

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE LITTÉRATURE.....	3
1.1 Concepts de base.....	3
1.1.1 Précision.....	3
1.1.2 Calibration.....	4
1.2 Sources d'imprécision et limites de la calibration	4
1.3 Modélisation robotique	5
1.4 Amélioration de la précision par la calibration des paramètres cinématiques.....	6
1.4.1 Techniques de calibration issues de la littérature	6
1.5 Techniques de calibration industrielles.....	11
1.6 Techniques de calibration retenues pour les développements	11
1.7 Intégration technologique	12
1.8 Conclusion	13
CHAPITRE 2 MODÉLISATION ROBOTIQUE	14
2.1 Cinématique directe	14
2.2 Cinématique inverse.....	18
2.2.1 Cinématique inverse exacte	19
2.2.2 Cinématique inverse itérative	20
2.3 Librairie robotique MATLAB®	21
2.4 Conclusion	21
CHAPITRE 3 ALGORITHMES DE CALIBRATION ET OPTIMISATION	22
3.1 La linéarisation.....	22
3.1.1 Linéarisation premier ordre.....	22
3.1.2 Linéarisation second ordre.....	23
3.2 Constructions mathématiques à partir d'une contrainte planaire.....	23
3.2.1 Produits vectoriels.....	24
3.2.2 Plan moyen.....	26
3.3 Calcul des dimensions de l'outil	28
3.3.1 Sphère moyenne.....	29
3.4 Utilisation des résultats dans un contexte industriel.....	30
3.4.1 Modèle cinématique.....	30
3.4.2 Sphère moyenne.....	32
3.5 Conclusion	32
CHAPITRE 4 SIMULATIONS	33
4.1 Conditions générales de simulation	33
4.2 Méthode des produits vectoriels simplifiés (Zhong, Lewis et L.N.-Nagy, 1996)	35
4.3 Produits vectoriels non simplifiés.....	41

4.4	Calcul du plan moyen	49
4.5	Technique de calibration du TCP de l'outil.....	56
4.5.1	Résultats de simulation	56
4.5.2	Résultats de la mise en œuvre.....	58
4.6	Conclusion	63
CHAPITRE 5 CARACTÉRISATION DES COMPOSANTES DE L'ÉTUDE DE CAS		64
5.1	Robot IRB-140 d'ABB	64
5.1.1	Répétabilité et précision.....	65
5.1.2	Technique de calibration.....	66
5.1.3	Communication.....	67
5.2	Palpeur	67
5.2.1	Répétabilité et précision.....	69
5.2.2	Technique de calibration.....	69
5.2.3	Communication.....	69
5.3	Axes auxiliaires.....	70
5.3.1	Répétabilité et précision.....	71
5.3.2	Technique de calibration.....	77
5.3.3	Communication.....	77
5.4	Système de vision	78
5.4.1	Répétabilité et précision.....	79
5.4.2	Technique de calibration.....	81
5.4.3	Communication.....	90
5.5	Outil rotatif.....	90
5.5.1	Répétabilité et précision.....	91
5.5.2	Technique de calibration.....	91
5.5.3	Communication.....	91
5.6	Abrasifs.....	91
5.7	Conclusion	93
CHAPITRE 6 ÉTUDE DE CAS : ÉBAVURAGE ROBOTISÉ		94
6.1	Intégration	94
6.1.1	Architecture logicielle.....	96
6.2	Conditions générales de tests	99
6.2.1	Choix de la géométrie	99
6.2.2	Environnement.....	100
6.2.3	Trajectoire	102
6.2.4	Variables	104
6.3	Tests sans rotation de l'abrasif.....	106
6.3.1	Évaluation de l'influence des paramètres F, D et LP.....	106
6.4	Ébavurage robotisé.....	115
6.5	Conclusion	120
CONCLUSION		122
RECOMMANDATIONS		123

ANNEXE I	GRILLE DE CALIBRATION DE LA CAMÉRA	124
	LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES	125

LISTE DES TABLEAUX

		Page
Tableau 2.1	DHm du IRB 140 d'ABB	16
Tableau 4.1	Rang et conditionnement de la matrice M selon différentes configurations d'erreurs à retrouver.....	36
Tableau 4.2	Erreurs de modélisation induites et retrouvées par matrice inverse	38
Tableau 4.3	Erreurs de modélisation induites et retrouvées par optimisation, en utilisant la fonction <i>lsqnonlin</i> et l'algorithme <i>Levenberg-Marquardt</i>	39
Tableau 4.4	Erreurs de modélisation induites et retrouvées par optimisation, en utilisant la fonction <i>fminsearch</i>	40
Tableau 4.5	Erreurs de modélisation induites pour tester l'influence de la linéarisation sur les positions estimées du robot.....	42
Tableau 4.6	Normes de l'erreur induite par la linéarisation de premier et second ordre dans l'équation (3.5).....	43
Tableau 4.7	Vecteurs (<i>Voir</i> Équation (3.5)) issus des positions réelles et estimées à partir du vecteur d'erreur initial et de la linéarisation de premier, puis de second ordre.....	44
Tableau 4.8	Normales (<i>Voir</i> Équation (3.6)) issues des positions réelles et estimées à partir du vecteur d'erreur initial et de la linéarisation de premier, puis de second ordre	45
Tableau 4.9	Exemple d'amélioration de la précision par optimisation de l'équation (3.12).....	46
Tableau 4.10	Exemple d'échec de l'algorithme d'optimisation avec l'équation (3.12)..	48
Tableau 4.11	Exemple d'amélioration de la précision par optimisation de l'équation (3.17).....	50
Tableau 4.12	Exemple d'échec de l'algorithme d'optimisation avec l'équation (3.17)..	52
Tableau 4.13	Exemple d'erreur sur un seul paramètre de longueur	55
Tableau 4.14	Exemple d'erreur sur un seul paramètre de longueur	56
Tableau 6.1	Sommaire des caractéristiques de répétabilité et précision des éléments de la cellule	96

LISTE DES FIGURES

		Page
Figure 1.1	Instrument de mesure fait sur mesure. Tirée de Vira et Estler (1990).....	8
Figure 1.2	Une des techniques proposée. Tirée de Gatla <i>et al.</i> (2007)	10
Figure 2.1	Robot ABB IRB 140. Tirée du site web d'ABB (2010).....	14
Figure 2.2	Modèle DHm du IRB 140 d'ABB.....	15
Figure 3.1	Algorithme de calcul des poses à envoyer au robot selon les poses désirées et les erreurs de modélisation.....	31
Figure 4.1	Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.12).	47
Figure 4.2	Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.12).	49
Figure 4.3	Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.17).	51
Figure 4.4	Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.17).	53
Figure 4.5	Norme de l'erreur entre la position réelle du robot et celle estimée selon trois configurations différentes du robot.	54
Figure 4.6	Norme de l'erreur entre la position réelle du robot et celle estimée selon le modèle nominal pour l'erreur de modélisation présentée au Tableau 4.13.....	55
Figure 4.7	Représentation des touches simulées sur la sphère.....	57
Figure 4.8	Norme de l'erreur de positionnement sur une surface plane selon les dimensions nominales et les dimensions optimisées de l'outil.....	58
Figure 4.9	Palpeur lors de la calibration.....	59
Figure 4.10	Évolution des dimensions du TCP selon le nombre de touches utilisées pour l'algorithme de calibration.....	60
Figure 4.11	Plaque de verre utilisée pour l'évaluation des TCP.....	61

Figure 4.12	Évolution des dimensions du TCP selon le nombre de touches utilisées pour l'algorithme de calibration.....	62
Figure 5.1	Assemblage du capteur de force et changeur d'outil.....	65
Figure 5.2	Assemblage du palpeur Renishaw, de l'adaptateur et du <i>styli</i>	68
Figure 5.3	Capteur du palpeur.....	68
Figure 5.4	Montage des axes auxiliaires.....	70
Figure 5.5	Variation de la position du zéro de l'axe linéaire autour de la valeur moyenne.....	72
Figure 5.6	Variation du déplacement lors de la commande de déplacement relatif de 200 mm.....	73
Figure 5.7	Positions du zéro de l'axe linéaire.....	74
Figure 5.8	Positions après la commande de positionnement à 150 mm.....	75
Figure 5.9	Lecture du centre des trous lors de l'étude de répétabilité de l'axe rotatif.....	76
Figure 5.10	Variation de la position du centre des trous.....	77
Figure 5.11	Assemblage caméra et éclairage.....	78
Figure 5.12	Répétabilité du système robot-caméra-éclairage selon le nombre d'images prises.....	81
Figure 5.13	Identification du point central de la grille de calibration du TCP de la caméra.....	83
Figure 5.14	Calibration du focus de la caméra, image prise avant l'atteinte du point focal.....	84
Figure 5.15	Calibration du focus de la caméra, image prise très près du point focal.....	84
Figure 5.16	Calibration du focus de la caméra, image prise après l'atteinte du point focal.....	85
Figure 5.17	Calibration du centre de la caméra, image avant la calibration.....	86
Figure 5.18	Calibration du centre de la caméra, image après la calibration.....	86
Figure 5.19	Calibration de l'angle autour de l'axe Z de la caméra, image avant la calibration.....	87

