pas de prise en compte de la trajectoire dans sa globalité, cette méthode peut conduire, en fonction des configurations initiales des robots, à arriver dans une solution trop proche d'une singularité et donc, à cause de la perte de mobilité, d'empêcher le calcul des poses suivantes de la trajectoire. Pour assurer la faisabilité d'une trajectoire particulière, il peut être envisagé d'effectuer un calcul d'optimisation global sur toute la trajectoire. Celui-ci permettrait alors de trouver le meilleur positionnement relatif des robots de manière à éviter à la fois les limites articulaires et les singularités. Calculer un positionnement relatif pour chaque trajectoire signifie en revanche qu'il faut être en mesure de déplacer facilement les robots.

Une autre solution qui peut être envisagée pour s'affranchir de ce problème de mobilité des robots est d'utiliser plus de DDL. En effet, utiliser un second robot à 6DDL pour tenir la table rotative sur laquelle est fixé le moule (portant à 13 le nombre de DDL total), permet

d'augmenter l'espace possible des solutions pour la séparation des trajectoires (figure 6.7). On peut alors penser, sous réserve que l'espace de travail des robots soit suffisamment important, qu'il sera plus probable de trouver des trajectoires dans l'espace articulaire des robots évitant les limites articulaire et les singularités des deux robots, sans avoir besoin de modifier leur position relative.

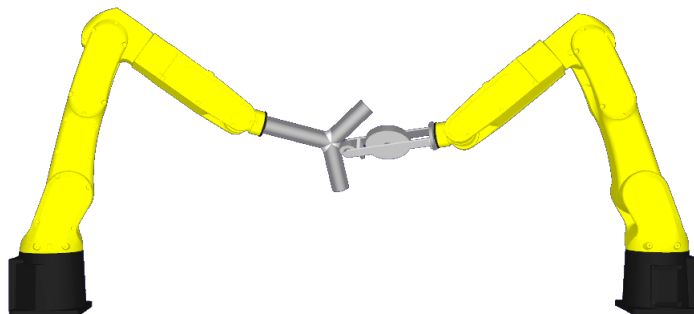


Figure 6.7 Utilisation de deux manipulateurs dans la cellule de travail

Pour finir, un dernier point qui n'a pas été traité dans le calcul de séparation des trajectoires est celui des collisions entre les robots ou entre une membrure d'un robot et le moule. Une fois encore on peut souligner que l'utilisation de 13 DDL et d'un calcul d'optimisation global sur toute la trajectoire permettrait d'éviter plus facilement ce type de collisions.

CHAPITRE 7 CONCLUSION

Le but du projet dans lequel s'inscrit cette recherche est de mettre en place une cellule de placement de fibre automatisée (*AFP*), permettant de fabriquer des structures en matériaux composites. La particularité de cette cellule est qu'elle doit, à terme, comprendre plusieurs robots travaillant en collaboration afin de fabriquer des pièces de géométrie complexe. L'hypothèse considérée est que l'ajout de degrés de liberté (DDL) va permettre de traiter des pièces de géométries plus complexes et de tailles plus faibles que celles qui sont actuellement produites par ce procédé. En effet, celles-ci sont souvent des plaques ou des coques de grandes dimensions que l'on peut considérer comme localement planes, comme par exemple des fuselages d'avion. Cependant différents secteurs d'activités industriels s'intéressent à présent à la production de nouvelles géométries de pièces.

L'objectif spécifique de cette recherche était de mettre au point un outil logiciel de planification de trajectoire pour *AFP* dédié à la production de pièces de faibles dimensions et de géométries complexes. Plus particulièrement, cette recherche s'est consacrée à la génération de chemins sur des surfaces de révolution multi-axes, telles que des tubes en forme de « Y » dont les branches ont une section circulaire et dont l'angle entre celles-ci peut être variable. Cette étude a permis de **développer quatre principaux algorithmes**, décrits dans ce mémoire, permettant la couverture en pré-imprégné de telles pièces.

La méthodologie utilisée pour développer les deux premiers algorithmes (HD et HA) fut de couvrir les différentes branches des « Y » individuellement grâce à des trajectoires formées à base d'hélices. Les branches doivent ensuite être liées entre elles grâce à une trajectoire générée sur l'intersection des cylindres formant les branches. Bien que cette approche semblait à première vue la plus intuitive, il s'est révélé qu'elle conduisait à de multiples problèmes. En effet le nombre important de coupes dans le cas de l'hélice discontinue (HD) risque d'entraîner un ralentissement du processus et l'on peut facilement envisager que ces multiples coupes pratiquées dans la fibre vont entraîner un affaiblissement des propriétés mécaniques de la pièce. La méthode de l'hélice ajustable (HA) conduit quant à elle à la définition de trajectoires dont la majeure partie présente une courbure géodésique supérieure à ce que peut supporter le procédé d'*AFP* classique. Une courbure géodésique trop importante des bandes de pré-imprégné conduira en effet au flambement local des fibres microscopiques et donc également à une diminution des propriétés mécaniques de la pièce finale. De plus il était également probable d'observer une faiblesse mécanique au niveau de la liaison entre les branches puisque celle-ci se trouvait être un ajout de matière en complément de la couverture

des branches.

Ces différents problèmes ont conduit à modifier les pistes de réflexion et à envisager deux nouveaux algorithmes. Le but de ces derniers fut de maximiser la raideur et la résistance mécanique des pièces produites en couvrant deux branches du « Y » avec une trajectoire continue, c'est-à-dire sans coupe. Ces deux algorithmes (*CPA*) et (*MCC*) sont décrits dans l'article soumis au journal *Robotics and Computer Integrated Manufacturing*. Le premier (*CPA*) reprend le concept, largement utilisé dans l'industrie, de générer chacune des couches du stratifié avec une orientation de fibre constante par rapport à une référence. Cette méthode est adaptée dans notre travail au cas des pièces en forme de « Y » en définissant l'angle de placement de la fibre par rapport à l'axe du cylindre qui est en train d'être couvert. Des trajectoires équidistantes, séparées d'une distance égale à la largeur de la bande déposée sont ensuite générées. Il est alors possible, grâce à une bonne combinaison des couches déposées, de couvrir l'intégralité du moule. Si cette méthode semble prometteuse pour générer une couverture uniforme, elle souffre tout de même d'un inconvénient majeur. En effet, elle a tendance à produire des trajectoires présentant de fortes courbures géodésiques dans la région de l'intersection des différents cylindres. Cependant comme discuté en section 6 il est possible que le développement des nouvelles techniques d'*AFP* apporte des solutions à ce problème dans le futur et permette l'utilisation de telles trajectoires sans apparition de défauts dans la pièce finale.

Le dernier algorithme proposé a été mis au point dans l'optique d'éliminer ce problème de courbure, tout en conservant une trajectoire couvrant deux branches sans coupe. Il permet alors de produire des trajectoires de placement de fibre réalisable en pratique avec les techniques d'*AFP* actuelles. Son principal inconvénient est qu'il est impossible de prévoir la forme complète de la trajectoire et donc d'assurer une couverture uniforme du moule. C'est pour palier à cet inconvénient qu'il a été décidé d'utiliser cette méthode pour générer un ensemble aussi exhaustif que possible de trajectoires pour une pièce donnée, parmi lesquelles il sera ensuite possible de choisir celles effectivement utilisées lors de la fabrication. Ce choix sera aidé par la définition de critères de sélection et l'utilisation d'un algorithme qui permet d'éliminer les trajectoires non désirées. Le choix dans les critères peut être appuyé en utilisant une analyse par éléments finis, mise au point avec un modèle simplifié, qui permet d'identifier les formes de trajectoires les plus appropriées à un chargement donné. Cette analyse qui dépasse le cadre de cette recherche est restée à l'état d'expérimentation. Un travail intéressant pour le futur serait de développer un modèle plus précis pour cette analyse par éléments finis et de l'intégrer directement à l'algorithme de sélection afin de définir une sélection de trajectoire optimisée pour un cas de chargement.

Pour compléter cette recherche des simulations ont été réalisées avec une cellule de travail comprenant un manipulateur FANUC et une table rotative (7 DDL au total). **Elles ont permis de valider la faisabilité de certaines trajectoires produites à l'aide des algorithmes précédemment cités**, mais aussi de souligner les différents challenges qui restent encore à relever avant de mettre en place une cellule d'*AFP* réelle permettant de fabriquer les pièces étudiées dans ce travail. Il est notamment ressorti que, certaines trajectoires ne seraient potentiellement pas réalisables à cause des limites physiques du manipulateur (limites articulaires, singularités), sans l'ajout de DDL supplémentaires. De plus, les simulations ont montré qu'il est souvent nécessaire de modifier la configuration de la cellule de travail pour permettre la réalisation de trajectoires de différentes formes, tout en respectant les contraintes du robot. Cela signifie qu'il faudrait être en mesure de calculer une configuration optimale de la cellule de travail pour chaque trajectoire et donc que la cellule soit modulable.

Pour conclure, on peut dire que les différentes méthodes présentées dans ce mémoire ont permis de générer des trajectoires sur de nouvelles géométries de pièces plus complexes que celles réalisées actuellement. Il reste néanmoins des travaux de recherche à réaliser, notamment pour le développement de la tête de placement et sur l'étude de la configuration de la cellule de travail, afin de pouvoir mettre en pratique le placement de fibre automatisé sur ces géométries complexes.

RÉFÉRENCES

- E. Abbena, S. Salamon, et A. Gray, *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 2006.
- J. Angeles, *Fundamentals of robotic mechanical systems*. Springer, 2002, vol. 2.
- Argon18. Argon18 e-119 tri+. En ligne : <https://www.argon18bike.com/en/bikes/triathlon/e-119-tri-plus>
- P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, et A. A. Rizzi, “Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage”, *Automation Science and Engineering, IEEE Transactions on*, vol. 6, no. 1, pp. 107–120, 2009.
- S. Black. (2003) New equipment brings tape placement within reach of smaller shops. En ligne : <http://www.compositesworld.com/articles/new-equipment-brings-tape-placement-within-reach-of-smaller-shops>
- A. W. Blom, M. M. Abdalla, et Z. Gürdal, “Optimization of course locations in fiber-placed panels for general fiber angle distributions”, *Composites science and technology*, vol. 70, no. 4, pp. 564–570, 2010.
- A. W. Blom, *Structural performance of fiber-placed, variable-stiffness composite conical and cylindrical shells*. TU Delft, Delft University of Technology, 2010.
- M. Bruyneel et S. Zein, “A modified fast marching method for defining fiber placement trajectories over meshes”, *Computers & Structures*, vol. 125, pp. 45–52, 2013.
- Cadfil®. The filament winding process. En ligne : <http://www.cadfil.com/filamentwindingprocess.html>
- H. Chen, T. Fuhlbrigge, et X. Li, “A review of cad-based robot path planning for spray painting”, *Industrial Robot : An International Journal*, vol. 36, no. 1, pp. 45–50, 2009.
- C. Composites. Fiber placement robot. En ligne : <http://www.coriolis-composites.com/products/fiber-placement-process.html>

K. Croft, L. Lessard, D. Pasini, M. Hojjati, J. Chen, et A. Yousefpour, “Experimental study of the effect of automated fiber placement induced defects on performance of composite laminates”, *Composites Part A : Applied Science and Manufacturing*, vol. 42, no. 5, pp. 484–491, 2011.

J. R. Current et D. A. Schilling, “The covering salesman problem”, *Transportation science*, vol. 23, no. 3, pp. 208–213, 1989.

H.-J. L. Dirk, C. Ward, et K. D. Potter, “The engineering aspects of automated prepreg layup : History, present and future”, *Composites Part B : Engineering*, vol. 43, no. 3, pp. 997–1009, 2012.

Fives. Cincinnati viper® fps (système à placement de fibres). En ligne : <http://metal-cutting-composites.fivesgroup.com/fr/produits/composites/fiber-placement-systems/cincinnati-viperr-fps-systeme-a-placement-de-fibres.html>

J. Goldfeather et V. Interrante, “A novel cubic-order algorithm for approximating principal direction vectors”, *ACM Transactions on Graphics (TOG)*, vol. 23, no. 1, pp. 45–63, 2004.

W. Goldsworthy, E. Hardesty, et H. Karlson, “Geodesic path length compensator for composite-tape placement head”, Mai 14 1974, uS Patent 3,810,805.

A. Grant, “Sporting composites”, *Reinforced Plastics*, vol. 49, no. 5, pp. 46–49, 2005.

Z. Gürdal, B. F. Tatting, et C. Wu, “Variable stiffness composite panels : effects of stiffness variation on the in-plane and buckling response”, *Composites Part A : Applied Science and Manufacturing*, vol. 39, no. 5, pp. 911–922, 2008.

H. Hahn, D. Jensen, S. Claus, S. Pai, et P. Hipp, “Structural design criteria for filament-wound composite shells”, 1994.

N. Hamila, “Simulation de la mise en forme des renforts composites mono et multi plis”, Thèse de doctorat, Villeurbanne, INSA, 2007.

Z. Han, F. Qiangqiang, Y. Fan, F. Yunzhong, et F. Hongya, “A path planning algorithm of closed surface for fiber placement”, dans *Proceedings of the 1st International Conference on Mechanical Engineering and Material Science*. Atlantis Press, 2012.

HEXCEL®. Hexply prepreg technology. En ligne : http://www.hexcel.com/Resources/DataSheets/Brochure-Data-Sheets/Prepreg_Technology.pdf

L. Huo et L. Baron, “The joint-limits and singularity avoidance in robotic welding”, *Industrial Robot : An International Journal*, vol. 35, no. 5, pp. 456–464, 2008.

R. M. Jones, *Mechanics of composite materials*. CRC press, 1998.

H. Khakpour, L. Birglen, et S.-A. Tahan, “Uniform scanning path generation for abrasive waterjet polishing of free-form surfaces modeled by triangulated meshes”, *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 5-8, pp. 1167–1176, 2015.

B. C. Kim, K. Potter, et P. M. Weaver, “Continuous tow shearing for manufacturing variable angle tow composites”, *Composites Part A : Applied Science and Manufacturing*, vol. 43, no. 8, pp. 1347–1356, 2012.

B. C. Kim, P. M. Weaver, et K. Potter, “Computer aided modelling of variable angle tow composites manufactured by continuous tow shearing”, *Composite Structures*, vol. 129, pp. 256–267, 2015.

A. Lasemi, D. Xue, et P. Gu, “Recent development in cnc machining of freeform surfaces : A state-of-the-art review”, *Computer-Aided Design*, vol. 42, no. 7, pp. 641–654, 2010.

E. Lemaire, S. Zein, et M. Bruyneel, “Optimization of composite structures with curved fiber trajectories”, *Composite Structures*, vol. 131, pp. 895 – 904, 2015. DOI : <http://dx.doi.org/10.1016/j.compstruct.2015.06.040>. En ligne : <http://www.sciencedirect.com/science/article/pii/S0263822315005048>

L. Li, X. Wang, D. Xu, et M. Tan, “Path planning of airfoil surface for robotic fibre placement”, dans *Advanced Mechatronic Systems (ICAMechS), 2015 International Conference on*. IEEE, 2015, pp. 316–321.

A. Liegeois, “Automatic supervisory control of the configuration and behavior of multibody mechanisms”, *IEEE transactions on systems, man, and cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

J. Liu, Y. Chen, J. M. Maisog, et G. Luta, “A new point containment test algorithm based on preprocessing and determining triangles”, *Computer-Aided Design*, vol. 42, no. 12, pp. 1143–1150, 2010.

A. Marouene, “Résistance à la compression et au flambage des composites carbone/époxy à rigidité variable fabriqués par le procédé de placement automatique des fibres”, Thèse de doctorat, École Polytechnique de Montréal, 2015.

D. A. McCarville, “Evolution of and projections for automated composite material placement equipment in the aerospace industry”, Thèse de doctorat, Walden University, 2009.

MTorres®. Torreslayup. En ligne : <http://www.mtorres.es/en/aeronautics/products/carbon-fiber/torreslayup>

K. Nilsson, “Industrial robot programming”, Thèse de doctorat, Lund University, 1996.

P. Olivieri, L. Birglen, X. Maldague, et I. Mantegh, “Coverage path planning for eddy current inspection on complex aeronautical parts”, *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 305–314, 2014.

—, “Path planning for eddy current inspection around probable defects”, *Canadian Aeronautics and Space Journal*, vol. 60, no. 3, pp. 49–57, 2014.

P. Olivieri, “Génération de chemins de couverture pour des opérations automatisées de contrôle non destructif appliquées dans l’industrie aérospatiale”, Mémoire de maîtrise, École Polytechnique de Montréal, 2013.

R. Olmedo et Z. Gurdal, “Buckling response of laminates with spatially varying fiber orientations”, dans *AIAA/ASME/ASCE/AHS/ASC 34th Structures, Structural Dynamics, and Materials Conference*, vol. 1, 1993, pp. 2261–2269.

H. B. Olsen et J. J. Craig, “Automated composite tape lay-up using robotic devices”, dans *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on.* IEEE, 1993, pp. 291–297.

L. Parnas, S. Oral, et Ü. Ceyhan, “Optimum design of composite structures with curved fiber courses”, *Composites Science and Technology*, vol. 63, no. 7, pp. 1071–1082, 2003.

G. Rainer. (2015) Variable stiffness composites. En ligne : <http://aerospaceengineeringblog.com/page/2/>

W. G. Roeseler, B. Sarh, M. Kismarton, J. Quinlivan, J. Sutter, et D. Roberts, “Composite structures : the first 100 years”, dans *16th International Conference on Composite Materials*, 2007, pp. 1–41.

A. Sawicki et P. Minguet, “The effect of intraply overlaps and gaps upon the compression strength of composite laminates”, dans *39th AIAA structural, dynamics, & materials conference. Long Beach, CA*, 1998, pp. 744–54.

S. Scott. (2011) Wind blade manufacture : Opportunities and limits. En ligne : <http://www.compositesworld.com/columns/wind-blade-manufacture-opportunities-and-limits>

J. C. Serrano et S. Vennam. (2012) Automated wind blade production. En ligne : <http://www.windsystemsmag.com/article/detail/364/automated-wind-blade-production>

J. A. Sethian, “A fast marching level set method for monotonically advancing fronts”, *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

—, *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge university press, 1999, vol. 3.

F. C. Shen, “A filament-wound structure technology overview”, *Materials Chemistry and Physics*, vol. 42, no. 2, pp. 96–100, 1995.

W. Sheng, H. Chen, N. Xi, et Y. Chen, “Tool path planning for compound surfaces in spray forming processes”, *Automation Science and Engineering, IEEE Transactions on*, vol. 2, no. 3, pp. 240–249, 2005.

B. Shirinzadeh, G. Alici, C. W. Foong, et G. Cassidy, “Fabrication process of open surfaces by robotic fibre placement”, *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 1, pp. 17–28, 2004.

B. Shirinzadeh, G. Cassidy, D. Oetomo, G. Alici, et M. H. Ang Jr, “Trajectory generation for open-contoured structures in robotic fibre placement”, *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 380–394, 2007.

J. Sloan. (2008) Atl and afp : Defining the megatrends in composite aerostructures. En ligne : <http://www.compositesworld.com/articles/atl-and-afp-defining-the-megatrends-in-composite-aerostructures>

H.-y. Tam, “Toward the uniform coverage of surfaces by scanning curves”, *Computer-Aided Design*, vol. 31, no. 9, pp. 585–596, 1999.

G. Thürmer et C. A. Wüthrich, “Computing vertex normals from polygonal facets”, *Journal of Graphics Tools*, vol. 3, no. 1, pp. 43–46, 1998.

T.-T. Tran, V.-T. Cao, et D. Laurendeau, “Extraction of cylinders and estimation of their parameters from point clouds”, *Computers & Graphics*, vol. 46, pp. 345–357, 2015.

L. E. Turoski, “Effects of manufacturing defects on the strength of toughened carbon/epoxy prepreg composites”, Mémoire de maîtrise, MONTANA STATE UNIVERSITY–BOZEMAN, 2000.

C. Waldhart, “Analysis of tow-placed, variable-stiffness laminates”, Mémoire de maîtrise, Virginia Polytechnic Institute and State University, 1996.

W. Xiaoping, A. Luling, Z. Liyan, et Z. Laishui, “Uniform coverage of fibres over open-contoured freeform structure based on arc-length parameter”, *Chinese Journal of Aeronautics*, vol. 21, no. 6, pp. 571–577, 2008.

L. Yan, Z. C. Chen, Y. Shi, et R. Mo, “An accurate approach to roller path generation for robotic fibre placement of free-form surface composites”, *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 277–286, 2014.

ANNEXE A LOGICIEL DE PLANIFICATION DE TRAJECTOIRES - MANUEL UTILISATEUR

Nous fournissons dans cette annexe un manuel expliquant le fonctionnement et l'utilisation du programme de génération de trajectoires développé pendant cette recherche. Celui-ci se présente sous la forme d'une interface utilisateur développée dans l'environnement MATLAB.

A.1 Présentation générale

Ce programme offre plusieurs possibilités pour générer des trajectoires sur différentes pièces. Celle-ci sont représentées sous forme de maillages triangulés contenus dans des fichiers .STL. Le programme fonctionne bien pour des fichiers .STL provenant du logiciel CATIA. Toutefois, il n'est pas garanti que toutes les fonctions fonctionnent correctement pour des fichiers .STL provenant d'autres logiciels de CAO. Deux types de pièces peuvent être traités par ce programme : les pièces en forme de « Y » et les pièces « quelconques » (de géométrie relativement simple). Il est possible de générer des trajectoires simples sur ces dernières grâce à l'algorithme itératif présenté en section 4.4.1. Quatre algorithmes plus poussés permettent de générer les trajectoires sur les pièces en « Y ». L'organigramme ci-dessous (figure A.1) résume les différentes fonctionnalités du programme :

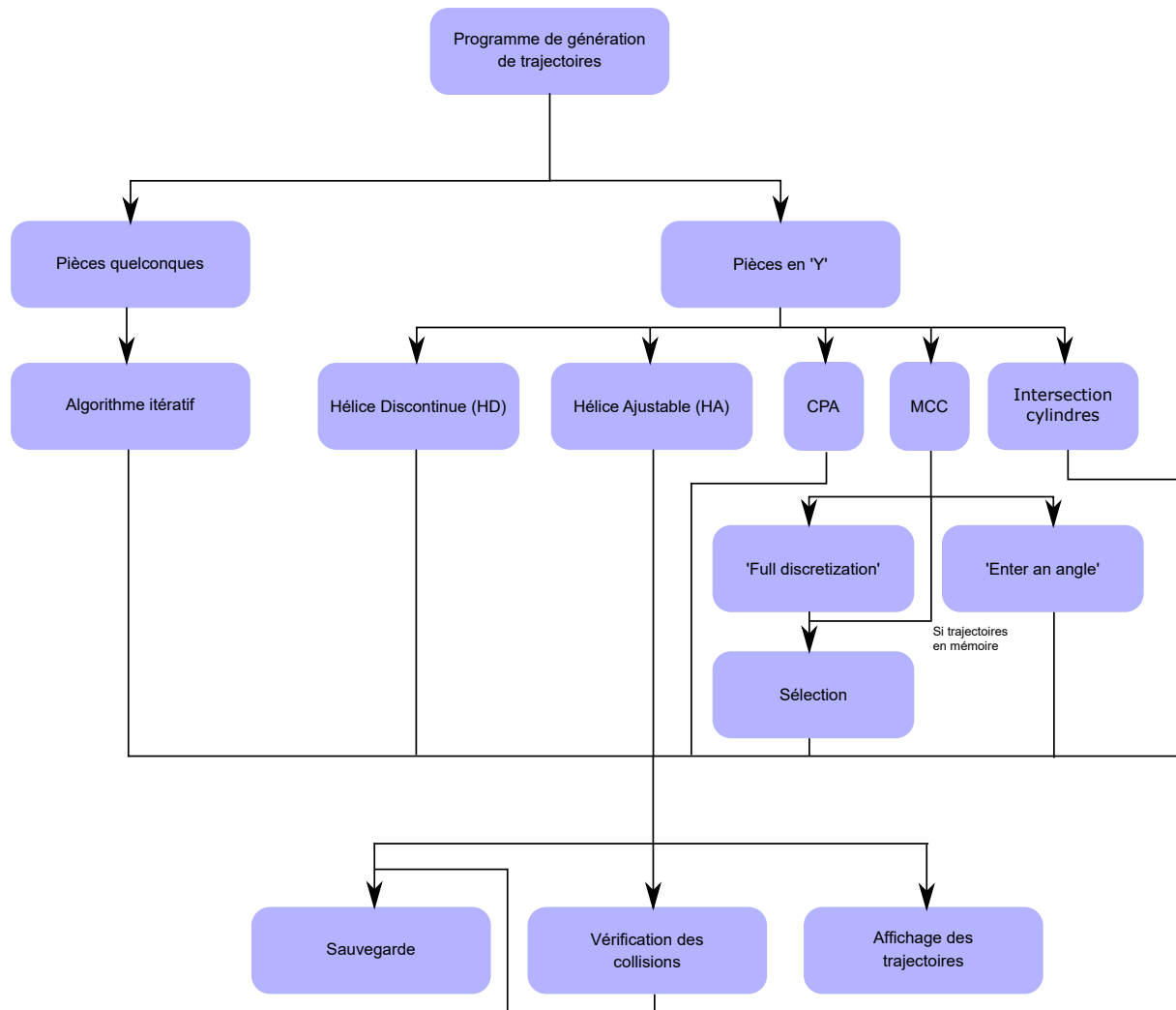


Figure A.1 Organigramme général du programme

Le démarrage du programme nécessite d'ouvrir MATLAB et de se placer dans le répertoire : **AFP_PathPlanning**. L'exécution de la commande **interface** dans l'invite de commandes va déclencher l'ouverture de la fenêtre présentée à la figure A.2. Celle-ci comporte plusieurs zones :

1. **Aire de visualisation** : permet l'affichage en 3D des pièces à traiter. On peut y visualiser les différentes régions obtenues grâce à la segmentation ainsi que les trajectoires générées. C'est dans cette fenêtre que s'effectue la sélection des régions de travail et points de départ des trajectoires lorsque cela est nécessaire.
2. **Boutons de commandes** : c'est dans cette zone que se situent les boutons permettant d'effectuer les actions principales, comme la segmentation et la génération de trajectoires. Elle permet également le réglage de plusieurs paramètres de calcul (pas, angle de placement ...).
3. **Barre de menus** : Permet des actions de base telles que la sauvegarde et le chargement de données, l'initialisation ainsi que la sélection de certaines options du programme.
4. **Barre d'outils** : comprend les outils classiques de MATLAB permettant d'interagir avec l'interface de visualisation 3D.
5. **Boite d'informations** : affiche à l'utilisateur des informations sur l'état du programme.

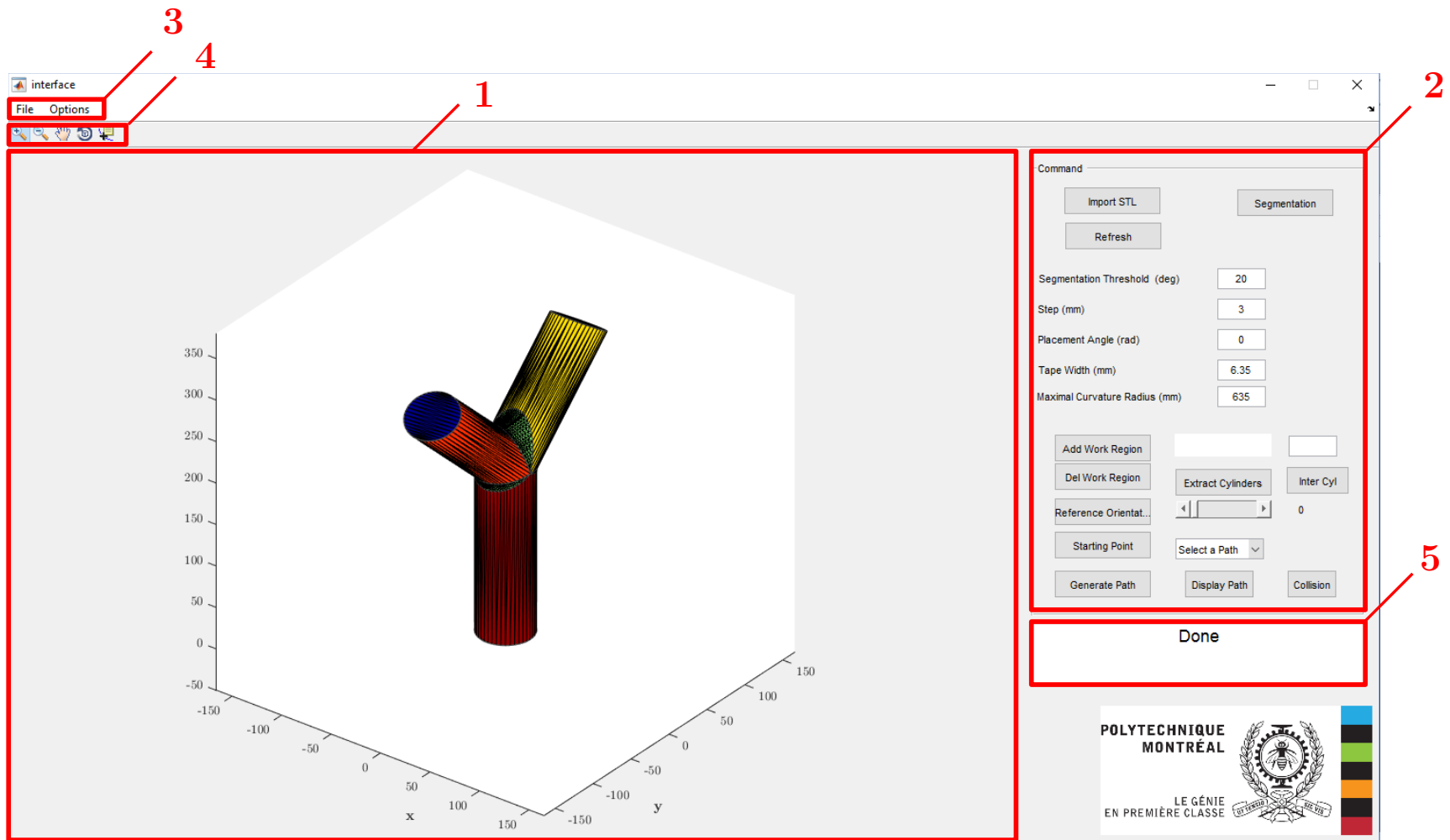


Figure A.2 Vue Générale du Programme de génération de trajectoires

A.2 Présentation des commandes

Nous détaillerons dans ce paragraphe la fonction des différentes commandes. La figure A.3 permet de visualiser l'emplacement de ces différents boutons.