Figure 5.20	Calibration de l'angle autour de l'axe Z de la caméra, image après la calibration.	88
Figure 5.21	Calibration de l'angle autour de l'axe X de la caméra, image durant la calibration.	89
Figure 5.22	Calibration de l'angle autour de l'axe Y de la caméra, image durant la calibration.	89
Figure 5.23	Outil rotatif.....	90
Figure 5.24	Abrasif 1.....	92
Figure 5.25	Abrasif 2.....	92
Figure 5.26	Adapteur pour abrasif.	93
Figure 6.1	Cellule de développement pour procédés robotisés.....	95
Figure 6.2	Schéma de la programmation du robot.	98
Figure 6.3	Petits trous sur les <i>balancing rims</i>	100
Figure 6.4	Cellule dans son environnement.	101
Figure 6.5	Rideaux pour le contrôle de l'éclairage.	101
Figure 6.6	Illustration des étapes pour l'ébavurage de petits trous sur les <i>balancing rims</i> , approche horizontale.....	102
Figure 6.7	Illustration des étapes pour l'ébavurage de petits trous sur les <i>balancing rims</i> , approche verticale.....	103
Figure 6.8	Effet du ratio d'amortissement et de la fréquence de coupure du filtre passe-bas sur la commande. Tirée de Application Manual, Force Control for Machining (ABB, 2005, p. 38)	105
Figure 6.9	Force et position selon l'axe Z pour 16 combinaisons de D et LP, avec $F = 5 \text{ N}$	108
Figure 6.10	Comportement particulier lors d'un aller-retour.....	109
Figure 6.11	Force et position selon l'axe Z pour 16 combinaisons de D et LP, avec $F = 10 \text{ N}$	110
Figure 6.12	Force et position selon l'axe Z pour 16 combinaisons de D et LP, avec $F = 15 \text{ N}$	111

Figure 6.13	Force et position pour une combinaison optimale de D et LP, avec $F = 5 \text{ N}$	113
Figure 6.14	Erreur en position selon les axes X et Y.....	114
Figure 6.15	Correction de l'erreur en position lors de l'application de la force désirée.....	115
Figure 6.16	Résultats de l'ébavurage robotisé avec $F = 5 \text{ N}$	117
Figure 6.17	Résultats de l'ébavurage robotisé avec $F = 3 \text{ N}$	118
Figure 6.18	Image d'un trou avant et après ébavurage.....	119
Figure 6.19	Résultats de l'ébavurage robotisé avec $F = 1.5 \text{ N}$	120

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

3D	trois dimensions
AI/O	analog input/output
CNC	ordinateur numérique
D	damping ratio
DDL	degré de liberté
DI/O	digital input/output
DH	Denavit-Hartenberg
DHm	Denavit-Hartenberg modifié
F	force
FCE	force change (start)
FCS	force change (end)
LP	low pass filter frequency
N/D	non-déterminé
P&WC	Pratt & Whitney Canada
RPM	rotation par minute
TCP	tool center point
V	vitesse
ZCF	zero contact force

LISTE DES SYMBOLES ET UNITÉS DE MESURE

UNITÉS GÉOMÉTRIQUES

Longueur

m	mètre
mm	millimètre ($= 10^{-3}$ m)
μm	micromètre ($= 10^{-6}$ m)

Angle

°	degré
---	-------

UNITÉS MÉCANIQUES

Vitesse

mm/s	millimètre par seconde
------	------------------------

Force

N	newton
N/s	newton par seconde (taux de variation)

UNITÉ DE MASSE

Masse

kg	kilogramme
----	------------

UNITÉ DE TEMPS

Fréquence

Hz	hertz
----	-------

INTRODUCTION

Le travail présenté dans ce document s'inscrit dans la lignée des recherches faites pour rendre plus accessible l'utilisation des bras robotisés dans les industries de haute précision comme l'aéronautique. En effet, ce domaine exige l'utilisation de techniques d'usinage, d'ébavurage, d'assemblage, de mesure et d'inspection démontrant une grande précision et une bonne flexibilité étant donné la variété de pièces produites à petit volume.

Les bras robotisés sont caractérisés par leur répétabilité, en plus d'offrir une bonne flexibilité à un coût qui est généralement plus faible que ceux d'autres systèmes permettant de réaliser les mêmes tâches. Ils sont donc très compétitifs par rapport aux commandes numériques par ordinateur (CNC). Par contre, la précision de ces bras robotisés n'est toujours pas à la hauteur des requis pour certaines tâches, comme l'ébavurage avec des abrasifs rigides.

Les travaux présentés ici permettent d'inventorier les différentes techniques permettant l'amélioration de la précision des bras robotisés par la calibration, d'étudier le potentiel de trois techniques nécessitant très peu de matériel et corriger l'impact de l'imprécision robotique dans une application d'ébavurage robotisé avec outillage rigide développée chez & Whitney Canada (P&WC). Cette application a nécessité l'intégration technologique d'un palpeur, d'un système de vision et d'un capteur de force pour arriver à démontrer la faisabilité de l'opération désirée. Le système complet a aussi été développé dans un environnement virtuel permettant la programmation hors-ligne des trajectoires devant être effectuées pour plus de flexibilité et pour répondre à la façon de faire chez P&WC.

Les éléments du système robotisé seront caractérisés et l'intégration de ceux-ci à la cellule robotique sera détaillée pour en arriver aux résultats de l'ébavurage d'une géométrie particulière avec des abrasifs rigides.

Le premier chapitre de ce travail est une revue de littérature sur l'amélioration de la précision robotique par la calibration. Il est entre autre question de plusieurs techniques de calibration qui sont commentées selon leur possible mise en œuvre lors de l'étude de cas.

Le second chapitre présente le modèle du robot utilisé lors des simulations et de l'étude de cas. Notamment, les matrices de transformées homogènes, la cinématique directe et la cinématique inverse exacte y sont présentées.

Le troisième chapitre explicite les différentes techniques de calibration qui seront simulées. Les développements mathématiques et les fonctions objectives qui seront utilisés pour permettre d'améliorer le modèle cinématique du robot y sont identifiés.

Le chapitre quatre porte sur les simulations, les conditions de simulation, les résultats et les analyses qui en ont découlé. Les conclusions de ce chapitre seront reprises dans les chapitres suivants qui traitent de l'étude de cas.

L'avant-dernier chapitre présente et caractérise les différents éléments qui feront partie de la cellule d'ébavurage robotisé. Les caractéristiques de précision et de répétabilité seront, entre autres, données pour ces éléments et les techniques de calibration particulières seront aussi décrites.

Le chapitre final présente les tests et résultats de l'étude de cas. L'environnement, la séquence des opérations et les différents paramètres à ajuster sont définis. Les valeurs optimales pour réaliser l'ébavurage robotisé avec un capteur de force sont identifiées. Les résultats de l'étude de cas sont ensuite présentés et analysés.

CHAPITRE 1

REVUE DE LITTÉRATURE

La première partie de ce travail fera la revue de ce qui s'est écrit et fait en terme de calibration robotique, autant pour la calibration que pour la mise en œuvre de différents systèmes. Le tout sera présenté de façon à mettre en évidence les choix qui ont été faits dans le reste de l'ouvrage.

1.1 Concepts de base

Comme nous traiterons principalement de calibration et de précision robotique, il est important de définir quelques termes reliés à ce domaine.

1.1.1 Précision

En robotique industrielle, il est pratique courante de quantifier une facette de la précision d'un robot en définissant sa répétabilité, soit la précision avec laquelle un robot peut retourner à une position enseignée (Craig, 2005). Cette caractéristique du robot réfère donc à un point où le robot a déjà été, c'est-à-dire à une configuration cinématique connue du robot.

D'autre part, la capacité d'un robot à se déplacer à une position précise dans l'espace en trois dimensions (3D) est connue sous le nom de justesse (Conrad, Shiakolas et Yih, 2000) ou *absolute accuracy*. Cette caractéristique du robot est beaucoup plus difficile à quantifier puisqu'elle varie beaucoup selon la portion du volume de travail du robot où elle est évaluée. Elle est donc généralement moins bien documentée pour les robots industriels, mais constitue un aspect crucial pour certaines applications dont celle qui sera présentée dans ce travail.

Finalement, le dernier concept relié à la précision robotique est la résolution, soit le déplacement minimum dont le robot peut garantir le mouvement (McKerrow, 1991). Cette caractéristique prend toute son importance lorsque le robot est guidé par un système externe.

1.1.2 Calibration

La calibration robotique est un terme appliqué à la procédure utilisée pour déterminer les valeurs réelles permettant de décrire les dimensions géométriques et les caractéristiques mécaniques d'un robot (Bernhardt et Albright, 1993). Dans notre cas, ces dimensions peuvent être les dimensions des membres, les zéros des articulations, les dimensions de l'outil utilisé, les masses et inerties, le frottement et la flexion des membres et articulations, les débattements des articulations, etc.

Pour calibrer un robot, quatre étapes sont nécessaires : modélisation, mesure, identification des paramètres et utilisation de l'information issue de la calibration (Mooring, Driels et Roth, 1991). Plusieurs exemples de techniques de calibration seront présentés plus bas.

Il est à noter que dans un contexte industriel, la calibration robotique peut aussi référer à la calibration de l'espace de travail par rapport au robot, à celle de la position de fixation ou directement de la position de l'objet sur lequel le robot travaillera ou celle du *tool center point* (TCP) utilisé, c'est-à-dire le point de travail de l'outil qui est monté à l'extrémité du robot.

1.2 Sources d'imprécision et limites de la calibration

Dans le cadre du présent travail, la justesse est l'aspect de la précision robotique qui doit être améliorée. Par contre, elle est difficilement quantifiable étant donné qu'elle est directement reliée à de multiples sources d'erreur de modélisation et au contrôle du robot. Plusieurs ouvrages définissent et classent les différentes sources d'erreur possibles (Bernhardt et Albright, 1993; Conrad, Shiakolas et Yih, 2000; Raucant, Samin et Gorez, 1991; Vira et

Estler, 1990). On retrouve généralement les deux catégories suivantes : les erreurs liées à la modélisation cinématique et celles liées à la modélisation dynamique du système robotique.

D'autre part, comme Craig (2005) et Caenen et Angue (1990) le mentionnent, la justesse intrinsèque d'un robot est limitée par sa répétabilité. C'est la capacité mécanique à reproduire une même position qui dicte la répétabilité du robot. C'est donc dire que malgré une modélisation parfaite de la structure robotique, les limites mécaniques limiteront la justesse de ce même robot.

Pour que la justesse du robot soit à la hauteur de la répétabilité, il faut que la modélisation du robot dans la boucle de contrôle (dans le contrôleur de mouvement du robot) ou dans le modèle logiciel (dans le cas de programmation hors-ligne) reflète exactement les caractéristiques physiques réelles du robot.

Renders *et al.* (1991) ont quantifié l'impact des différents types d'erreur sur la justesse en position du robot. Selon eux, près de 90% de l'erreur de positionnement est causée par l'erreur du modèle cinématique du robot. Nos efforts seront donc concentrés à améliorer la connaissance des longueurs des membres du robot et du zéro de ses encodeurs.

1.3 Modélisation robotique

Pour améliorer la modélisation cinématique du robot, il faut établir un premier modèle en se basant sur les dimensions nominales de ce même robot. Quelques types de modèles différents sont d'ailleurs proposés, la modélisation de Denavit-Hartenberg (DH) (Denavit et Hartenberg, 1955), Denavit-Hartenberg modifié (DHm) (Craig, 2005), l'ajout d'un paramètre β (Hayati et Mirmirani, 1985) pour la modélisation d'articulations parallèles consécutives ou même l'utilisation d'autres modèles moins répandus permettant de faciliter l'observabilité de certains paramètres (Abderrahim et Whittaker, 2000).

En ce qui concerne les modèles qui tiennent compte des paramètres dynamiques, ils seront utilisés avec un robot dont les boucles de contrôle sont non linéaires (Craig, 2005). Malheureusement, peu de contrôleurs industriels offrent la possibilité d'implémenter un tel type de contrôle. En effet, bien que ce type de contrôle soit beaucoup plus performant que le contrôle linéaire lorsque la vitesse augmente (Raucent, Samin et Gorez, 1991), plus de précautions sont nécessaires pour assurer la stabilité du système. La demande pour le contrôle non linéaire ne semble pas être suffisante pour que les fabricants de robots industriels s'y attaquent.

1.4 Amélioration de la précision par la calibration des paramètres cinématiques

Dans le présent ouvrage, l'objectif final est d'améliorer la précision d'un robot industriel fonctionnant avec un contrôle linéaire. Comme l'erreur de modélisation cinématique a un impact très important sur la justesse, c'est la calibration des paramètres cinématiques qui sera étudiée.

Telles que présentées précédemment, plusieurs techniques de modélisation sont établies dans la littérature. En ce qui concerne les méthodes pour déterminer les paramètres du robot, il en existe une variété utilisant divers modèles, équipements, concepts mathématiques et stratégies de mise en œuvre.

1.4.1 Techniques de calibration issues de la littérature

Une première stratégie est de créer une boucle fermée en fixant les extrémités du robot (base et effecteur) et en utilisant un capteur de force/moment entre l'extrémité du robot et le point de fixation. Par exemple, Bennet *et al.* (1992) fixent l'effecteur à un point dans l'espace cartésien par un capteur de force/moment. En faisant bouger les articulations libres du robot, il est possible de recueillir certaines informations sur ce dernier. Ces auteurs proposent essentiellement deux méthodes : la lecture des couples aux articulations et la lecture des dérivées des positions des articulations. Hollerback *et al.* (1993) présentent eux aussi une

technique très semblable. De leur côté, Liu *et al.* (1998) utilisent un capteur de force/moment, mais cette fois sous la base du robot pour retrouver les paramètres inertiels de chacun des membres du robot. Ces techniques ont comme inconvénients majeurs la rareté des points de fixation permettant d'avoir un nombre suffisant de mesures et présentent un danger pour l'intégrité du robot lors des manipulations.

Dans un autre ordre d'idées, certains auteurs ferment la boucle à l'aide d'instruments faits sur mesure. C'est le cas de Vira et Estler (1990) qui ont testé plusieurs outils de calibration et proposent une technique de calibration utilisant un outil fait sur mesure. L'instrument possède trois degrés de liberté (DDL) et l'extrémité est une sphère qui rend libre l'orientation (Voir Figure 1.1). Il l'utilise pour connaître l'erreur entre la position désirée et la position réelle du robot. Renders *et al.* (1991) utilisent le même genre d'instrument de mesure mais proposent l'utilisation d'un modèle combinant les paramètres cinématiques de la représentation DH et y ajoute celui proposé par Hayati et Mirmirani (1985). L'erreur entre les déplacements commandés et ceux observés peut donc être lue. Une minimisation des erreurs par maximum de vraisemblance avec une matrice jacobienne spéciale (Mirman et Gupta, 1992) permet de retrouver les paramètres réels du robot. Chacune de ces techniques présente un fonctionnement très simple, autant au niveau de la mise en œuvre que de l'algorithme utilisé pour retrouver les erreurs de modélisation. Par contre, elles exigent la fabrication d'un outil d'une très grande précision, ce qui entraîne des coûts supplémentaires considérables.

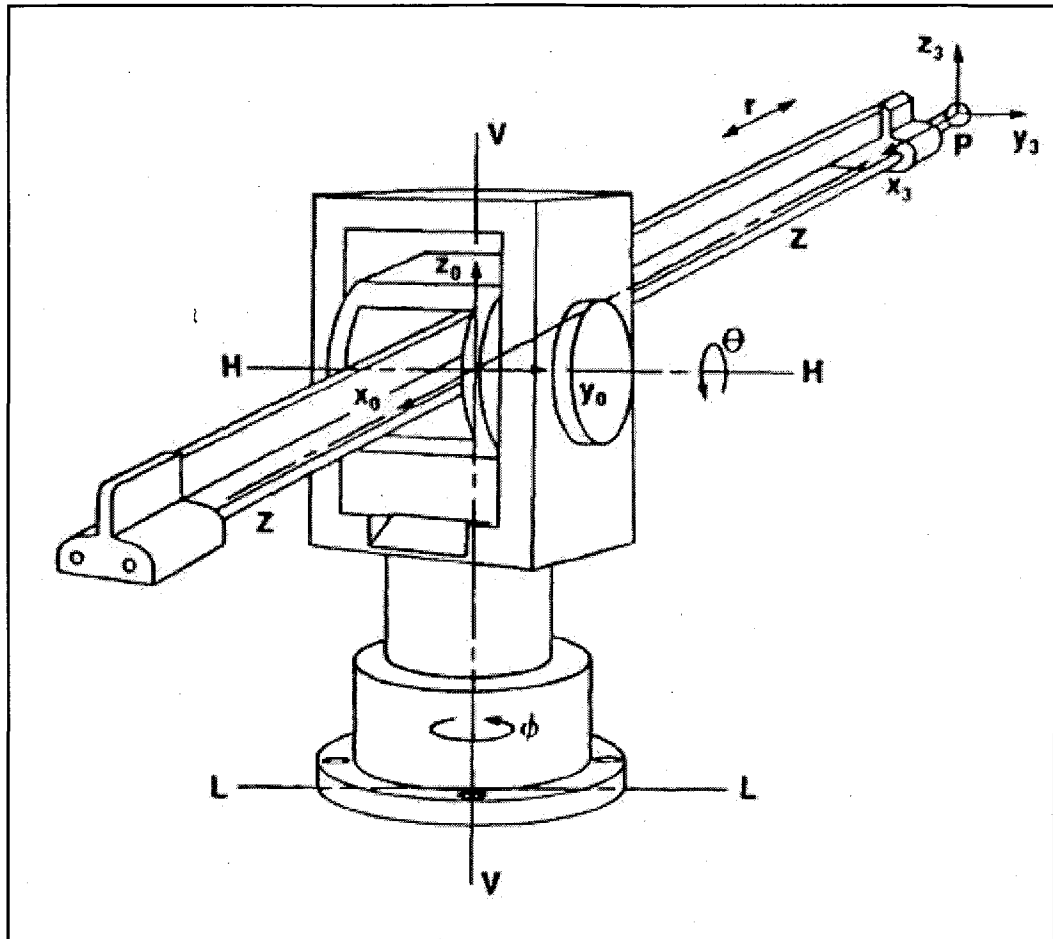


Figure 1.1 Instrument de mesure fait sur mesure.