1. **Import STL** : permet de charger un fichier CAO au format .STL, représentant la pièce sur laquelle on souhaite générer les trajectoires. Les deux formats de sauvegarde autorisés en .STL (binaire et ASCII) sont traités par ce programme. Une fois le chargement terminé, une image de la pièce s'affiche dans la zone de visualisation.
2. **Segmentation** : permet d'effectuer la segmentation en régions lisses. Les frontières entre ces régions seront situées entre les facettes dont les normales forment un angle supérieur à un seuil, choisi avec le paramètre **Segmentation threshold (deg)**. Cette segmentation offre également une option permettant de séparer les régions par la taille de leur plus grand coté. Pour l'activer vérifier que **Options** → **Segmentation By Edge Length** est bien coché. Cela est utile dans le cas des pièces en forme de « Y » pour extraire la région de la jonction et ainsi obtenir une extraction des cylindres plus précise. Une fois la segmentation effectuée, les différentes régions de la pièce sont représentées dans l'aire de visualisation en utilisant différentes couleurs.
3. **Refresh** : permet de réinitialiser l'affichage en ne montrant que la pièce.
4. **Segmentation Threshold (deg)** : permet de choisir l'angle (en degrés) maximal entre les normales de deux facettes voisines appartenant à une même région.
5. **Step** : définit le pas utilisé dans l'algorithme itératif de génération de trajectoire présenté en section 4.4.1, c'est-à-dire la distance entre deux points de la trajectoire.
6. **Placement Angle** : permet de définir l'angle de placement de fibre par rapport à une référence (qui sera choisie plus tard). Ce réglage n'influence le calcul que si l'on traite des pièces de géométrie quelconque (différentes des « Y »). Dans le cas des « Y », cet angle est calculé automatiquement.
7. **Tape Width** : définit la largeur de la bande de pré-imprégné déposée (en mm). Cette valeur est par exemple utilisée pour définir la distance de décalage des trajectoires dans l'algorithme CPA.
8. **Maximal Curvature Radius** : définit le rayon de courbure géodésique maximum autorisé pour la génération de trajectoires dans l'algorithme MCC.

9. **Add Work Region** : permet de sélectionner une région de travail en cliquant sur la région correspondante dans l'aire de visualisation (ex : pour l'extraction des cylindres, pour sélectionner une branche de travail). Les labels régions sont ajoutés à la liste située dans le cadre à droite de ce bouton. Attention ce bouton doit être pressé à chaque fois que l'on souhaite ajouter une nouvelle région (il n'est pas possible d'en ajouter plusieurs à la suite sans cliquer à nouveau dessus).
10. **Del Work Region** : permet de supprimer une région de la liste des régions de travail en cliquant sur cette région dans l'aire de visualisation. Son fonctionnement est identique à celui de la commande **Add Work Region**.
11. **Extract Cylinders** : permet de lancer l'algorithme d'extraction des cylindres. Nécessite de sélectionner la région sur laquelle effectuer le calcul. Un seuil réglant un des paramètres de validation de l'algorithme doit également être réglé grâce au curseur de défilement situé sous le bouton **Extract Cylinders**. Une valeur de 100% fonctionne en général bien avec les pièces produites grâce au logiciel CATIA, mais cette valeur est à ajuster en fonction du raffinement du maillage.
12. **Inter Cyl** : permet de générer une trajectoire sur l'intersection des cylindres. Nécessite d'avoir préalablement effectué l'extraction des cylindres.
13. **Reference Orientation** : permet de définir une direction de référence pour le placement de fibre. Cette définition n'est obligatoire que pour les pièces quelconques, les algorithmes de génération de trajectoires sur les pièces en « Y » calculent cette direction automatiquement. Ce calcul nécessite la sélection préalable d'une région de travail. La direction choisie sera alors projetée sur la surface de la pièce, en chaque sommet du maillage. Six options sont possibles :
 - (a) **x, y** ou **z**, les trois axes du repère global.
 - (b) **Draw it**. La dessiner avec une spline. Dans ce cas la direction de référence choisie est celle de la direction de courbure minimale de la surface de la pièce. La spline ne sert qu'à orienter ces vecteurs pour qu'ils pointent tous dans le sens voulu.
 - (c) **Entrer a vector**. L'utilisateur peut lui-même donner les coordonnées d'un vecteur (dans le repère global) qui va définir la direction de référence.
 - (d) **Cylinder**. Choisi la direction de l'axe du cylindre sélectionné comme région de travail (dans le cas où une extraction de cylindre a été préalablement réalisée).
14. **Starting Point** : Permet de sélectionner un point de départ pour la trajectoire. Ce point est situé sur un des sommets du maillage. La définition du point de départ n'est

pas nécessaire pour tous les algorithmes, se référer à la section A.4 pour savoir lesquels proposent un calcul automatique de ce point de départ.

15. **Generate Path** : permet de lancer le calcul de génération de trajectoire. Différents algorithmes sont disponibles, se référer à la section A.4 pour plus d'informations.
16. Menu déroulant **Select a Path** : permet de sélectionner une trajectoire à afficher. Plusieurs options sont disponibles :
 - (a) **All** : affiche toutes les trajectoires en mémoire.
 - (b) **B1**, **B2** ou **B3** : affiche respectivement les trajectoires sur les branches 1, 2 ou 3 (dans le cas des pièces en « Y »)
 - (c) Sélection du numéro de la trajectoire à afficher.

La trajectoire s'affiche ensuite automatiquement. Attention un premier clic sur **Select a Path** est nécessaire pour que le menu déroulant s'initialise.

17. **Display Path** : permet d'afficher la trajectoire sélectionnée par le menu déroulant décrit dans le point précédent.
18. **Collision** : permet de lancer le programme de vérification des collisions entre l'outil et la pièce. Un modèle CAO (.STL) de l'outil doit être préalablement chargé (voir section A.3).
19. Zone d'affichage des labels des régions de travail sélectionnées.
20. Zone d'affichage de l'indice de la face sélectionnée.

A.3 Barre des menus

La barre des menus propose différentes fonctionnalités décrites dans la liste suivante :

1. **File** :

- **New** : permet de réinitialiser le programme. Supprime toutes les données en mémoire (pièce, régions, trajectoires...).
- **Import Tool** : permet de charger un fichier d'outil (.STL) qui sera ensuite utilisé pour vérifier si la trajectoire entraîne des collisions entre l'outil et la pièce.

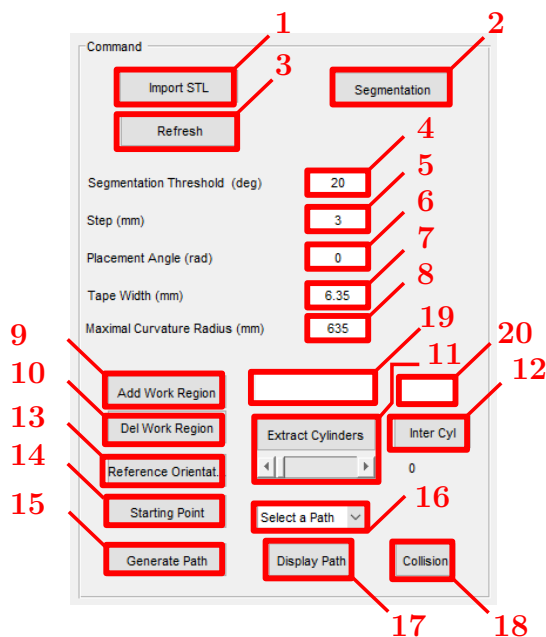


Figure A.3 Identification des commandes

- **Save** : permet de sauvegarder les données en mémoire (trajectoires, pièce, régions, cylindres extraits).
 - **Save figure** : permet de générer une figure MATLAB de l'aire de visualisation. Attention cette figure n'est pas sauvegardée automatiquement.
 - **Load** : permet de charger un travail (trajectoires, pièce, régions, cylindres extraits) précédemment sauvegardé.
2. **Options** : les différentes options proposées dans ce menu déroulant peuvent être cochées (activées) ou décochées (désactivées) en cliquant dessus.
- **Display Orientation** : Permet d'afficher les repères associés à chaque pose de la trajectoire, représentant l'orientation de l'outil. Cette option ralentit considérablement l'affichage.
 - **Dynamic Display** : permet l'affichage en temps réel des points de la trajectoire lors du calcul de celle-ci. Cette option ralentit considérablement le calcul de la trajectoire.
 - **Y-shaped part** : permet d'indiquer au programme si l'on travaille ou non sur une pièce en forme de « Y ».
 - **Color Grey** : permet d'afficher la pièce en couleur grise ou en couleurs représentant les différentes régions.

- **Self Intersection** : permet d'autoriser ou non la conservation des trajectoires effectuant des boucles sur elles-mêmes par l'algorithme de sélection.
- **Segmentation By Edge Length** : permet d'activer l'option permettant de séparer les facettes ayant de longs côtés de celles en ayant des petits lors de la segmentation. Cela permet, dans le cas des pièces en « Y », d'extraire la région jonction et d'obtenir une meilleure approximation des cylindres.



A.4 Procédure de génération de trajectoires

Lors de la génération de trajectoires, deux cas principaux peuvent se présenter, on cherche à générer une trajectoire sur une pièce quelconque, ou sur une pièce en forme de « Y ». Pour choisir entre ces deux options utiliser : **Options** → **Y-shaped part** (voir section A.3). Ces deux cas sont détaillés dans les paragraphes suivants.

A.4.1 Pièces quelconques

Cette section décrit pas à pas la procédure à suivre pour générer des trajectoires sur des pièces de géométries quelconques (différentes de « Y »). Pour plus d'informations sur les commandes utilisées, voir section A.2.

1. Ouvrir **MATLAB**. Se placer dans le répertoire **AFP_PathPlanning**. Exécuter la commande **interface** dans l'invite de commandes pour lancer le programme. La fenêtre présentée en figure A.2 s'ouvre.
2. Dans le menu **Options** *décocher* **Y-Shaped Part**.
3. Cliquer sur **Import STL** et choisir le fichier à ouvrir dans l'explorateur Windows qui s'est ouvert (dossier **CAO**). Une fois la pièce chargée, elle s'affiche dans l'aire de visualisation.
4. Cliquer sur **Segmentation**. La pièce est divisée en régions séparées par des angles vifs. Il est possible de régler le seuil de segmentation dans la case **Segmentation Threshold (deg)**, celui-ci correspond à l'angle maximum autorisé entre deux facettes voisines appartenant à une même région (par défaut 20°). Si l'option **Color Grey** n'est pas activée, les différentes régions s'affichent en différentes couleurs. Il est conseillé de désactiver l'option **Segmentation by Edge Length** dans le menu **Options**.
5. Définir les paramètres pour la génération de la trajectoire **Step** et **Placement Angle** (par défaut : 3 mm et 0 rad).

6. Cliquer sur **Add Work Region** pour ajouter les régions sur lesquelles la trajectoire doit être calculée (voir section A.2 pour plus d'informations sur le fonctionnement de cette commande). Si, pendant le calcul, la trajectoire arrive sur une région qui ne fait pas partie de la sélection, le calcul s'arrête.
7. Cliquer sur **Reference Orientation** pour choisir une direction de référence à partir de laquelle sera calculée l'orientation des fibres à l'aide de l'angle de placement. Une fenêtre s'ouvre, sélectionner une des 6 options proposées. Une fois le calcul effectué, une autre fenêtre s'ouvre, proposant d'inverser le sens de la direction de référence calculée. Choisir ce qui vous convient.
8. Cliquer sur **Starting Point** pour choisir un point de départ. Utiliser les boutons  pour déplacer la vue (zoom +/- , translation, rotation) et  pour sélectionner le point voulu (ce dernier bouton est activé par défaut lors du clic sur **Starting Point**). Une fois le point de départ sélectionné, valider en appuyant sur Entrée. (Il est parfois possible que MATLAB ait en mémoire que la touche Entrée a déjà été activée, empêchant la sélection du point de départ. Dans ce cas appuyer sur une autre touche du clavier avant de cliquer sur **Starting Point**).
9. Cliquer sur **Generate Path** pour lancer la génération de la trajectoire. Le nombre de points générés est affiché en temps réel dans la boîte d'informations. Pour afficher en temps réel ces points sur la pièce cliquer sur **Options** → **Dynamic Display**. Lorsque le calcul est terminé, le nombre de point total ainsi que le temps de calcul est affiché dans la boîte de dialogue (attention il est possible que le calcul du dernier point de la trajectoire prenne plus de temps).
10. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
11. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).

A.4.2 Pièces en forme de « Y »

Cette section décrit pas à pas la procédure à suivre pour générer des trajectoires sur des pièces en forme de « Y ». Pour plus d'informations sur les commandes utilisées, voir section A.2.