Tirée de Vira et Estler (1990)

Une troisième façon de fermer la boucle est d'utiliser un capteur sans contact. Deux techniques présentées par Hollerback *et al.* (1993) utilisent des capteurs de positions et orientations sans contact. Elles sont mathématiquement semblables, la position de l'effecteur est trouvée, comparée avec celle calculée selon les valeurs nominales des paramètres puis une optimisation est faite pour retrouver les valeurs réelles des paramètres. D'autres auteurs, comme Caenen et Angue (1990) et Duelen et Schroer (1991), utilisent des théodolites. Dans leur cas, plusieurs poses sont essayées et l'erreur entre la position demandée et celle réelle est prise en note par le système. Une matrice jacobienne spéciale reliant les paramètres géométriques et non géométriques du robot à la position cartésienne est utilisée. Les paramètres sont simplement estimés par optimisation avec une fonction de moindre carré.

Les paramètres retrouvés sont utilisés dans un logiciel de simulation hors-ligne et la commande envoyée au robot permet le déplacement réel désiré. Malheureusement, les appareils permettant de capter la position et l'orientation du robot sans contact sont très dispendieux et peuvent aller jusqu'à deux, trois ou quatre fois le prix du robot. Une telle dépense n'est pas judicieuse pour une compagnie dont le but principal n'est pas la calibration de robots.

Pour diminuer les coûts reliés à la calibration, Motta *et al.* (2001) utilisent une caméra pouvant être positionnée directement sur le bras robotisé ou sur un support : la précision reste par contre très limitée avec ce type d'appareil. Sun *et al.* (2004) et Gatla *et al.* (2007) proposent aussi l'utilisation d'une caméra mais ajoutent au système un ou plusieurs lasers (Voir Figure 1.2). Bien qu'elles soient très intéressantes et peu coûteuses, ces techniques demandent beaucoup d'espace et un environnement contrôlé, ce qui est rarement possible dans un contexte industriel.

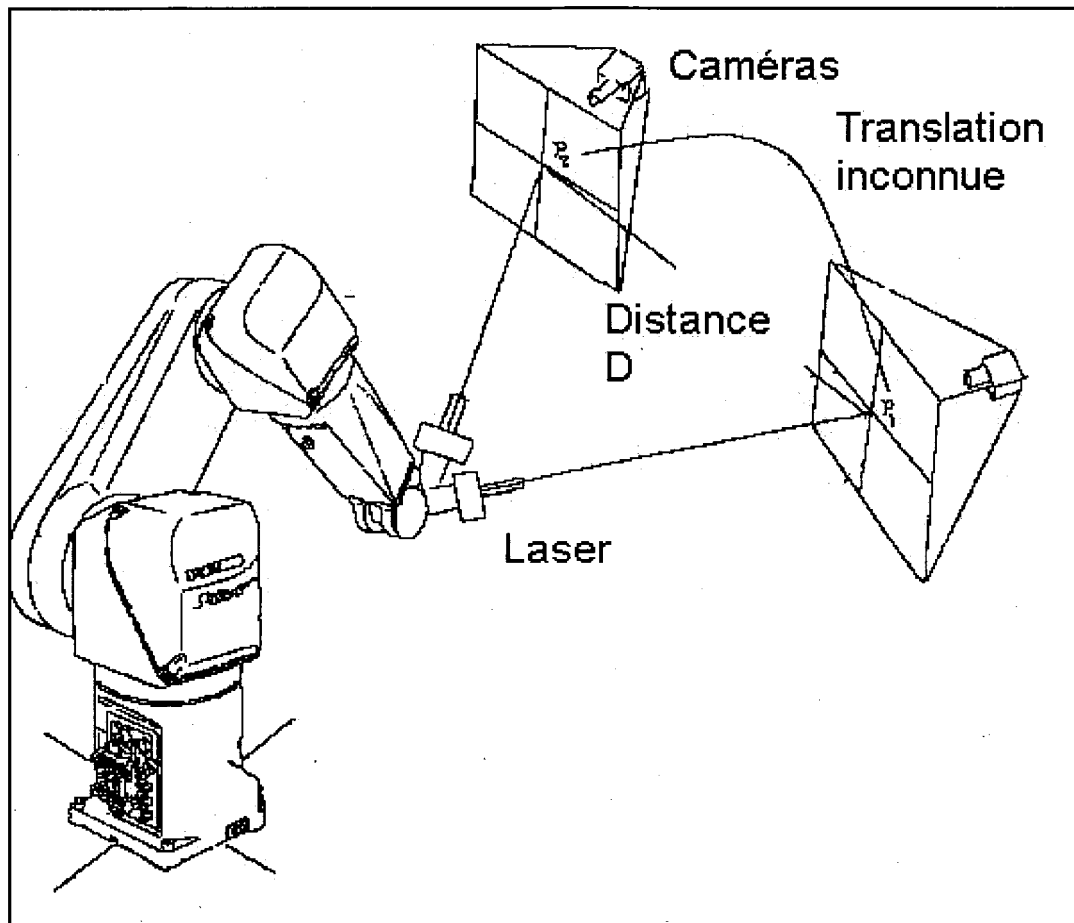


Figure 1.2 Une des techniques proposée.
Tirée de Gatla *et al.* (2007)

Finalement, Zhong *et al.* (1995; 1996) proposent de fermer la boucle en utilisant un palpeur et des contraintes planaires. Comme la position exacte n'est pas connue, l'optimisation doit se faire autour d'une fonction mathématique exprimant la contrainte. Les auteurs ont choisi d'utiliser les produits vectoriels. Khalil *et al.* (1996) et Ikits et Hollerbach (1997) proposent le même genre de contrainte, mais expriment la contrainte selon l'équation du plan. Hanqi *et al.* (1999) formulent le problème différemment en exprimant la position du robot en fonction du plan. Ils minimisent donc directement la distance entre chaque point et le plan. Malgré des difficultés notées au niveau de l'observabilité de certaines variables, ces techniques sont toutes indiquées pour un environnement industriel comme celui dans lequel la mise en œuvre sera faite.

Tel que mentionné au tout début de cette section, les paramètres cinématiques exacts du robot sont les éléments recherchés. Une approche plus simpliste serait de tenter de retrouver uniquement les paramètres cinématiques de l'outil qui sera mis à l'extrémité du robot. Effectivement, il est plus commun de calibrer le TCP de l'outil et généralement les fabricants de robots industriels proposent une technique permettant de le faire. Dans la littérature, plusieurs techniques et brevets concernent aussi ce type de calibration. Dans notre cas, nous considérerons ceux utilisant une sphère, comme nous en avons une sous la main (ABB, 1997; 2001; Wenbiao et al., 2009).

1.5 Techniques de calibration industrielles

Il est important de mentionner que les fabricants de robots offrent presque tous une calibration additionnelle du robot permettant une justesse accrue. Généralement, cette option est très dispendieuse et nécessite que le robot soit calibré avec un capteur de position et orientation sans contact, tel qu'utilisé par certains des auteurs présentés plus haut. En fait, le modèle du robot doit être ajusté en utilisant des algorithmes très semblables à ceux présentés par ces mêmes auteurs. Malgré tout, les robots n'ont, à ce jour, pas atteint les précisions nécessaires pour satisfaire toutes les exigences de l'industrie de l'aéronautique. C'est pourquoi l'ajout de capteurs devient nécessaire pour robotiser certaines applications.

1.6 Techniques de calibration retenues pour les développements

Dans le but de maximiser la précision des robots et de limiter l'emploi d'autres capteurs, nous tenterons de calibrer un robot utilisé pour l'ébavurage robotisé avec contrôle de force. Comme l'application visée en est une industrielle, les techniques n'utilisant aucun capteur externe sont des plus intéressantes. Aussi, les techniques contraignant le robot à faire ses touches sur un seul plan demandent peu d'équipement et sont donc réalisables à faibles coûts. De plus, des outils de mesures comme des marbres et des plaques dont la planéité est contrôlée sont disponibles à même le site de P&WC, via les départements d'inspection. Bref, les techniques présentées dans les articles de Zhong *et al.* (1995; 1996) ainsi que dans

l'article de Ikits et Hollerbach (1997) sont donc toutes indiquées. De plus, la calibration du TCP de l'outil faite avec une sphère sera aussi comparée à celle faite par la technique du fabricant.

1.7 Intégration technologique

Le meilleur résultat obtenu suite à la calibration pourra être utilisé dans une application d'ébavurage robotisé avec un capteur de force/moment. Par contre, étant donné l'équipement industriel utilisé, la boucle de contrôle ne peut être modifiée. Les informations retournées par le capteur de force/moment ne peuvent être utilisées qu'à travers les fonctions rendues disponibles par le contrôleur.