1. Ouvrir **MATLAB**. Se placer dans le répertoire **AFP_PathPlanning**. Exécuter la commande **interface** dans l'invite de commandes pour lancer le programme. La fenêtre présentée en figure A.2 s'ouvre.
2. Dans le menu **Options** vérifier que l'option **Y-Shaped Part** est bien cochée.
3. Cliquer sur **Import STL** et choisir le fichier à ouvrir dans l'explorateur Windows qui s'est ouvert (dossier **CAO**). Une fois la pièce chargée, elle s'affiche dans l'aire de visualisation.
4. Cliquer sur **Segmentation**. La pièce est divisée en régions séparées par des angles vifs. Il est possible de régler le seuil de segmentation dans la case **Segmentation Threshold (deg)**, celui-ci correspond à l'angle maximum autorisé entre deux facettes voisines appartenant à une même région (par défaut 20°). Si l'option **Segmentation By Edge Length** est cochée, les facettes ayant de petits côtés (jonction) sont également regroupées dans une même région (figure A.4). Si l'option **Color Grey** n'est pas activée, les différentes régions s'affichent en différentes couleurs.
5. Définir les paramètres pour la génération de la trajectoire (Step, Tape Width et Maximal Curvature Radius. Placement Angle sera choisi automatiquement par l'algorithme de calcul).
6. Cliquer sur **Add Work Region** et sélectionner la région sur laquelle l'extraction de cylindres doit être effectuée.
7. Régler le curseur de défilement situé sous le bouton **Extract Cylinders** à la valeur voulue. Celui-ci permet de modifier un des paramètres influant sur l'extraction des cylindres. Une valeur faible risque d'entraîner la séparation d'un cylindre réel en plusieurs sous-cylindres que l'algorithme n'arrivera pas à reconnaître comme faisant partie d'une même entité (figure A.6). Une valeur trop forte risque d'entraîner l'intégration de certaines faces dans un cylindre, alors que celles-ci n'en font pas partie (figure A.7). Si la région jonction a préalablement été séparée, une valeur de 100% fonctionne en général bien.
8. Cliquer sur **Extract Cylinders** pour lancer le calcul d'extraction des cylindres. Une fois le calcul terminé, les nouvelles régions créées s'affichent dans de nouvelles couleurs (si l'option **Color Grey** est désactivée).

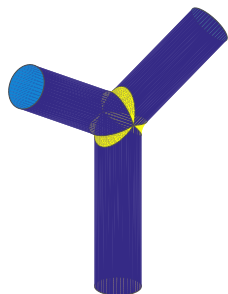


Figure A.4 Segmentation avec l'option **Segmentation By Edge Length**

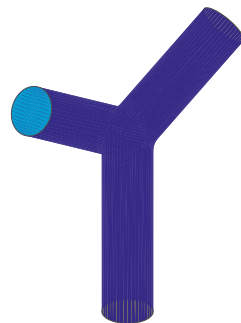


Figure A.5 Segmentation sans l'option **Segmentation By Edge Length**

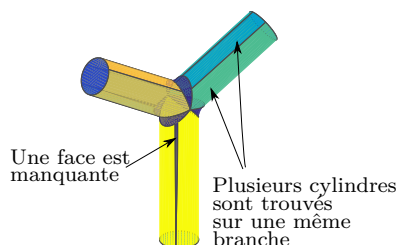


Figure A.6 Extraction des cylindres avec un seuil à 0% (à partir d'une segmentation sans l'option **Segmentation By Edge Length**)

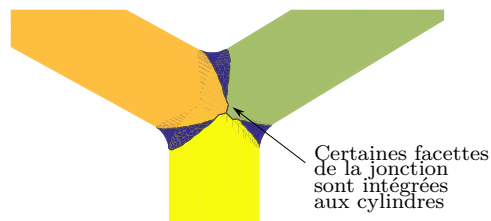


Figure A.7 Extraction des cylindres avec un seuil à 100% (à partir d'une segmentation sans l'option **Segmentation By Edge Length**)

La suite de la procédure dépend de l'algorithme choisi pour calculer les trajectoires. Quatre choix sont possibles dans ce programme, correspondant aux quatre algorithmes présentés dans ce mémoire :

1. **Discontinuous Helix (HD).**
2. **Adjustable Helix (HA).**
3. **Constant Placement Angle (CPA).**
4. **Maximum Curvature Constraint (MCC).**

Les procédures correspondant à chacun de ces algorithmes sont décrites dans les paragraphes A.4.3 à A.4.6.

A.4.3 Discontinuous Helix (HD)

Cet algorithme permet de couvrir une branche du « Y » avec une trajectoire en forme d'hélice, coupée au niveau de la jonction (voir section 5.3).

1. Cliquer sur **Add Work Region** puis cliquer sur la branche sur laquelle doit être calculée la trajectoire dans l'aire de visualisation.
2. Cliquer sur **Generate Path**, une fenêtre de sélection s'ouvre. Choisir la première option : **Discontinuous Helix (HD)**.
3. Une deuxième fenêtre de dialogue s'ouvre et propose d'utiliser un point de départ calculé automatiquement. Cliquer sur **Yes** pour choisir le point affiché en rouge dans l'aire de visualisation. Cliquer sur **No** pour sélectionner un autre point. Si **No** est choisi, se référer à la section A.4.1, point 8, pour la sélection du point de départ (Attention dans ce cas il n'est pas demandé de cliquer sur **Starting Point**, le processus de sélection est lancé automatiquement).
4. Une fois le point de départ choisi, le calcul de la trajectoire est lancé. Le nombre de points calculés sur la trajectoire s'affiche en direct dans la boîte d'information. Le nombre total de points créés et le temps de calcul s'affichent dans cette même boîte une fois que le calcul est terminé.
5. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
6. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).
7. Fermer le programme de génération de trajectoire.

A.4.4 Adjustable Helix (HA)

Cet algorithme permet de couvrir une branche du « Y » avec une trajectoire en forme de pseudo-hélice, ajustée au niveau de la jonction (voir section 5.4).

1. Cliquer sur **Add Work Region** puis cliquer sur la branche sur laquelle doit être calculée la trajectoire dans l'aire de visualisation.

2. Cliquer sur **Generate Path**, une fenêtre de sélection s'ouvre. Choisir la deuxième option : **Adjustable Helix (HA)**.
3. Une deuxième fenêtre de dialogue s'ouvre et propose d'utiliser un point de départ calculé automatiquement. Cliquer sur **Yes** pour choisir le point affiché en rouge dans l'aire de visualisation. Cliquer sur **No** pour sélectionner un autre point. Si **No** est choisi, se référer à la section A.4.1, point 8, pour la sélection du point de départ (Attention dans ce cas il n'est pas demandé de cliquer sur **Starting Point**, le processus de sélection est lancé automatiquement).
4. Une fenêtre de dialogue s'ouvre, il est possible de choisir le sens d'enroulement de l'hélice : Clockwise ou Counterclockwise (resp. Horaire et Antihoraire).
5. Une trajectoire est alors proposée et affichée dans l'aire de visualisation. Une fenêtre de dialogue s'ouvre. Cliquer sur **Yes** si la trajectoire convient, sur **No : change parameters** pour modifier les paramètres d'ajustement utilisés dans le calcul de cette trajectoire.
6. Si **No : change parameters** est choisi, les paramètres ajustables sont (voir section 5.4) :
 - (a) **Nb Lap** : le nombre de tours de l'hélice.
 - (b) **Amplitude** : l'amplitude de la déformation.
 - (c) **Period** : période de la déformation (utile s'il y a plus de trois branches).
 - (d) **Phase** : déphasage de la déformation, utile pour modifier la position de la déformation.
 - (e) **Direction** : 1 ou -1 permet de gérer le sens dans lequel l'hélice est calculée à partir du point de départ (vers l'intersection des cylindres ou dans l'autre direction).

Valider en cliquant sur **Ok**. Retour au point 5.

7. Une fois la trajectoire proposée acceptée, elle est interpolée afin d'obtenir la bonne distance entre les points, puis projetée sur la surface réelle de la pièce.
8. Une fois le calcul terminé une fenêtre de dialogue s'ouvre. On peut alors choisir de générer une deuxième passe sur cette même branche (**Yes**) ou terminer le calcul (**No**). Si **Yes** est choisi, une deuxième passe est générée partant du point diamétralement opposé à celui choisi en étape 3. Si **No** est choisi le calcul de trajectoire se termine.

9. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
10. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).
11. Fermer le programme de génération de trajectoire.

A.4.5 Constant Placement Angle (CPA)

Cet algorithme permet de générer des trajectoires dont les fibres ont une orientation constante sur les branches (voir section 4.4.1).

1. Cliquer sur **Add Work Region** pour sélectionner la branche de départ de la trajectoire.
2. Cliquer sur **Generate Path**, une fenêtre de sélection s'ouvre. Choisir la troisième option : **Constant Placement Angle (CPA)**.
3. Une fenêtre de dialogue s'ouvre, choisir l'angle de placement voulu (0°, 45° ou 90°)
4. Une fenêtre de dialogue s'ouvre demandant de choisir le point de départ. Deux choix sont possibles : le point rouge ou le jaune (calculés automatiquement). Le choix d'un ou de l'autre va influencer la seconde branche sur laquelle se propageront les trajectoires.
5. Une fois ce choix effectué, le calcul de la trajectoire commence. Celui-ci est relativement long et peut durer plusieurs minutes. Il est possible de visualiser en direct la création des points de la trajectoire si l'option **Dynamic Display** est cochée. Attention, activer cette option ralenti considérablement le calcul.
6. Lorsque le calcul est terminé, le message **Done** s'affiche dans la boîte de dialogue.
7. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
8. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).
9. Fermer le programme de génération de trajectoire.

A.4.6 Maximal Curvature Constraint (MCC)

Cet algorithme permet de générer des trajectoires respectant la courbure géodésique maximale autorisée par le procédé d'*AFP* (voir section 4.4.2) puis de sélectionner parmi celles-ci celles respectant certains critères (section 4.4.3), définis au point i en 3(a).

1. Cliquer sur **Generate Path**, une fenêtre de sélection s'ouvre. Choisir la quatrième option : **Maximal Curvature Constraint (MCC)**.
2. Une fenêtre s'ouvre proposant de choisir la forme que doit prendre la partie finale de la trajectoire (Voir section 4.4.2) :
 - **Natural** : trajectoire de courbure géodésique la plus faible possible.
 - **Reach Cylinder Axis - Helix** : forcer la trajectoire à finir en ligne droite (parallèle à l'axe du cylindre) ou en hélice dont le pas est la largeur de la bande de pré-imprégné déposée (le choix entre ces deux options s'effectuera en fonction de la manière dont la trajectoire arrive sur la seconde branche : si elle ressemble plus à une ligne droite, l'algorithme la fera tendre vers une ligne droite, sinon ce sera vers une hélice).
3. Une fenêtre s'ouvre proposant de choisir la méthode de sélection :
 - (a) **Full Discretization** : génère un ensemble de trajectoires puis choisit celles correspondantes aux critères fixés par l'utilisateur au point i. En sélectionnant cette option, les étapes suivantes s'offrent à vous :
 - i. Une fenêtre s'ouvre permettant de définir les critères de sélection des trajectoires :
 - **Clockwise** : nombre de trajectoires s'enroulant en sens horaire sur chaque branche de départ.
 - **Counterclockwise** : nombre de trajectoires s'enroulant en sens antihoraire sur chaque branche de départ.
 - **% Straight** : pourcentage de trajectoires finissant en ligne droite sur la seconde branche.
 - **% Helix** : pourcentage de trajectoires finissant en hélice sur la seconde branche.
 - ii. Une fois ce choix effectué, si une (ou un ensemble de) trajectoire(s) est déjà en mémoire, une fenêtre s'ouvre et propose de directement passer à la sélection

sur les trajectoires en mémoire. Si **Yes** est choisi, l'algorithme de sélection est lancé.

- iii. Si **No** est choisi une fenêtre s'ouvre demandant de choisir combien de points de départs seront à placer sur chaque branche pour générer l'ensemble de trajectoires sur lequel sera effectué la sélection. De chaque point de départ partira une trajectoire démarrant en sens horaire et une démarrant en sens anti-horaire. Cliquer sur **Ok** pour valider.
- iv. Le calcul de génération de trajectoires démarre.
- v. Lorsque ce calcul se termine, un tableau récapitulatif résumant les trajectoires voulues et les trajectoires obtenues s'affiche (figure A.8).
- vi. Si certaines trajectoires voulues n'ont pas été trouvées avec l'ensemble de trajectoires initial, une fenêtre s'ouvre proposant de raffiner (**Refine**) cet ensemble en calculant le double de trajectoires ou de garder la sélection actuelle (**Keep less layers**), qui ne correspond pas aux critères établis en étape i.

What we obtain:			
Starting in Counter-Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	2	22	19
Paths arriving in Helix on B2	22	1	19
Paths arriving in Helix on B3	21	22	5
Paths arriving in Straight Line on B1	0	4	8
Paths arriving in Straight Line on B2	4	0	9
Paths arriving in Straight Line on B3	11	11	0

What we obtain:			
Starting in Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	0	11	13
Paths arriving in Helix on B2	9	0	13
Paths arriving in Helix on B3	3	4	4
Paths arriving in Straight Line on B1	0	21	15
Paths arriving in Straight Line on B2	23	0	15
Paths arriving in Straight Line on B3	25	24	0

What we want:			
Starting in Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	0	0	0
Paths arriving in Helix on B2	0	0	0
Paths arriving in Helix on B3	0	0	0
Paths arriving in Straight Line on B1	0	5	5
Paths arriving in Straight Line on B2	5	0	5
Paths arriving in Straight Line on B3	5	5	0

What we want:			
Starting in Counter-Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	0	0	0
Paths arriving in Helix on B2	0	0	0
Paths arriving in Helix on B3	0	0	0
Paths arriving in Straight Line on B1	0	5	5
Paths arriving in Straight Line on B2	5	0	5
Paths arriving in Straight Line on B3	5	5	0

What we can have (Keep less layers option):			
Starting in Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	0	0	0
Paths arriving in Helix on B2	0	0	0
Paths arriving in Helix on B3	0	0	0
Paths arriving in Straight Line on B1	0	5	5
Paths arriving in Straight Line on B2	5	0	5
Paths arriving in Straight Line on B3	5	5	0

What we can have (Keep less layers option):			
Starting in Counter-Clockwise Helix From:	B1	B2	B3
Paths arriving in Helix on B1	0	0	0
Paths arriving in Helix on B2	0	0	0
Paths arriving in Helix on B3	0	0	0
Paths arriving in Straight Line on B1	0	4	4
Paths arriving in Straight Line on B2	4	0	5
Paths arriving in Straight Line on B3	5	5	0

Figure A.8 Tableau récapitulatif des trajectoires voulues et obtenues

- (b) **Enter an angle** : génère une seule trajectoire démarrant d'un point situé à un angle donné (par rapport au repère local du cylindre) sur le cercle de base du

- cylindre de départ (nécessite de sélectionner un cylindre de départ avec **Select Work Region**).
- i. Une fenêtre s'ouvre demandant de définir l'angle du point de départ et le sens de l'hélice calculée sur la branche de départ (horaire ou anti-horaire). Entrer les valeurs voulues et cliquer sur **Ok** pour valider.
 - ii. Le point de départ calculé est affiché en rouge. Une fenêtre s'ouvre demandant de valider ce choix. Cliquer sur **No** pour revenir à l'étape précédente, sur **Yes** pour passer à la suivante.
 - iii. Le calcul de la trajectoire s'effectue. Le message **Done** s'affiche dans la boîte de dialogue lorsque le calcul est terminé.
4. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
 5. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).
 6. Fermer le programme de génération de trajectoire.

A.5 Calcul de trajectoires sur l'intersection des cylindres

Ce programme offre également la possibilité de calculer des trajectoires sur l'intersection des cylindres formant les branches d'une pièce en forme de « Y ». Cet algorithme a été généralisé pour être capable de traiter des pièces formées d'un nombre arbitraire de branches. La démarche à effectuer pour réaliser ce calcul est la suivante :

1. Ouvrir MATLAB. Se placer dans le répertoire **AFP_PathPlanning**. Exécuter la commande **interface** dans l'invite de commandes pour lancer le programme. La fenêtre présentée en figure A.2 s'ouvre.
2. Dans le menu **Options** vérifier que l'option **Y-Shaped Part** est bien cochée.
3. Cliquer sur **Import STL** et choisir le fichier à ouvrir dans l'explorateur Windows qui s'est ouvert (dossier **CAO**). Une fois la pièce chargée, elle s'affiche dans l'aire de visualisation.
4. Cliquer sur **Segmentation**. La pièce est divisée en régions séparées par des angles vifs. Il est possible de régler le seuil de segmentation dans la case **Segmentation**

Threshold (deg), celui-ci correspond à l'angle maximum autorisé entre deux facettes voisines appartenant à une même région (par défaut 20°). Si l'option **Segmentation By Edge Length** est cochée, les facettes ayant de petits côtés (jonction) sont également regroupées dans une même région (figure A.4). Si l'option **Color Grey** n'est pas activée, les différentes régions s'affichent en différentes couleurs.

5. Définir le paramètre **Step** pour la génération de la trajectoire (distance séparant deux points consécutifs de la trajectoire, par défaut : 3 mm).
6. Cliquer sur **Add Work Region** et sélectionner la région sur laquelle l'extraction de cylindres doit être effectuée.
7. Régler le curseur de défilement situé sous le bouton **Extract Cylinders** à la valeur voulue. Celui-ci permet de modifier un des paramètres influant sur l'extraction des cylindres. Une valeur faible risque d'entraîner la séparation d'un cylindre réel en plusieurs sous-cylindres que l'algorithme n'arrivera pas à reconnaître comme faisant partie d'une même entité (figure A.6). Une valeur trop forte risque d'entraîner l'intégration de certaines faces dans un cylindre, alors que celles-ci n'en font pas partie (figure A.7). Si la région jonction a préalablement été séparée, une valeur de 100% fonctionne généralement bien.
8. Cliquer sur **Extract Cylinders** pour lancer le calcul d'extraction des cylindres. Une fois le calcul terminé, les nouvelles régions créées s'affichent dans de nouvelles couleurs (si l'option **Color Grey** est désactivée).
9. Cliquer sur **Inter Cyl** pour lancer le calcul de la trajectoire sur l'intersection des cylindres.
10. Lorsque le calcul est terminé, le nombre de points de la trajectoire est affiché dans la boîte de dialogue.
11. Pour afficher la trajectoire, utiliser le menu déroulant **Select a Path** (voir section A.2 point 16).
12. Pour sauvegarder cette trajectoire, cliquer sur **File** → **Save**. Dans l'explorateur Windows, entrer un nom de fichier. Celui-ci est enregistré dans le répertoire courant de MATLAB (**AFP_PathPlanning**).
13. Fermer le programme de génération de trajectoire.

A.6 Vérification des collisions

Le logiciel de planification de trajectoire proposé dans ce travail permet également de vérifier s'il existe des collisions entre l'outil et le moule sur lequel sont définies les trajectoires. Une option a été ajoutée permettant de modifier l'orientation de l'outil si une collision est détectée, de manière à l'éviter. Attention cependant, la force de pression du rouleau ne sera alors plus normale à la surface de la pièce. Une fois qu'une trajectoire a été générée sur une pièce, les étapes pour réaliser la vérification de collisions sont les suivantes :

1. Cliquer sur **File** → **Import Tool** → **CAO** → **outil2.stl** pour charger le fichier CAO de l'outil¹.
2. Une fenêtre s'ouvre demandant l'angle maximal d'inclinaison de l'outil (en degrés). Cela correspond à l'angle maximal d'inclinaison par rapport à l'orientation originale que l'on autorise à l'outil pour éviter une collision (par défaut : 0°, pas d'inclinaison, les collisions ne seront pas évitées, juste détectées).
3. Choisir une trajectoire à vérifier à l'aide du menu déroulant **Select a Path**.
4. Cliquer sur **Collision** pour lancer le calcul de détection des collisions. Si l'option **Dynamic Display** est activée il est possible de visualiser en direct les déplacements de l'outil pendant la vérification. Attention cela ralentit considérablement le calcul.
5. Une fois le calcul terminé, une fenêtre de dialogue mentionne le nombre de collisions trouvées et, dans le cas où des collisions ont été trouvées, propose de sauvegarder ou non la trajectoire calculée avec les nouvelles orientations permettant de les éviter. Cliquer sur **Yes** pour sauvegarder la trajectoire avec les nouvelles orientations, sur **No** pour abandonner les modifications.
6. Fermer le programme de génération de trajectoire.

A.7 Structure des données sauvegardées

Cette interface propose de sauvegarder les données générées lors du calcul de trajectoire (**Files** → **Save**). Ces données sont sauvegardées sous la forme d'un fichier MATLAB (.mat) contenant une structure MATLAB. Cette structure peut contenir trois entités :

1. Attention ce programme ne fonctionne qu'avec l'outil défini en section 5.8

1. **FVN** : structure contenant les données du maillage triangulé, ainsi que toutes les données relatives à ce maillage calculées par le programme. Utilisation : `FVN.prop`, avec `prop` la propriété du maillage que l'on souhaite afficher. Les propriétés enregistrées dans cette structure sont les suivantes :

- **XG**, **YG** et **ZG** : coordonnées des barycentres de chaque face.
- **faces** : liste des faces du maillage telles qu'elles sont sous la forme utilisée par la fonction `Patch` de MATLAB.
- **vertices** : liste des sommets du maillage sous la forme utilisée par la fonction `Patch` de MATLAB.
- **normals** : liste des normales aux faces du maillage.
- **Flab** : liste des faces du maillage contenant une colonne supplémentaire indiquant la région d'appartenance de la face.
- **voisinFace** : tableau contenant pour chaque face, la liste de ces voisins.
- **voisinFaceLC** : tableau contenant pour chaque face, la liste de ces voisines le long des deux plus grands côtés de la face.
- **nvert** : liste des normales aux sommets du maillage.
- **Surfacetot** : surface totale du maillage.
- **fichier** : nom du fichier CAO importé.
- **AreteFrontiere** : liste des arêtes modélisant les frontières entre régions.
- **AreteRegion** : liste des régions séparées par les arêtes décrites par la liste précédente.
- **LstArete** : liste de toutes les arêtes du maillage.
- **LstAreteBary** : liste des barycentres de toutes les arêtes.
- **NormReg** : structure contenant les normales aux sommets calculées pour chaque région individuellement.

2. **Cyl** : structure contenant les cylindres extraient ainsi que leur propriétés. Utilisation : `Cyl(i).prop`, avec i le numéro du cylindre et 'prop' la propriété désirée. Les propriétés enregistrées dans cette structure sont les suivantes :

- **faces** : liste des faces du maillage appartenant à ce cylindre.
- **Erreur** : Erreur dans l'approximation du cylindre (voir Tran et al. (2015) pour le calcul de cette erreur).

- **Rayon** : rayon du cylindre.
 - **Axe** : vecteur décrivant l'axe du cylindre.
 - **C** : point d'intersection des axes des cylindres
 - **label** : label définissant la région du cylindre.
 - **repere** : repère associé au cylindre.
 - **hauteur** : longueur du cylindre.
 - **FrontiereCercle** : liste des coordonnées des sommets formant la frontière circulaire du cylindre.
 - **FrontiereJunction** : liste des coordonnées des sommets formant la frontière entre le cylindre et la région jonction.
 - **FrontiereJhaut** : hauteur du point de la frontière avec la jonction étant le plus haut sur le cylindre (coordonnée z dans le repère local du cylindre).
 - **FrontiereJbas** : hauteur du point de la frontière avec la jonction étant le plus bas sur le cylindre (coordonnées z dans le repère local du cylindre).
3. **Traj** : structure contenant les données des trajectoires, celle-ci peut contenir un ensemble de trajectoires. Utilisation : $\text{Traj}\{i\}.\text{T}\{j\}\{k\}.\text{prop}$ avec "prop" la propriété désirée, i la branche sur laquelle démarrent les trajectoires, j le numéro de la trajectoire, et k le numéro du point sur cette trajectoire. Dans le cas d'une pièce quelconque, i et j seront toujours égaux à 1. Les propriétés enregistrées dans cette structure sont les suivantes :
- **p** : coordonnées du point de la trajectoire dans le repère absolu.
 - **n** : vecteur normal à la surface en ce point.
 - **f** : face du maillage contenant le point.
 - **angle** : angle de placement de la fibre (radian).
 - **pas** : pas utilisé pour l'algorithme itératif (équivalent à la distance entre les points de la trajectoire).
 - **outil_n** : vecteur normal de l'outil en ce point.
 - **outil_o** : vecteur d'orientation de l'outil en ce point.
 - **outil_a** : vecteur d'approche de l'outil en ce point.
 - **Repere** : repère associé à la pose de l'outil en ce point.

ANNEXE B CALCUL DES TRAJECTOIRES DANS L'ESPACE ARTICULAIRE DES ROBOTS

B.1 Séparation des trajectoires

Cette annexe décrit l'utilisation de la fonction `Methode_Jacobiene.m` permettant de réaliser la transformation d'une trajectoire Cartésienne vers les trajectoires correspondantes dans les espaces articulaires des robots de la cellule de travail (voir sections 4.6 et 5.9). Ce programme est écrit pour une cellule comprenant un manipulateur FANUC M20iA et une table rotative, il peut cependant être modifié pour être utilisé avec d'autres robots. Pour l'utiliser il faut se placer dans le sous-répertoire : **AFP_PathPlanning/Fonctions_InverseKine**. La fonction `Methode_Jacobiene.m` contient la définition du manipulateur utilisé (paramètres de Denavit-Hartenberg) de la table rotative ainsi que de la configuration de la cellule de travail. Elle fait également appel à d'autres fonctions assurant entre autre le calcul de la cinématique directe et inverse du manipulateur ainsi que des matrices jacobiniennes du manipulateur et de la table rotative. Les étapes à suivre pour réaliser la séparation des trajectoires sont les suivantes :

1. Exécuter la commande `Methode_Jacobiene.m` dans l'invite de commandes de MATLAB.
2. Une fenêtre s'ouvre. Sélectionner le fichier de trajectoire voulu (généré par le programme *interface*) et cliquer sur **ouvrir**.
3. Il est ensuite proposé dans l'invite de commandes de MATLAB de sélectionner la trajectoire sur laquelle on souhaite effectuer le calcul en indiquant son numéro. Celui-ci correspond au numéro utilisé dans le menu déroulant **Select a Path** du programme *interface* pour afficher la trajectoire (voir section A.2, point 7). Entrer le numéro voulu et appuyer sur **Entrer**.
4. Le premier point de la trajectoire est alors calculé. Les 8 différentes configurations du manipulateur possibles pour atteindre cette pose sont affichées dans l'invite de commandes sous la forme d'un tableau (figure B.1). Celui-ci contient 8 colonnes et 6 lignes correspondant aux 6 valeurs des articulations du manipulateur (en degrés). Les valeurs affichées en rouge correspondent aux valeurs dépassant les limites articulaires. Choisir la configuration voulue pour le calcul en entrant un chiffre entre 1 et 8 et valider en appuyant sur **Entrer**.

5. Le calcul s'effectue ensuite. Lorsqu'il est terminé, un graphique montrant l'évolution des valeurs des articulations s'affiche. Trois fichiers sont également créés (avec « filename » le nom du fichier initial contenant la trajectoire, choisi à l'étape 2) :
- filename.ls** : fichier .LS contenant le programme pour le manipulateur, importable dans ROBOGUIDE (voir annexe C.1).
 - filename.txt** : fichier texte contenant les valeurs articulaires des 7 articulations (les 6 du manipulateur et celle de table rotative) pour une utilisation dans RoboDK.
 - filename_theta.mat** : fichier MATLAB contenant les valeurs articulaires des 7 articulations (les 6 du manipulateur et celle de table rotative). Utilisable directement dans MATLAB.

1	2	3	4	5	6	7	8
-5.89	-5.89	-5.89	-5.89	174.11	174.11	174.11	174.11
49.19	49.19	162.69	162.69	-143.54	-143.54	-61.04	-61.04
-81.90	-81.90	16.68	16.68	138.92	138.92	212.32	212.32
-157.69	22.31	-54.80	125.20	148.26	-31.74	37.21	-142.79
-82.51	82.51	-27.43	27.43	-45.69	45.69	-38.49	38.49
-93.07	86.93	141.51	-38.49	113.36	-66.64	-120.74	59.26

entrer le numéro de la configuration à utiliser (1-8) :

Figure B.1 Tableau des modes d'assemblage du manipulateur

B.2 Simulations dans RoboDK

Afin de réaliser rapidement et facilement des simulations utilisant les trajectoires générées par la fonction `Methode_Jacobienne.m` (voir section B.1), une cellule de travail correspondant à celle définie dans cette dernière fonction a été modélisée grâce au logiciel RoboDK¹. Un script Python permet d'y importer le fichier texte de la trajectoire généré par la fonction `Methode_Jacobienne.m`. La procédure pour importer un fichier texte de trajectoire dans la cellule RoboDK est la suivante :

- Copier le fichier texte de trajectoire dans le sous-répertoire :
AFP_PathPlanning/Simulation.
- Ouvrir la cellule RoboDK en double cliquant sur le fichier **7DDL-M20iA.rdk** se trouvant dans ce répertoire.