Les théories sur le contrôle en force et position (hybride, parallèle, impédance, etc.) développées dans plusieurs références (Chiaverini, Siciliano et Villani, 1994; Khatib et Burdick, 1986; Siciliano et Villani, 1999) ne pourront donc s'appliquer tel que mentionné précédemment.

Par contre, bien qu'elles ne puissent être mise en œuvre avec l'équipement dont nous disposons, certaines techniques ont permis de mieux comprendre les particularités de l'ébavurage robotisé avec contrôle de force. Par exemple, Liao *et al.* (2008) expliquent le fonctionnement d'une application d'ébavurage qui utilise un outil particulier, capable de s'ajuster à la surface et de retourner l'information sur l'ajustement qu'il fait. Le robot peut donc compenser sa trajectoire en utilisant cette information. D'autres ont utilisé un capteur de force/moment pour contrôler les déplacements du robot en fonction de l'information qu'il retourne (Ramming *et al.*, 2002; Xianlun, Yong et Yunna, 2007). De façon générale, tous mettent l'accent sur l'importance de la qualité de l'information tirée du capteur (sensibilité et bruit), la rapidité de traitement de cette information et la capacité du robot à réagir promptement aux commandes envoyées par le contrôleur. Ces aspects du contrôle avec retour de force seront étudiés lors de la mise en œuvre de la solution.

1.8 Conclusion

Dans cette section, les termes importants, les enjeux et limites de la calibration ont été expliqués et le choix d'avenues à approfondir a été explicité. De plus, bien que l'application qui sera développée ne puisse profiter directement de ce qui a été publié en terme de contrôle de force, plusieurs concepts clefs tirés des références guideront sa mise en œuvre.

CHAPITRE 2

MODÉLISATION ROBOTIQUE

Ce chapitre développe la cinématique directe et inverse utilisée et présente brièvement les outils mathématiques qui ont été mis à profit pour simuler les différentes techniques. Les aboutissements de ce chapitre seront mis à contribution tout au long de l'ouvrage.

Le robot utilisé lors des simulations et des manipulations est le IRB 140 d'ABB, tel qu'illustré à la Figure 2.1. C'est un robot à six articulations rotatives permettant d'avoir six DDL sur la position de son extrémité.

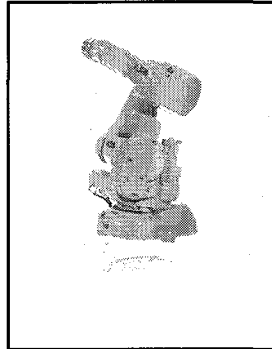


Figure 2.1 Robot ABB IRB 140.
Tirée du site web d'ABB (2010)

Dans le cadre de l'application d'ébavurage robotisé avec contrôle de force, différents outils y seront fixés, calibrés et utilisés. Par contre, seul un palpeur sera considéré lors de la calibration.

2.1 Cinématique directe

La cinématique directe (*voir* Équation (2.1)) est la fonction permettant d'exprimer la pose du robot selon une configuration donnée des articulations. La pose du robot est l'expression de la position et de l'orientation de l'extrémité du robot dans le domaine cartésien et est

représentée par $\mathbf{x} = [x, y, z, C, B, A]^T$, où C, B et A sont les angles de rotation autour des axes Z, Y et X de la base du robot, tel que définit par la convention d'Euler (Craig, 2005). La position des articulations est représentée par le vecteur $\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T$. Dans ce cas on parle aussi de position dans le domaine articulaire. Étant donné la nature rotative de chacune des articulations, un angle θ est assigné à chacun d'eux pour définir la configuration du robot.

$$\mathbf{x} = f(\mathbf{q}) \quad (2.1)$$

D'abord, la modélisation nominale est faite selon les dimensions du robot fournies par le fabricant et la représentation DHm, telle que présentée par Craig (2005). Le Tableau 2.1 présente les paramètres DHm selon le modèle de la Figure 2.2. Les angles de la dernière colonne du tableau correspondent à la position pour laquelle l'analyse est faite.

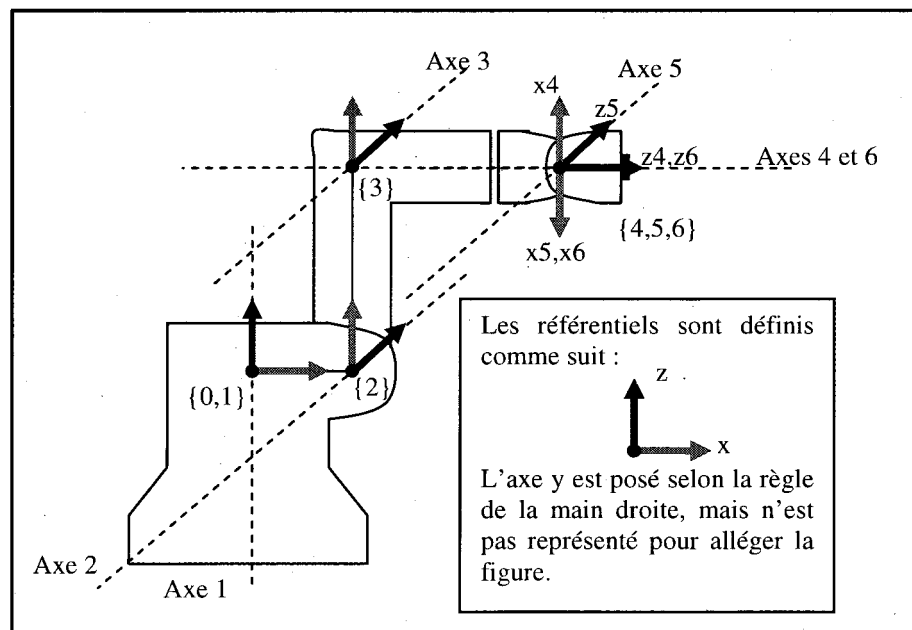


Figure 2.2 Modèle DHm du IRB 140 d'ABB.

Tableau 2.1 DHm du IRB 140 d'ABB

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1	0	0	0	0
2	-pi/2	70	0	-pi/2
3	0	360	0	0
4	-pi/2	0	380	0
5	pi/2	0	0	pi
6	pi/2	0	0	0

À partir de ce dernier tableau, les équations de la cinématique directe sont définies, toujours selon la méthode présentée par Craig (2005). Les transformées homogènes, matrices exprimant les translations et rotations entre différents référentiels, suivront donc les standards définis dans cet ouvrage de référence (*voir* Équations (2.2) et (2.3)).

$${}^{i-1}\mathbf{T}_{(\alpha_j, a_j, d_j, \theta_j)} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1})\sin(\theta_i) & \cos(\alpha_{i-1})\cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i \sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1})\sin(\theta_i) & \sin(\alpha_{i-1})\cos(\theta_i) & \cos(\alpha_{i-1}) & d_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$${}^0\mathbf{T}_n = {}^0\mathbf{T}_1 \mathbf{T}_2^1 \mathbf{T}_3^2 \mathbf{T} \dots {}^{n-2}\mathbf{T}_{n-1}^{n-2} \mathbf{T}_n^{n-1} \quad (2.3)$$

Dans le cas présent, elles sont exprimées par les équations (2.4) à (2.9). Ce choix de modélisation permet de simplifier l'approche, entre autres, en permettant l'utilisation de la librairie robotique rendue disponible dans MATLAB®.

$${}^0\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$${}^1_2\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 70 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$${}^2_3\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 360 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$${}^3_4\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 380 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$${}^4_5\mathbf{T} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$${}^5_6\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Le modèle nominal présenté ici ne tient pas compte des tolérances de fabrication et des erreurs dimensionnelles provenant de l'assemblage. Le modèle réel donnerait un tableau DHm semblable au Tableau 2.1, mais toutes les dimensions du robot seraient légèrement différentes des dimensions nominales. C'est ce vecteur d'erreur de modélisation, $\Delta\mathbf{p} = [\Delta\mathbf{a}^T, \Delta\mathbf{a}^T, \Delta\mathbf{d}^T, \Delta\theta^T]^T$, qui sera recherchée par la mise en œuvre des techniques de calibration.

L'algorithme de calibration devra donc générer un tableau DHm corrigé, se rapprochant le plus près possible de celui que donnerait le modèle réel. Dans un contexte de mise en œuvre,

le modèle réel est inconnu. Pour vérifier la qualité du modèle corrigé, nous devons comparer les touches faites sur une surface connue avant et après correction du modèle.

En ce qui concerne la calibration du TCP de l'outil, ce dernier peut être considéré comme un membre ajouté au robot. La transformée homogène exprimant son extrémité {T} par rapport à l'extrémité du robot {E} ${}^E_T\mathbf{T}$ multipliera la transformée homogène de l'extrémité du robot {E} par rapport à la base {0} ${}^0_E\mathbf{T}$:

$${}^0_T\mathbf{T} = {}^0_1\mathbf{T} {}^1_2\mathbf{T} {}^2_3\mathbf{T} {}^3_4\mathbf{T} {}^4_5\mathbf{T} {}^5_6\mathbf{T} {}^6_E\mathbf{T} {}^E_T\mathbf{T}, \quad (2.10)$$

où ${}^6_E\mathbf{T}$ est une matrice présentant uniquement une translation :

$${}^6_E\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 65 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

Dans ce cas, ce sont les paramètres de la transformée homogène de l'outil ${}^E_T\mathbf{T}$ qui seront les seuls paramètres inconnus et le modèle du robot sera considéré comme idéal.