1. <https://www.robodk.com/>

3. Il est possible de changer la pièce sur laquelle doit être déposée la fibre. Pour cela se référer à la section B.3.1.
4. Dans l'arbre des spécifications (figure B.2) effectuer un clic droit sur **Prog1**, puis sélectionner **Modifier programme Python**.
5. L'éditeur de programmes intégré à RoboDK s'ouvre alors (figure B.3). Sous la ligne « **# Ouverture d'un fichier en *lecture* :** » indiquer le nom du fichier texte à ouvrir entre guillemets dans la fonction `open`.
6. Sauvegarder le programme avec **Ctrl+S** ou **File → Save**.
7. Revenir dans la fenêtre de RoboDK. Double-cliquer sur **Prog1** pour lancer l'exécution du programme. Il est possible que le programme renvoie une erreur car il ne trouve pas le fichier texte de trajectoire. Dans ce cas, dans le script Python, enlever le commentaire de la ligne `#os.chdir("H :\AFP_PathPlanning\Simulation")` et remplacer le texte entre guillemet par le chemin du répertoire dans lequel se trouve la cellule de travail ouverte ainsi que le fichier de trajectoire.

Les deux robots vont alors se mettre en mouvement et la dépose de bandes de fibres sera simulée par l'affichage d'une ligne rouge sur la surface de la pièce.

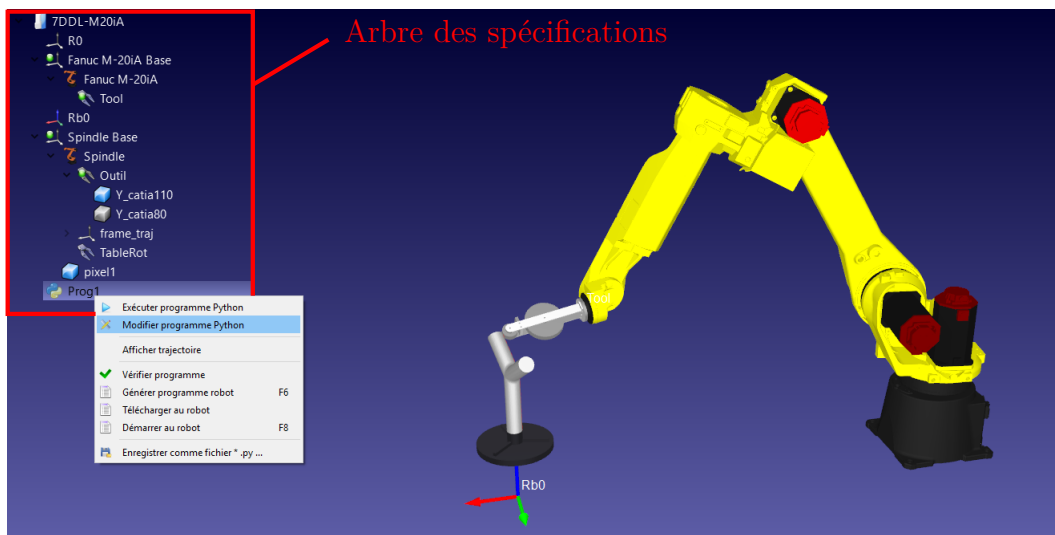
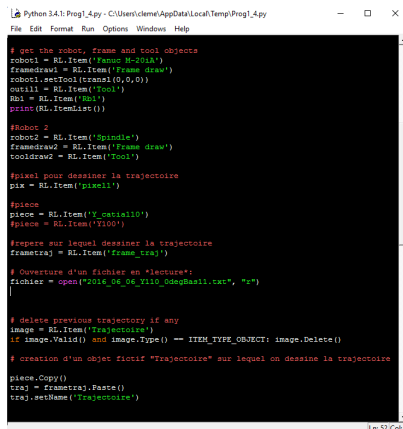


Figure B.2 Fenêtre RoboDK



```

Python 3.4: Prog1_Apy - C:\Users\dem\AppData\LocalTemp\Prog1_Apy
File Edit Format Run Options Windows Help

# get the robot, frame and tool objects
robot1 = RL.Item("Frame_Robot1")
framendrav1 = RL.Item("Frame_dreav")
robot1.setTool(framendrav1(0,0,0))
outil1 = RL.Item("Outil")
db1 = RL.Item("DB1")
print(RL.ItemList())

Robot 2
robot2 = RL.Item("Spindle")
framendrav2 = RL.Item("Frame_dreav")
toolndrav2 = RL.Item("Tool")

#Initial pour dessiner la trajectoire
pic = RL.Item("piece1")

#pièce
piece = RL.Item("Y_catia110")
#piece = RL.Item("Y100")

#repère sur lequel dessiner la trajectoire
frametraj = RL.Item("Frame_tra")

# Ouverture d'un fichier en "lecture"
fichier = open("C:\Users\dem\Desktop\110_060411.stp", "r")

# Delete previous trajectory if any
image = RL.Item("Trajectoire")
if image.Valid() and image.Type() == ITEM_TYPE_OBJECT: image.Delete()

# creation d'un objet fictif "Trajectoire" sur lequel on dessine la trajectoire
piece.Copy()
traj = frametraj.Paste()
traj.setName("Trajectoire")


```

Figure B.3 Éditeur de programme RoboDK

B.3 Modification de la cellule de travail

B.3.1 Changement de pièce

Il est possible de charger les fichier CAO des pièces sur lesquelles doit être déposé la fibre dans RoboDK. La procédure à suivre est la suivante :

1. Cliquer sur **Charger un fichier**, icône .
2. Dans l'explorateur Windows qui s'est ouvert, rechercher le fichier à importer (au format .stl, .iges ou .stp). Sélectionner le fichier et cliquer sur **Ouvrir**.
3. La pièce apparaît ensuite en bas de l'arbre des spécifications. Un double clic sur cet objet permet de faire apparaître les détails le concernant. En particulier on a accès à ses coordonnées dans l'espace (attention il est probable que l'objet importé ne sois pas visible car situé « dans » la base du manipulateur).
4. Faire apparaître les objets « Y_catia110 » et « Y_catia80 » en déroulant les menus **Spindle Base** → **Spindle** → **Outil** dans l'arbre des spécifications (figure B.4).
5. Afin de placer la nouvelle pièce sur la table rotative, cliquer-glisser cet objet sur **Outil**.
6. Il est possible d'afficher ou de cacher les pièces présentes dans **Outil**, en effectuant un clic droit sur la pièce concernée et cochant ou décochant **Visible**.
7. Avant de lancer une simulation, il est nécessaire de modifier le nom de la pièce dans l'éditeur de programme de roboDK. Ouvrir l'éditeur en effectuant un clic droit sur **Prog1** et en sélectionnant **Modifier programme Python**. A la ligne : **piece =**

`RL.Item('Y__catia110')`, remplacer le nom de l'ancienne pièce, situé entre les guillemets, par le nom de la nouvelle.

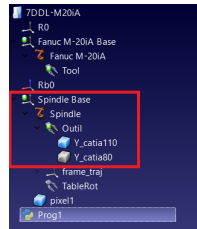
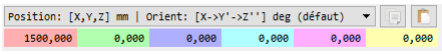


Figure B.4 Menus déroulant de l'arbre des spécifications

B.3.2 Déplacement des objets et changement de coordonnées

Cette section décrit comment déplacer les objets de la cellule de travail les uns par rapport aux autres. Le repère associé à la base du manipulateur est défini comme étant au centre de la cellule de travail (modélisée par le repère **R0** dans RoboDK). Le repère **Rb0** représente la base d'un support sur lequel serait posée la table rotative. Il est positionné par rapport à **R0**. Le repère associé à la table rotative (**Spindle Base**) est lui positionné par rapport à **Rb0**. Pour modifier la position et l'orientation d'un repère, la procédure est la suivante :

1. Double cliquer sur l'objet associé au repère à déplacer dans l'arbre des spécification.
2. Un volet s'ouvre sur la droite de la fenêtre, contenant les options de ce repère (figure B.5). Dans le menu déroulant **Position du repère par rapport à**, sélectionner le repère de référence par rapport auquel doit être positionner le repère à déplacer.

3. Entrer les nouvelles positions et orientations dans . Les trois premières case correspondent aux coordonnées x , y et z . Les trois suivantes sont les rotations autour des axes x , y' et z'' , avec y' l'axe du repère après qu'il ait subit la rotation autour de x et z'' l'axe du repère après qu'il ait subit la rotation autour de x et y' .

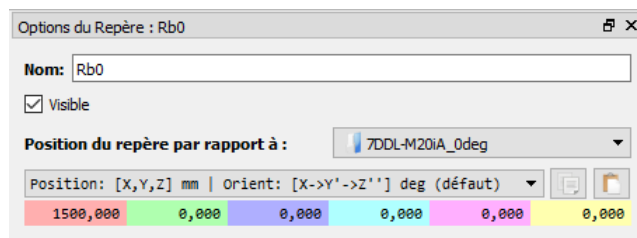


Figure B.5 Volet d'options du repère Rb0

ANNEXE C GÉNÉRATION ET STRUCTURE D'UN PROGRAMME .LS

C.1 Génération et structure d'un programme .LS

Cette annexe présente l'utilisation de la fonction MATLAB : `PostProcFanuc.m`, écrite pour ce travail, qui permet de générer les programmes .LS importables dans ROBOGUIDE. Nous présenterons ici brièvement leur structure. Les programmes générés par cette fonction restent basiques et permettent de déplacer le robot le long d'une trajectoire définie par une succession de poses. Les bases de la programmation FANUC peuvent être trouvées dans la documentation FANUC.¹

La fonction `PostProcFanuc.m(Pose, mode, modeInter)` écrit un fichier texte à l'extension .LS. Le nom d'enregistrement du fichier est demandé à l'utilisateur lors de son démarrage. Celle-ci prend en entrée trois arguments :

1. **Pose** : cet argument décrit la pose que le robot doit atteindre. Il peut se présenter sous deux formes :
 - (a) Représentation dans l'espace articulaire, $[J_1 \ J_2 \ J_3 \ J_4 \ J_5 \ J_6]$. J_1 à J_6 étant les valeurs articulaires du robot, en degré.
 - (b) Représentation dans l'espace Cartésien, $[x \ y \ z \ w \ p \ r]$. Les valeurs x, y, z étant les coordonnées du centre de l'outil (TCP) dans le repère global, en millimètre et w, p et r les angles d'Euler représentant son orientation, en degré. La convention utilisée par FANUC pour la définition des angles d'Euler est : xyz (soit $\mathbf{R} = \mathbf{R}_z(r) \times \mathbf{R}_y(p) \times \mathbf{R}_x(w)$, avec \mathbf{R} la matrice de rotation totale décrivant l'orientation de l'outil et $\mathbf{R}_i(k)$ la matrice de rotation d'un angle k autour de l'axe i).
2. **mode** : cet argument permet de spécifier laquelle des deux représentations précédentes est utilisée :
 - (a) Si `mode = 1`, le programme utilise la représentation dans l'espace articulaire.
 - (b) Si `mode = 2`, le programme utilise la représentation dans l'espace Cartésien.

1. Software Manuals → Operations → HandlingTool and MATE HandlingTool Setup and Operations Manual, Chapitre 8.

3. `modeInter` : cet argument permet de choisir le type d'interpolation que va effectuer le contrôleur du robot entre deux poses successives. Le contrôleur FANUC propose trois types d'interpolation :
- (a) `modeInter` = 1, interpolation linéaire dans l'espace articulaire.
 - (b) `modeInter` = 2, interpolation linéaire dans l'espace Cartésien. L'outil du robot effectue donc une ligne droite entre deux poses successives.
 - (c) `modeInter` = 3, interpolation circulaire dans l'espace Cartésien. Dans ce cas, deux poses doivent être spécifiées, celle d'arrivée et une intermédiaire par laquelle va passer le cercle.

Dans notre programme, l'argument `Pose` peut également se présenter sous forme d'une matrice $n \times 6$ contenant toutes les poses de la trajectoire (n étant le nombre de poses dans la trajectoire). Chaque pose sera alors lue et intégrée aux lignes d'instructions et de données du programme `.LS`. Un exemple d'un tel programme est donné dans la figure C.1.

La structure de ce programme est la suivante :

- Ligne 1 : Nom du programme.
- Ligne 2 : Déclaration des attributs du fichier.
- Lignes 3 à 20 : Attributs du fichier.
- Ligne 21 : Déclaration du commencement des instructions.
- Lignes 22 à 37 : Instructions, définition des repères.
- Lignes 38 à 39 : Instructions, déplacement du robot.
- Ligne 40 : Déclaration du commencement de la définition des poses.
- Lignes 41 à 52 : Définition des données des poses.
- Ligne 53 : Fin du programme.

Les lignes d'instructions générées par cette fonction ont toujours la même structure. On y retrouve le numéro de la ligne, suivi du mode d'interpolation, **J**, **L** ou **C**. **J** désignant une interpolation dans l'espace articulaire, **L** est utilisé pour l'interpolation dans l'espace cartésien et **C** pour l'interpolation circulaire. $\mathbf{P}[i]$ désigne ensuite la pose à atteindre et fait référence à la i^{eme} pose définie dans le bloc débutant par `/POS` (ligne 24 figure C.1). Le bloc


```

1  /PROG MonProgramme
2  /ATTR
3  OWNER          = MNEDITOR;
4  COMMENT        = "ClemSequence";
5  PROG_SIZE      = 0;
6  CREATE         = DATE 31-12-14 TIME 12:00:00;
7  MODIFIED       = DATE 31-12-14 TIME 12:00:00;
8  FILE_NAME      = ;
9  VERSION = 0;
10 LINE_COUNT     = 0;
11 MEMORY_SIZE    = 0;
12 PROTECT        = READ_WRITE;
13 TCD: STACK_SIZE = 0,
14     TASK_PRIORITY = 50,
15 TIME_SLICE     = 0,
16 BUSY_LAMP_OFF  = 0,
17 ABORT_REQUEST  = 0,
18 PAUSE_REQUEST  = 0;
19 DEFAULT_GROUP  = 1,*,*,*,*,*,*;
20 CONTROL_CODE   = 00000000 00000000;
21 /MN
22 1: UTOOL_NUM=1;
23 2: UFRAME_NUM=1;
24 3: PR[10,1]=0.000 ;
25 4: PR[10,2]=0.000 ;
26 5: PR[10,3]=0.000 ;
27 6: PR[10,4]=0.000 ;
28 7: PR[10,5]=0.000 ;
29 8: PR[10,6]=0.000 ;
30 9: UFRAME[1]=PR[10] ;
31 10: PR[10,1]=0.000 ;
32 11: PR[10,2]=0.000 ;
33 12: PR[10,3]=0.000 ;
34 13: PR[10,4]=0.000 ;
35 14: PR[10,5]=0.000 ;
36 15: PR[10,6]=0.000 ;
37 16: UTOOL[1]=PR[10] ;
38 17:J P[1] 50% CNT100 ;
39 18:J P[2] 50% CNT100 ;
40 /POS
41 P[1]{
42     GP1:
43     UF : 1, UT : 1,          CONFIG : ' F U T, 0, 0, 0',
44     X = 580.000 mm,        Y = 0.000 mm, Z = 180.000 mm,
45     W = 0.000 deg,         P = -90.000 deg, R = -180.000 deg
46 };
47 P[2]{
48     GP1:
49     UF : 1, UT : 1,          CONFIG : ' F U T, 0, 0, 0',
50     X = 580.000 mm,        Y = 0.000 mm, Z = 100.000 mm,
51     W = 0.000 deg,         P = -90.000 deg, R = -180.000 deg
52 };
53 /END

```

Figure C.1 Structure d'un programme .LS

50% désigne la vitesse de déplacement du robot. Dans ce cas, celui-ci se déplacera à 50% de la vitesse maximum de l'articulation la plus lente, toutes les articulations commenceront et finiront le mouvement au même moment. Il est possible de définir cette vitesse en terme de vitesse linéaire ou angulaire (mm/sec, cm/min, deg/sec) ou également de préciser la durée


du mouvement entre deux poses (ex : 3sec). Le dernier élément (CNT100) représente la précision avec laquelle le robot doit s'approcher de la pose définie. Deux arguments sont possibles, FINE ou CNTxx. FINE force le robot à s'arrêter exactement sur chaque pose de la trajectoire. CNTxx (pour continuous) permet de passer proche ou exactement (xx variant respectivement de 0 à 100) sur la pose voulue, en gardant un mouvement continu.