2.2 Cinématique inverse

La cinématique inverse du robot est l'inverse de la cinématique directe (Voir Équation (2.12)).

$$\mathbf{q} = f^{-1}(\mathbf{x}) \quad (2.12)$$

Elle permet donc d'obtenir la configuration articulaire permettant d'atteindre une pose donnée. La valeur de chacune des articulations est exprimée dans un vecteur \mathbf{q} de dimension n , égale au nombre d'articulations du robot. Pour un robot relativement simple, cette fonction

peut se retrouver géométriquement ou algébriquement. Par contre, lorsque le tableau DHm contient plusieurs cases différentes de zéro, une approche itérative sera plutôt envisagée. Ce sera, entre autres le cas lorsque le modèle réel du robot sera à considérer.

2.2.1 Cinématique inverse exacte

Selon la configuration cinématique nominale du IRB 140, il est possible de déterminer la position de l'extrémité du robot par les trois premières articulations (Voir Équations (2.13) à (2.17)) et l'orientation par les trois derniers. Cette approche est définie comme étant celle de Pieper (Craig, 2005).

$$q_1 = \text{atan2}(y, x) \quad (2.13)$$

$$q_2 = - \left(\text{atan} \left(dz, \sqrt{x^2 + y^2} \right) + \text{asin} \left(\frac{\sin(\beta) * l_2}{\sqrt{dz^2 + x^2 + y^2}} \right) \right) \quad (2.14)$$

$$q_3 = \pi - \beta \quad (2.15)$$

$$\beta = \frac{l_1^2 + l_2^2 - \text{sqrt}(x^2 + y^2)}{2 * l_1 * l_2} \quad (2.16)$$

$$dz = z - h_0 \quad (2.17)$$

Où x , y et z sont les coordonnées de la position désirée, l_1 et l_2 les longueurs des membres du robot et h_0 la hauteur du référentiel 0 du robot par rapport au plan où est fixé la base du robot.

Pour les orientations, il faut tout d'abord retrouver la valeur de q_5 selon les équations (2.18) et (2.19).

$$q_5 = \text{atan2} \left(\sqrt{1 - {}^3R(2,3)^2}, {}^3R(2,3) \right) \quad (2.18)$$

$${}^3R = \left({}^0R \Big|_{q_1, q_2, q_3} \right)^{-1} {}^0R \quad (2.19)$$

Où ${}^0_3\mathbf{R}$ est la matrice de rotation exprimant l'orientation de l'articulation trois par rapport à la base et ${}^0_6\mathbf{R}$ est la matrice de rotation désirée à la sixième articulation.

Puis si q_5 vaut 0, les équations (2.20) et (2.21) s'appliqueront.

$$q_4 = 0 \quad (2.20)$$

$$q_6 = \text{atan2}\left(-{}^3_6R(3,1), {}^3_6R(1,1)\right) \quad (2.21)$$

Si q_5 vaut π , les équations (2.22) et (2.23) s'appliqueront.

$$q_4 = \pi \quad (2.22)$$

$$q_6 = \text{atan2}\left(-{}^3_6R(3,1), -{}^3_6R(1,1)\right) \quad (2.23)$$

Finalement, dans tous les autres cas, ce sont les équations (2.24) et (2.25) qui s'appliquent.

$$q_4 = \text{atan2}\left(\frac{{}^3_6R(3,3)}{\sin(q_5)}, \frac{-{}^3_6R(1,3)}{\sin(q_5)}\right) \quad (2.24)$$

$$q_6 = \text{atan2}\left(\frac{-{}^3_6R(2,2)}{\sin(q_5)}, \frac{{}^3_6R(2,1)}{\sin(q_5)}\right) \quad (2.25)$$

2.2.2 Cinématique inverse itérative

Lors de l'optimisation, c'est le modèle réel du robot qui sera considéré. La cinématique inverse exacte ne sera donc plus suffisante puisqu'elle tient compte de plusieurs simplifications qui ne seront plus vraies lorsque tous les paramètres du tableau DHm ont une valeur différente de zéro.

La librairie robotique de MATLAB® (Corke, 1996) offre une fonction permettant de réaliser la cinématique inverse de façon itérative. Son fonctionnement nécessite de connaître le modèle cinématique du robot (modèle DHm) et la pose à atteindre. Il est aussi intéressant de fournir une position de départ dans le domaine articulaire pour éviter que le robot ne converge autour d'une position rendue impossible, par exemple à cause de contraintes de l'environnement. La cinématique inverse exacte pourra être utilisée à cette fin.

La fonction retournera donc la position dans le domaine articulaire correspondant à la pose commandée, à l'intérieur d'une tolérance qui peut être ajustée au besoin.

2.3 Librairie robotique MATLAB®

Autre que par sa fonction de cinématique inverse itérative, la librairie robotique MATLAB® développée par P. Corke est un outil qui permettra d'accélérer considérablement les travaux. La modélisation des robots est grandement facilitée, plusieurs fonctions spécifiques à la robotique sont déjà disponibles et des fonctions utilitaires y ont été intégrées.

Lors des simulations d'algorithmes de calibration, quelques développements mathématiques différents ont été essayés. Par contre, tous ces développements ont nécessité l'utilisation de la librairie robotique pour effectuer les tâches de modélisation du robot, de génération de points de tests selon certaines contraintes, de validation des poses à chacun des points et d'affichage de résultats graphiques.

2.4 Conclusion

Le modèle développé dans ce chapitre ainsi que les formules et les outils mathématiques expliqués seront mis à contribution tout au long du reste de l'ouvrage, notamment pour le développement des algorithmes de calibration et d'optimisation au prochain chapitre.

CHAPITRE 3

ALGORITHMES DE CALIBRATION ET OPTIMISATION

Dans ce chapitre, la théorie et les équations nécessaires à la simulation et la mise en œuvre des solutions de calibrations explorées seront présentées. Les estimations par linéarisation de premier et second ordre seront d'abord développées, étant donné qu'elles seront utilisées pour deux des techniques présentées. Puis, les constructions mathématiques et les fonctions objectives particulières à chacune des techniques de calibration seront présentées. Finalement, les différentes approches permettant la mise en œuvre des résultats seront aussi explicitées.

3.1 La linéarisation

La linéarisation est l'approximation linéaire des sorties d'un système par rapport à la variation des entrées autour d'une valeur donnée, aussi appelé point d'équilibre. Cette approximation est faite par l'expansion des séries de Taylor. Il est très courant de n'utiliser que le premier ordre de la série, ce qui permet d'obtenir une équation linéaire reliant les sorties aux entrées d'un système non linéaire. Par contre, comme notre application demande un très grand degré de précision, nous explorerons l'approximation de second ordre, malgré la complexité ajoutée relative à l'introduction du terme quadratique.

3.1.1 Linéarisation premier ordre

Dans les cas qui nous intéressent, les variables recherchées sont les erreurs de modélisation cinématique du robot. Pour les retrouver, une matrice jacobienne spéciale (Mirman et Gupta, 1992), liant linéairement l'erreur de position du robot aux erreurs de modélisation cinématiques sera utilisée. Cette matrice est une approximation linéaire de premier ordre calculée selon l'équation (3.1) :

$$\mathbf{J}_i = \left[\frac{\partial f_{p,nominal}}{\partial \boldsymbol{\rho}} \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}, \quad (3.1)$$

où $f_{p,nominal} = [x^0, y^0, z^0]^T$ est la partie position de la cinématique directe évaluée selon le modèle nominal du robot. Le vecteur $\boldsymbol{\rho} = [\boldsymbol{\alpha}^T, \mathbf{a}^T, \mathbf{d}^T, \boldsymbol{\theta}^T]^T$ est un vecteur colonne contenant tous les paramètres du modèle DHm présenté au Tableau 2.1. Chaque dérivée partielle est évaluée pour une configuration articulaire donnée, ce qui donne une matrice \mathbf{J} de dimension 3 par m , où il y a m paramètres erronés.

3.1.2 Linéarisation second ordre

La linéarisation de second ordre permet de préciser l'approximation linéaire en ajoutant une matrice liant l'erreur de la position cartésienne de l'effecteur au carré de l'erreur de paramétrisation. Dans le contexte actuel, on appellera cette matrice la matrice hessienne spéciale et elle sera calculée de la façon suivant :

$$\mathbf{H}_i = \left[\begin{array}{ccc} \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_1^2} & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_1 \partial \boldsymbol{\rho}_2} & \dots & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_1 \partial \boldsymbol{\rho}_m} \\ \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_2 \partial \boldsymbol{\rho}_1} & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_2^2} & \dots & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_2 \partial \boldsymbol{\rho}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_m \partial \boldsymbol{\rho}_1} & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_m \partial \boldsymbol{\rho}_2} & \dots & \frac{\partial^2 f_{p,nominal}}{\partial \boldsymbol{\rho}_m^2} \end{array} \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \quad (3.2)$$

Les matrices $\mathbf{H}_{i,x}$, $\mathbf{H}_{i,y}$ et $\mathbf{H}_{i,z}$ seront de dimension m par m .

3.2 Constructions mathématiques à partir d'une contrainte planaire

Les développements mathématiques suivants sont fortement inspirés des articles (Zhong et Lewis, 1995; Zhong, Lewis et L.N.-Nagy, 1996) et (Ikits et Hollerbach, 1997). La technique exploitée consiste à imposer une contrainte planaire aux touches faites par le robot. Deux

approches mathématiques seront explorées pour traduire cette contrainte. Une fonction objective dépendante de l'erreur de modélisation peut être construite, l'erreur de modélisation sera retrouvée par optimisation. Plusieurs techniques d'optimisation sont rendues disponibles par l'utilisation de MATLAB®. Par contre, certaines d'entre elles montrent des résultats plus concluants.

3.2.1 Produits vectoriels

La première approche exploite la multiplication de vecteurs parallèles qui doit donner un vecteur nul (Zhong, Lewis et L.N.-Nagy, 1996).

Tel que mentionné au chapitre 2, étant donné l'imprécision du modèle nominal du robot par rapport au modèle réel, la cinématique directe ne donnera jamais la position réelle du robot. Par contre, on peut améliorer l'approximation de la position réelle du robot en utilisant le vecteur d'erreur de modélisation $\Delta \mathbf{p} = [\Delta \mathbf{a}^T, \Delta \mathbf{a}^T, \Delta \mathbf{d}^T, \Delta \boldsymbol{\theta}^T]^T$ (Voir Équations (3.3) et (3.4), selon l'utilisation ou non de la matrice hessienne spéciale).

$$\mathbf{x}_{p,i} \approx f_{p,nominal}(\boldsymbol{\theta}_i) + \mathbf{J}_i \Delta \mathbf{p} = [x_i^0, y_i^0, z_i^0]^T + \mathbf{J}_i \Delta \mathbf{p} \quad (3.3)$$

$$\mathbf{x}_{p,i} \approx f_{p,nominal}(\boldsymbol{\theta}_i) + \mathbf{J}_i \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^T \mathbf{H}_i \Delta \mathbf{p} = [x_i^0, y_i^0, z_i^0]^T + \mathbf{J}_i \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^T \mathbf{H}_i \Delta \mathbf{p} \quad (3.4)$$

Pour construire la contrainte mathématique, des vecteurs sont créés en reliant deux touches consécutives (Voir Équations (3.5)). Ces vecteurs doivent donc eux aussi répondre à la contrainte planaire. De plus, pour chaque paire de vecteurs consécutifs, un vecteur normal au plan sera calculé (Voir Équation (3.6)). Selon la contrainte planaire, ils devront être parallèles. Par ailleurs, les vecteurs normaux au plan doivent être normalisés. Leur longueur dépend de la longueur des vecteurs à partir desquels ils ont été calculés et de l'angle entre ces vecteurs, deux facteurs qui ne doivent pas influencer le reste des développements.

$$\mathbf{v}_i = \mathbf{x}_{p,i+1} - \mathbf{x}_{p,i} \quad (3.5)$$

$$\mathbf{n}_i = \frac{\mathbf{v}_{i+1} \times \mathbf{v}_i}{\|\mathbf{v}_{i+1} \times \mathbf{v}_i\|} \quad (3.6)$$

Finalement, tel que mentionné précédemment, le produit de deux vecteurs normaux consécutifs doit donner un vecteur nul.

Il est à noter que dans les articles de Zhong (Zhong et Lewis, 1995; Zhong, Lewis et L.N.-Nagy, 1996), les positions robot sont estimées selon l'équation (3.3), les vecteurs sont normalisés (Voir Équation (3.7)), l'expression à minimiser est exprimée sous forme de déterminant (Voir Équation (3.8)) et les produit vectoriels des normes sont simplifiés tel que présentés dans l'équation (3.9) pour en arriver à la construction d'une matrice \mathbf{M} (Voir Équation (3.10)). Ce qui permet d'écrire une équation de contrainte linéaire pour en arriver à une fonction objective relativement simple (Voir Équation (3.11)).

$$\mathbf{v}_i \approx \frac{[\Delta x_i^0 \quad \Delta y_i^0 \quad \Delta z_i^0]^T + [\Delta \mathbf{J}_i^x \quad \Delta \mathbf{J}_i^y \quad \Delta \mathbf{J}_i^z]^T * \Delta \boldsymbol{\rho}}{\sqrt{(\Delta x_i^0)^2 + (\Delta y_i^0)^2 + (\Delta z_i^0)^2}} \quad (3.7)$$

$$\begin{bmatrix} \mathbf{v}_i & \mathbf{v}_{i+1} & \mathbf{v}_{i+2} \end{bmatrix} = \begin{bmatrix} \Delta x_i^0 + \Delta \mathbf{J}_i^x \Delta \boldsymbol{\rho} & \Delta x_{i+1}^0 + \Delta \mathbf{J}_{i+1}^x \Delta \boldsymbol{\rho} & \Delta x_{i+2}^0 + \Delta \mathbf{J}_{i+2}^x \Delta \boldsymbol{\rho} \\ \Delta y_i^0 + \Delta \mathbf{J}_i^y \Delta \boldsymbol{\rho} & \Delta y_{i+1}^0 + \Delta \mathbf{J}_{i+1}^y \Delta \boldsymbol{\rho} & \Delta y_{i+2}^0 + \Delta \mathbf{J}_{i+2}^y \Delta \boldsymbol{\rho} \\ \Delta z_i^0 + \Delta \mathbf{J}_i^z \Delta \boldsymbol{\rho} & \Delta z_{i+1}^0 + \Delta \mathbf{J}_{i+1}^z \Delta \boldsymbol{\rho} & \Delta z_{i+2}^0 + \Delta \mathbf{J}_{i+2}^z \Delta \boldsymbol{\rho} \end{bmatrix} = 0 \quad (3.8)$$

$$\begin{bmatrix} \mathbf{v}_i & \mathbf{v}_{i+1} & \mathbf{v}_{i+2} \end{bmatrix} \approx \begin{bmatrix} \Delta \mathbf{J}_i^x \Delta \boldsymbol{\rho} & \Delta x_{i+1}^0 & \Delta x_{i+2}^0 \\ \Delta \mathbf{J}_i^y \Delta \boldsymbol{\rho} & \Delta y_{i+1}^0 & \Delta y_{i+2}^0 \\ \Delta \mathbf{J}_i^z \Delta \boldsymbol{\rho} & \Delta z_{i+1}^0 & \Delta z_{i+2}^0 \end{bmatrix} + \begin{bmatrix} \Delta x_i^0 & \Delta \mathbf{J}_{i+1}^x \Delta \boldsymbol{\rho} & \Delta x_{i+2}^0 \\ \Delta y_i^0 & \Delta \mathbf{J}_{i+1}^y \Delta \boldsymbol{\rho} & \Delta y_{i+2}^0 \\ \Delta z_i^0 & \Delta \mathbf{J}_{i+1}^z \Delta \boldsymbol{\rho} & \Delta z_{i+2}^0 \end{bmatrix} + \quad (3.9)$$

$$\begin{bmatrix} \Delta x_i^0 & \Delta x_{i+1}^0 & \Delta \mathbf{J}_{i+2}^x \Delta \boldsymbol{\rho} \\ \Delta y_i^0 & \Delta y_{i+1}^0 & \Delta \mathbf{J}_{i+2}^y \Delta \boldsymbol{\rho} \\ \Delta z_i^0 & \Delta z_{i+1}^0 & \Delta \mathbf{J}_{i+2}^z \Delta \boldsymbol{\rho} \end{bmatrix} + \underbrace{\begin{bmatrix} \Delta x_i^0 & \Delta x_{i+1}^0 & \Delta x_{i+2}^0 \\ \Delta y_i^0 & \Delta y_{i+1}^0 & \Delta y_{i+2}^0 \\ \Delta z_i^0 & \Delta z_{i+1}^0 & \Delta z_{i+2}^0 \end{bmatrix}}_{\Delta \mathbf{X}_i} = 0$$

$$\mathbf{M}_i = \begin{vmatrix} \Delta \mathbf{J}_i^x & \Delta x_{i+1}^0 & \Delta x_{i+2}^0 \\ \Delta \mathbf{J}_i^y & \Delta y_{i+1}^0 & \Delta y_{i+2}^0 \\ \Delta \mathbf{J}_i^z & \Delta z_{i+1}^0 & \Delta z_{i+2}^0 \end{vmatrix} + \begin{vmatrix} \Delta x_i^0 & \Delta \mathbf{J}_{i+1}^x & \Delta x_{i+2}^0 \\ \Delta y_i^0 & \Delta \mathbf{J}_{i+1}^y & \Delta y_{i+2}^0 \\ \Delta z_i^0 & \Delta \mathbf{J}_{i+1}^z & \Delta z_{i+2}^0 \end{vmatrix} + \begin{vmatrix} \Delta x_i^0 & \Delta x_{i+1}^0 & \Delta \mathbf{J}_{i+2}^x \\ \Delta y_i^0 & \Delta y_{i+1}^0 & \Delta \mathbf{J}_{i+2}^y \\ \Delta z_i^0 & \Delta z_{i+1}^0 & \Delta \mathbf{J}_{i+2}^z \end{vmatrix} \quad (3.10)$$

$$Q = \sum_{i=1}^{t-3} (\mathbf{M}_i \Delta \boldsymbol{\rho} + \Delta \mathbf{X}_i)^2 + R(\Delta \boldsymbol{\rho}^T \Delta \boldsymbol{\rho}) \quad (3.11)$$

où t est le nombre de touches faites sur la surface planaire par le robot et R est un paramètre de pondération pour le terme de régulation, dans le but d'améliorer les performances de l'algorithme d'optimisation. Dans le cadre du présent travail, les résultats obtenus suivant cette simplification seront comparés aux résultats obtenus selon les fonctions objectives présentées plus loin.