De nombreuses autres possibilités de programmations sont offertes permettant de réaliser des programmes plus complexes, et entre autres de commander les entrées/sorties du robot (ex : distributeurs pneumatiques). Celles-ci ne sont cependant pas implémentées dans la fonction présentée ici, qui se restreint à la génération d'une trajectoire. Le lecteur se référera donc au manuel FANUC sus-cité pour de plus amples informations.

ANNEXE D UTILISATION DU LOGICIEL ROBOGUIDE

Cette annexe présente une brève introduction à l'utilisation du logiciel de simulation ROBOGUIDE édité par la société FANUC. Nous verrons ici comment créer une cellule de travail et comment importer un programme au format .LS afin de le compiler en un fichier binaire (.TP) directement lisible par les contrôleurs FANUC.

D.1 Création d'une cellule de travail

1. Ouvrir ROBOGUIDE.
2. Créer d'une nouvelle cellule de travail¹ : **File** → **New Cell** ou . La cellule va être créée dans un dossier portant le nom défini à l'étape 'Step 2'. Ce dossier est créé dans le répertoire par défaut² et il contient tous les éléments de la cellule de travail.
3. Un assistant s'ouvre alors pour vous guider dans la sélection des options de la cellule :

Step 1 : *Process selection* : Détermine quels robots et options vont être disponibles dans les prochaines étapes. Permet de créer des cellules dédiées à un processus de fabrication (exemple : soudure à l'arc). Choisir dans notre cas : **HandlingPRO**.

Step 2 : *Workcell Name* : donner un nom à votre cellule de travail.

Step 3 : *Robot Creation Method* : utiliser l'option par défaut ; **Create a new robot with the default HandlingPRO config**.

Step 4 : *Robot Software Version* : sélection de la version du contrôleur. **V8.30 - R30iB, 8.30139.18.06**.

Step 5 : *Robot Application/Tool* : sélection des paquets d'outils à charger. Choisir **HandlingTool (H552)**.

Step 6 : *Group 1 Robot Model* : sélection du robot. Polytechnique → **LR Mate 200iD/7L**. Concordia → **M20iA**.

Step 7 : *Additional Motion Group* : sélection de robots supplémentaires à ajouter à la cellule de travail. Ne rien sélectionner et cliquer sur **Next**

1. Attention, cette étape peut conduire à un message d'erreur si le dossier d'enregistrement par défaut n'est pas accessible en écriture par ROBOGUIDE. Dans ce cas reportez vous à l'annexe D.6.


2. Pour savoir quel est le répertoire par défaut et comment le modifier se reporter également à la section D.6

Step 8 : *Robot Options* : sélection des options logicielles à charger. Laisser le choix par **défaut ; Ascii Programm Loader (R796), PC interface (R641)**.

Step 9 : *Summary* : Résumé de la configuration de la cellule qui va être créée. Cliquer sur **Finish** pour lancer la création de la cellule. Celle-ci est créée dans le dossier portant le nom défini à l'étape 'Step 2' situé dans le répertoire par défaut³.

Step 10 : Dans le cas du M20iA plusieurs options doivent être choisies pour le contrôleur dans la fenêtre d'initialisation du contrôleur (figure D.1).

- i. ROBOT TYPE SETTING : choisir l'option, **M-20iA (20kg mode)**.
- ii. CABLE DRESS-OUT TYPE SETTING : choisir la configuration correspondant à votre robot.
- iii. J1 MOTION RANGE SETTING : choisir la limite articulaire de la première articulation de votre robot.

Une fois cette première étape terminée, la cellule de travail contenant le robot s'affiche dans ROBOGUIDE (voir figure D.2) et il est possible d'afficher le TeachPendant virtuel en cliquant sur l'icône : . Cet outil permet d'accéder aux différentes fonctionnalités du robot. Il est par exemple possible de le faire bouger manuellement, de rédiger des programmes, de sélectionner et exécuter des programmes qui ont préalablement été chargés dans sa mémoire, etc. Le TeachPendant virtuel se comporte de la même manière que celui qui est associé physiquement au robot et permet donc de prendre en main cet instrument sans risque de dommages pour le robot réel.

3. Pour savoir quel est le répertoire par défaut et comment le modifier se reporter à la section D.6

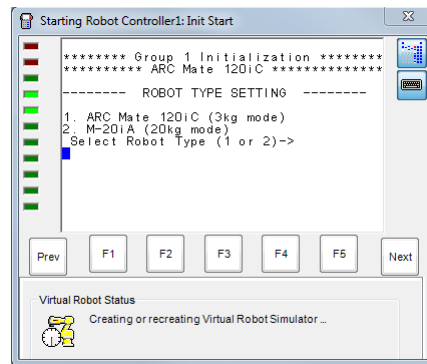


Figure D.1 Options pour le contrôleur du M20iA

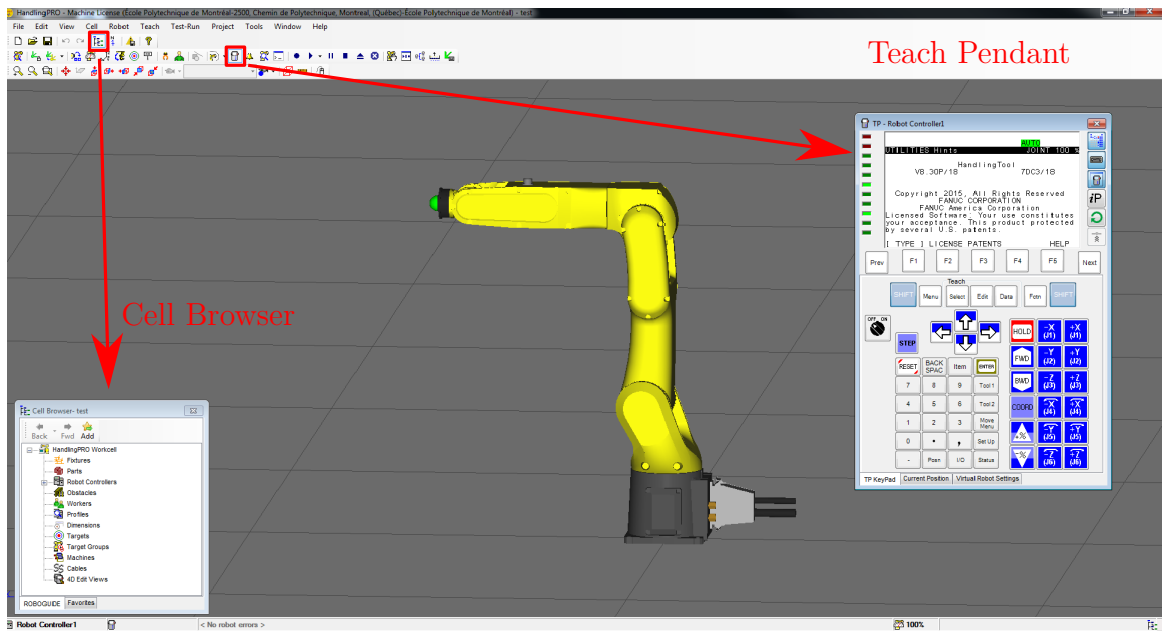



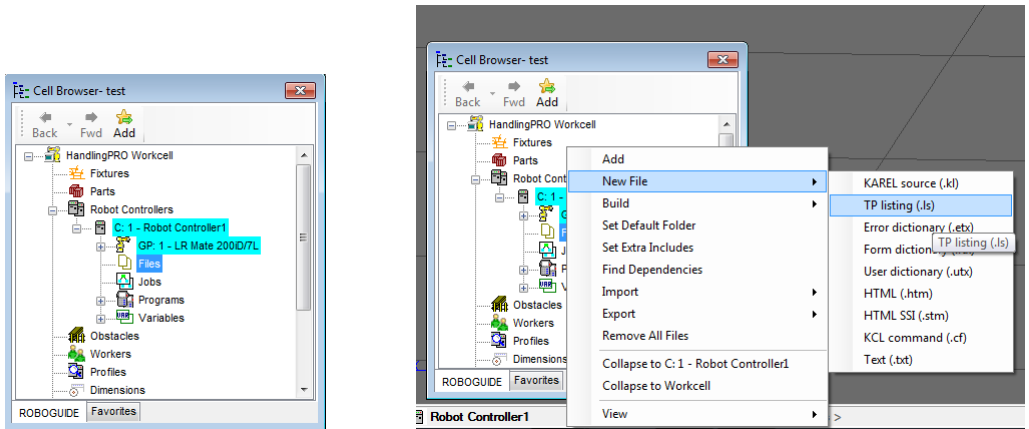


Figure D.2 Vue générale de la cellule de travail

D.2 Import de programme .LS

Nous allons voir dans ce chapitre comment importer un programme sous forme de fichier texte à l'extension .LS dans ROBOGUIDE. Celui-ci pourra ensuite être compilé en un fichier binaire .TP et chargé dans le contrôleur du robot FANUC.

1. Ouvrir une cellule de travail ROBOGUIDE.
2. Ouvrir le Cell Browser, icône  (voir figure D.2).
3. Dans le Cell Browser, développer **Robot Controllers** → **C : 1- Robot Controllers** → **Files** (figure D.3a).
4. Effectuer un clic droit sur **Files**, sélectionner **New File** → **TP listing (.ls)** (figure D.3b). Un éditeur de texte s'ouvre dans lequel il est possible d'écrire le programme.
5. Écrire ou copier-coller le programme voulu dans l'éditeur de texte (figure D.4). Sauvegarder ensuite ce programme dans le dossier contenant la cellule de travail, en cliquant sur **Save as**, icône . Attention le nom du fichier doit être le même que celui du programme (qui suit la commande /PROG à la première ligne). Veillez à bien indiquer l'extension du fichier (.ls) lors de la sauvegarde.
6. A la (aux deux) question(s) qui suit(vent), demandant si l'on souhaite remplacer l'ancien fichier, répondre **Yes**.
7. Cliquer sur **Build**, icône  pour compiler le fichier. Une fenêtre apparaît indiquant le statut de l'opération.
8. Une fois la compilation complétée, un nouveau fichier .TP apparaît dans le menu Files du Cell Browser. Ce programme est normalement utilisable directement depuis le Teach-Pendant virtuel.
9. Fermer la fenêtre de statut de la compilation.
10. Fermer l'éditeur de programmes.



(a) Développement des sous-répertoires

(b) Ajout du fichier .LS

Figure D.3 Cell Browser

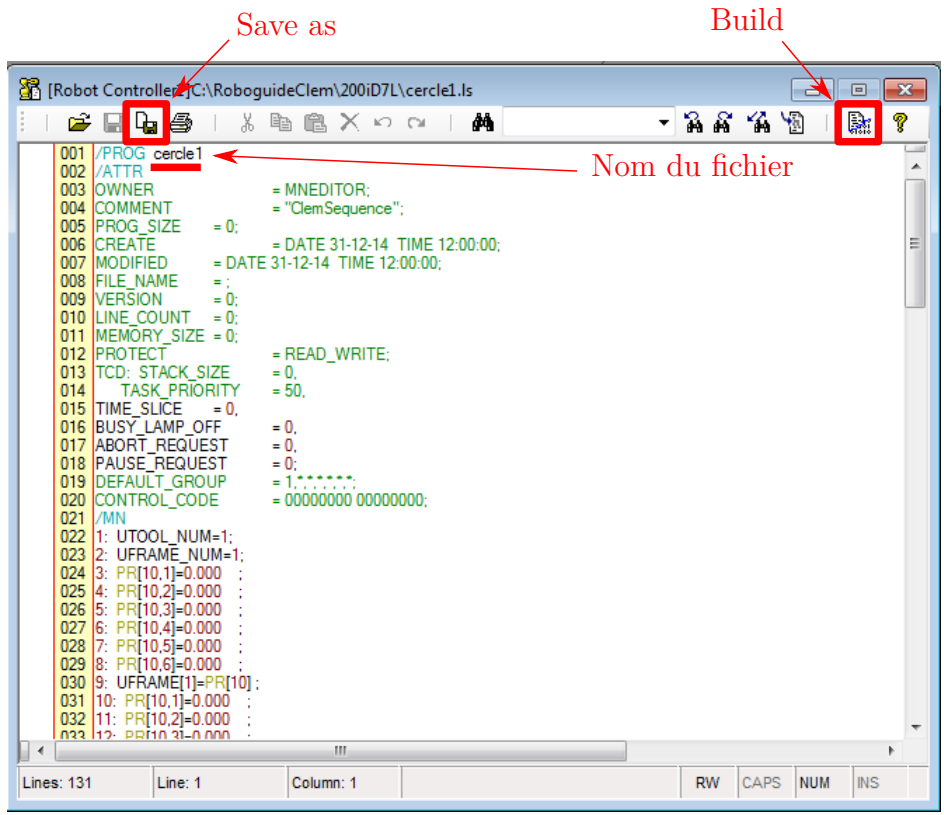




Figure D.4 Editeur de Programme de ROBOGUIDE

D.3 Exécution d'un programme depuis le TeachPendant virtuel



Voici les étapes à suivre pour exécuter le programme depuis le TeachPendant (se référer à la figure D.5 pour la localisation des différents boutons de l'interface) :

1. Dans une cellule de travail ROBOGUIDE, cliquer sur l'icône  pour ouvrir le TeachPendant virtuel (figure D.5).



2. Mettre le TeachPendant sur **ON** en cliquant sur l'icône .
3. Cliquer sur **Select**, pour afficher la liste des programmes en mémoire.
4. Utiliser les flèches directionnelles pour mettre votre programme en surbrillance.
5. Cliquer sur **ENTER** pour le sélectionner. Les lignes du programme correspondant aux instructions écrites dans le fichier .LS s'affichent à l'écran du TeachPendant.
6. Pour exécuter le programme dans son ensemble, cliquer une fois sur **SHIFT** (le bouton est considéré comme étant maintenu enfoncé) puis sur **FWD**.
7. Pour exécuter le programme pas à pas, cliquer sur **STEP**. Le 3^{ème} voyant (vert) en partant du haut s'allume sur le coté gauche du TeachPendant virtuel. A chaque appui sur **FWD** une seule ligne du programme est à présent exécutée.



8. Pour modifier la vitesse de déplacement du robot utiliser les boutons :  pour l'augmenter ou  pour la diminuer. La variation se fait en pourcentage de la vitesse des articulations (entre 1% et 100%). La vitesse actuelle s'affiche en haut à droite de l'écran du TeachPendant au moment de l'appui sur un de ces boutons.

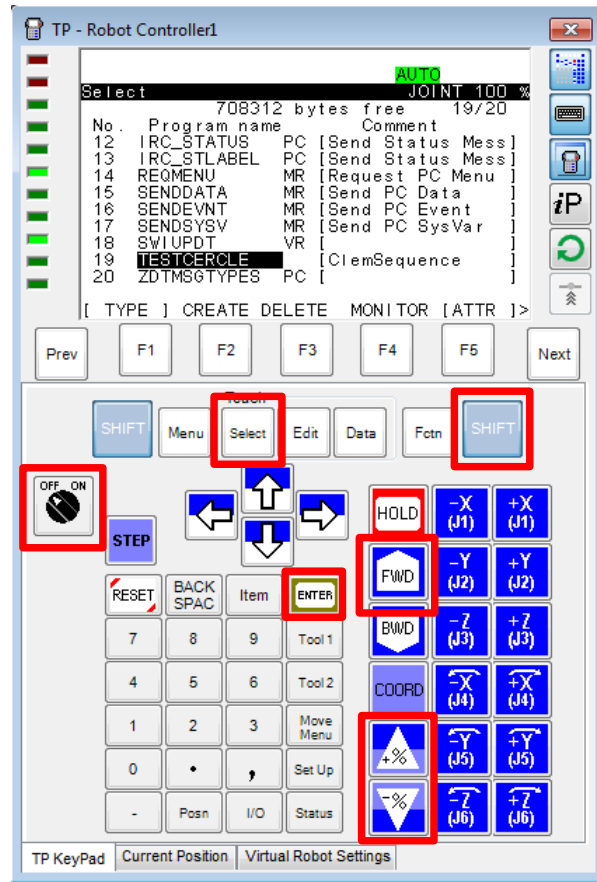


Figure D.5 TeachPendant

D.4 Chargement d'un programme .TP dans le TeachPendant depuis une clef USB

Lorsque le fichier .TP du programme est généré, il est possible de le charger dans le TeachPendant réel en utilisant une clef USB. Pour cela, copier ce programme sur votre clef USB (ROBOGUIDE peut être fermé) et insérer cette dernière dans le port situé sur le coté droit du TeachPendant, puis :

1. Appuyer sur le bouton **Menu** (figure D.6).
2. Sélectionner **File** grâce aux flèches directionnelles.
3. Cliquer sur **ENTER** pour afficher la liste des extensions des fichiers lisibles par le contrôleur. Le répertoire de recherche par défaut n'est en général pas celui de la clef USB.
4. Pour se placer dans le bon répertoire, cliquer sur **F5** (UTIL).
5. Sélectionner ensuite **Set Device** avec les flèches directionnelles, valider avec **ENTER**.
6. Sélectionner **USB on TP (UT1 :)** avec les flèches directionnelles, valider avec **ENTER**.
7. Sélectionner l'extension **8* TP** avec les flèches directionnelles, valider avec **ENTER**.
8. La liste des programmes .TP présents sur votre clef USB apparait. Sélectionner le programme voulu et cliquer sur **F3** (LOAD).
9. Valider votre choix avec **F4** (YES).
10. Un message de succès ou d'échec du chargement s'affiche à l'écran du TeachPendant (figure D.7).
11. Si le chargement a fonctionné, il est à présent possible d'exécuter le programme (se référer à la section D.5).

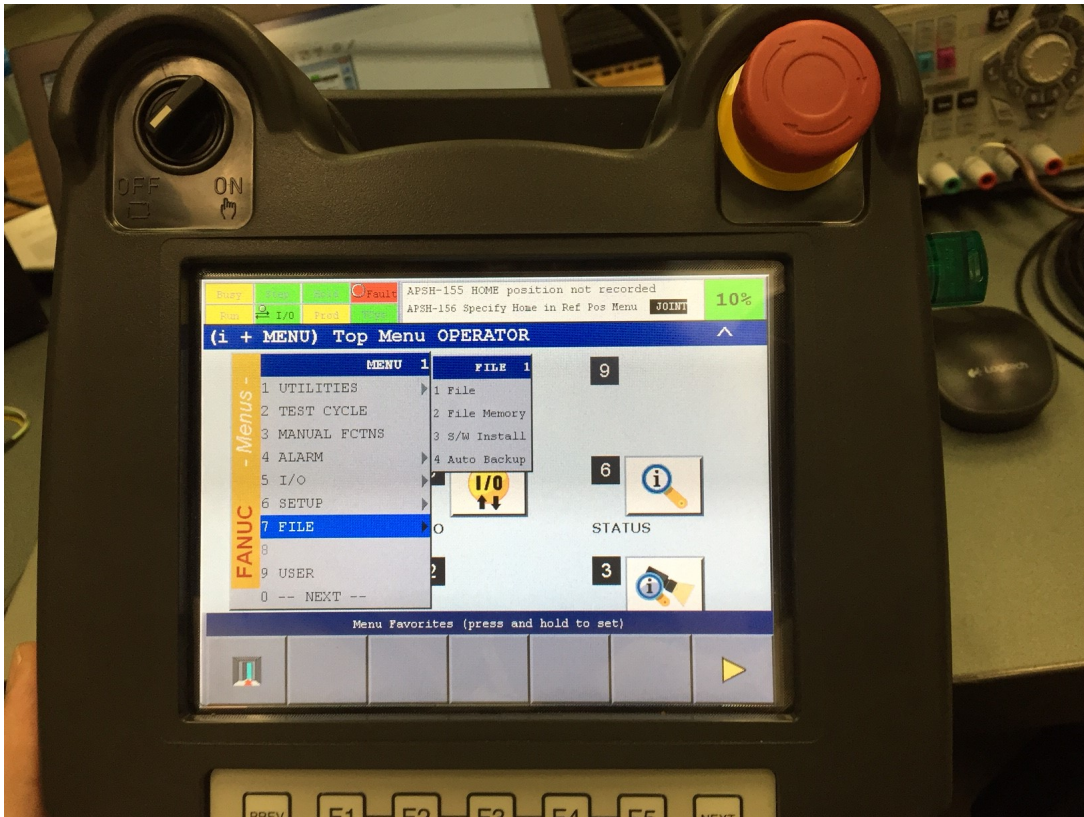


Figure D.6 Menu du TeachPendant

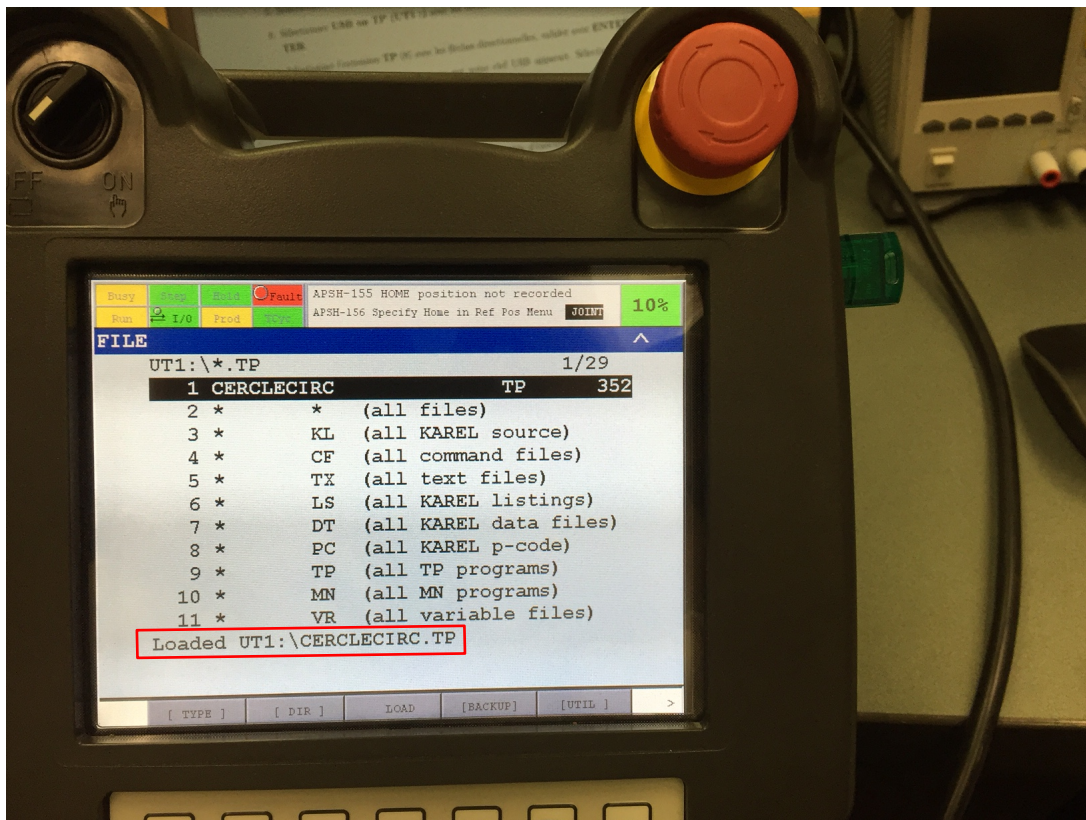


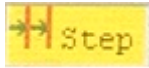
Figure D.7 Succès du chargement du programme TP dans le TeachPendant



D.5 Exécution d'un programme depuis le TeachPendant

Les étapes à suivre pour exécuter un programme depuis le TeachPendant réel sont globalement les mêmes que depuis le TeachPendant virtuel (section D.3). Les principales différences sont les étapes de sécurité.

1. Vérifier que les interrupteurs d'arrêt d'urgence du TeachPendant et du contrôleur ne sont pas enclenchés.



2. Mettre le TeachPendant sur **ON** en tournant le bouton.
3. Appuyer sur **Select**, pour afficher la liste des programmes en mémoire.
4. Utiliser les flèches directionnelles pour mettre votre programme en surbrillance.
5. Cliquer sur **ENTER** pour le sélectionner. Les lignes du programme correspondant aux instructions écrites dans le fichier .LS s'affichent à l'écran du TeachPendant.
6. Par mesure de sécurité, pour pouvoir exécuter un programme avec le robot réel, il faut maintenir en permanence enfoncé le **Dead Man Switch** (sur l'arrière du TeachPendant).
7. Généralement le contrôleur est en mode **Fault** (voyant rouge en haut de l'écran, voir figure D.7). Avant d'exécuter un programme il faut alors initialiser le contrôleur en maintenant enfoncé **SHIFT** et en appuyant simultanément sur **RESET**.
8. Pour exécuter le programme dans son ensemble, maintenir enfoncé **SHIFT** puis cliquer sur **FWD**. Si **SHIFT** ou le **Dead Man Switch** est relâché le robot stoppera toute exécution par sécurité. Si seul **SHIFT** a été relâché, l'exécution peut reprendre en appuyant sur **SHIFT** et **FWD**. Si le **Dead Man Switch** a été relâché, il est nécessaire d'effectuer un reset (étape 7).
9. Pour exécuter le programme pas à pas, appuyer sur **STEP**. L'icône STEP en haut à gauche de l'écran du TeachPendant devient : . A chaque appui sur **FWD** (avec **SHIFT** enfoncé) une seule ligne du programme est à présent exécutée.

10. Pour modifier la vitesse de déplacement du robot utiliser les boutons :  pour l'augmenter ou  pour la diminuer. La variation se fait en pourcentage de la vitesse des articulations (entre 1% et 100%). La vitesse actuelle s'affiche en haut à droite de l'écran du TeachPendant au moment de l'appui sur un de ces boutons.
11. En cas de blocage du programme, pour arrêter son exécution, appuyer sur **Fctn**, sélectionner **ABORT (ALL)** et cliquer sur **ENTER**.

D.6 Modification du dossier d'enregistrement par défaut

Pour fonctionner, ROBOGUIDE doit être autorisé à écrire dans le dossier sélectionné pour effectuer les sauvegardes. Si ce n'est pas le cas, il est possible de remédier à ce problème en changeant le dossier de sauvegarde par défaut. Choisir un dossier situé sur le disque local de la machine (plutôt qu'un emplacement réseau) permet en général de régler ce problème. Pour cela, cliquer dans le menu **Tools** → **Options**. Une fenêtre s'ouvre (figure D.8). Dans l'onglet **General**, définir le dossier de sauvegarde voulu dans **Default Workcell Path**. Valider en cliquant sur **OK**.

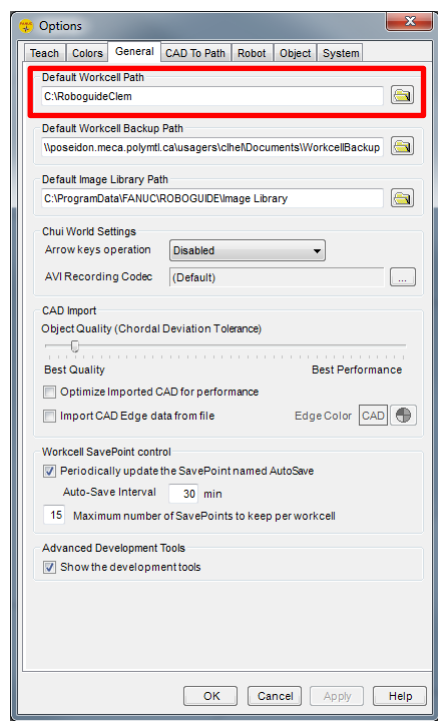


Figure D.8 Changement du dossier de sauvegarde par défaut