D'autre part, en omettant les simplifications suggérées par Zhong, la fonction objective suivante est construite :

$$Q = \sum_{i=1}^{t-3} \|\mathbf{n}_i \times \mathbf{n}_{i+1}\| + R(\Delta \boldsymbol{\rho}^T \Delta \boldsymbol{\rho}), \quad (3.12)$$

où le produit vectoriel de vecteurs normaux \mathbf{n}_i résulte en une construction mathématique complexe utilisant ou non la matrice hessienne spéciale. Il est clair que l'optimisation de cette fonction présente un plus grand défi que celle de l'équation (3.11).

3.2.2 Plan moyen

En utilisant le même type de contraintes planaires, il est possible de poser une fonction objective selon le plan moyen défini par les touches du robot, telles qu'estimées par les équations (3.3) ou (3.4). Cette fonction objective tentera de minimiser la distance entre chaque point et le plan moyen, en optimisant la valeur des erreurs de modélisation.

L'expression du plan selon l'équation (3.13) peut aussi être faite sous forme matricielle (Voir Équation (3.14) en posant $C = -1$. Les paramètres A , B et D sont retrouvés en utilisant l'équation (3.15).

$$Ax + By + Cz + D = 0 \quad (3.13)$$

$$\underbrace{\begin{bmatrix} x_i & y_i & 1 \\ \vdots & \vdots & \vdots \\ x_t & y_t & 1 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} A \\ B \\ D \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} z_i \\ \vdots \\ z_t \end{bmatrix}}_{\mathbf{Y}} \quad (3.14)$$

$$\mathbf{X} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T \mathbf{Y} \quad (3.15)$$

x_i , y_i et z_i sont les coordonnées des touches faites par le robot, exprimées selon l'équation (3.3) ou (3.4). Puis la distance entre chacun des points et le plan est calculée selon l'équation (3.16). La fonction objective peut donc être bâtie pour minimiser la somme des distances (Voir l'Équation (3.17)). Un terme de régulation, pondéré par le paramètre R , est encore une fois ajouté pour améliorer les performances de l'optimisation.

$$d_i = \frac{Ax_i + By_i + Cz_i + D}{\sqrt{A^2 + B^2 + C^2}} \quad (3.16)$$

$$Q = \sum_{i=1}^t (\mathbf{d}_i)^2 + R(\Delta \boldsymbol{\rho}^T \Delta \boldsymbol{\rho}) \quad (3.17)$$

De plus, pour maximiser les comparatifs, une fonction objective a été développée en supposant que la position du plan est connue. Dans ce cas, les paramètres A , B , C et D de l'équation (3.16) sont connus et n'ont pas à être évalués. Par contre, la même fonction objective sera utilisée.

Il en va de même pour une situation où trois plans non parallèles entre eux seraient utilisés pour maximiser l'observabilité des paramètres. L'équation (3.15) est utilisée pour retrouver les paramètres A , B , C et D de chacun des trois plans en fonction des touches qui ont été faites sur chacun d'eux. Puis, l'équation (3.17) fait office de fonction objective. Dans le cas où la position des trois plans est connue, l'équation (3.15) n'a pas à être utilisée et on calcule directement les distances entre les touches faites et le plan où elles ont été faites en utilisant les équations (3.16) et (3.17).

Les problèmes d'optimisation posés par chacune des trois fonctions objectives montrent une grande non-linéarité. Il faudra porter attention à l'observabilité de chacune des variables à retrouver, c'est-à-dire au conditionnement du problème.

Finalement, plusieurs techniques d'optimisation sont rendues accessibles par l'utilisation de MATLAB®, les techniques les plus intéressantes sont *fminsearch*, *fminunc* et *lsqnonlin* qui permettent d'utiliser les algorithmes de *Levenberg-Marquardt*.

3.3 Calcul des dimensions de l'outil

En revanche, la calibration d'un outil ou du TCP d'un outil implique beaucoup moins d'approximation et de non-linéarité. Comme l'approche devra être testée en industrie, l'équipement dont nous disposons est tout indiqué pour tenter de mettre en œuvre une technique utilisant un palpeur et une sphère de dimension connue. Ce genre de dispositif est très commun dans les industries de précision comme l'aéronautique.

Pour ce genre de calibration, on suppose que la modélisation du robot est parfaite, c'est-à-dire que la position de l'extrémité du robot $\{E\}$, telle que retournée par le robot correspond à la position réelle dans l'espace cartésien. Bien que ce ne soit pas l'objectif premier de cette opération, nous sommes tout de même en droit de penser que la calibration d'un outil pour un certain volume d'opération pourra permettre de compenser certaines erreurs de modélisation du robot lui-même. Ainsi donc, les dimensions recherchées ne sont pas nécessairement les dimensions réelles de l'outil, mais bien celles qui minimisent la fonction objective et la contrainte de positionnement et maximise la précision pour ces configurations de robot dans le volume donné.

3.3.1 Sphère moyenne

Plusieurs touches seront donc faites sur une sphère avec un palpeur. Comme les dimensions du palpeur sont inconnues, mais que nous considérons la position du robot dans l'espace cartésien comme étant parfaite, c'est la position de l'extrémité du robot {E} qui sera prise en note.

Les dimensions du palpeur ${}^E x_T, {}^E y_T, {}^E z_T$ sont utilisées pour calculer les t positions des touches ($P_i = [x_i, y_i, z_i]^T$) sur la sphère (Voir Équation (3.18)). À partir de ces touches, l'équation de la sphère moyenne est calculée. L'équation (3.19) définit une sphère selon son centre $[x_c, y_c, z_c]^T$ et son rayon r .

$$P_i = P_{W,i} + {}^0R_i \begin{bmatrix} {}^E x_T \\ {}^E y_T \\ {}^E z_T \end{bmatrix} \quad (3.18)$$

$$x_i^2 + y_i^2 + z_i^2 - 2x_i x_c - 2y_i y_c - 2z_i z_c + x_c^2 + y_c^2 + z_c^2 = r^2 \quad (3.19)$$

En posant les équivalences suivantes :

$$a = -2x_c, \quad (3.20)$$

$$b = -2y_c, \quad (3.21)$$

$$c = -2z_c, \quad (3.22)$$

$$d = x_c^2 + y_c^2 + z_c^2 - r^2, \quad (3.23)$$

et en les substituant dans l'équation (3.19), on pose l'équation matricielle (3.24). Comme pour l'équation (3.14) on utilise l'équation (3.15), puis les équations (3.20) à (3.23) pour retrouver r .

$$\underbrace{\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_i & x_i & x_i & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_t & y_t & z_t & 1 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{\mathbf{X}} = - \underbrace{\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 \\ \vdots & \vdots & \vdots \\ x_i^2 & y_i^2 & z_i^2 \\ \vdots & \vdots & \vdots \\ x_t^2 & y_t^2 & z_t^2 \end{bmatrix}}_{\mathbf{Y}} \quad (3.24)$$

Comme la position de la sphère dans l'espace n'est pas connue, mais que son rayon l'est, l'algorithme d'optimisation devra trouver les dimensions de l'outil permettant de minimiser la fonction objective suivante :

$$Q = (r^0 - r)^2, \quad (3.25)$$

où r^0 est le rayon réel de la sphère et r , le rayon calculé suite à l'optimisation.

3.4 Utilisation des résultats dans un contexte industriel

Dans les deux sections précédentes, il a été question de techniques pour retrouver des paramètres permettant d'améliorer la connaissance cinématique de la structure robotisée utilisée. Une fois ces dimensions retrouvées, il faut établir une stratégie pour pouvoir les utiliser dans un contexte réel. La réflexion est d'autant plus nécessaire que dans un contexte industriel, les contrôleurs robotisés n'ont pas la même flexibilité que ceux qui sont bâtis sur mesure, comme on en retrouve généralement dans les laboratoires de recherche ou à vocations académiques.

3.4.1 Modèle cinématique

Le modèle cinématique d'un robot industriel est largement utilisé puisqu'il permet de calculer les cinématiques directes et inverses du robot. Par contre, les paramètres sont généralement inaccessibles dans le contrôleur et les compagnies sont très réticentes à donner accès à ceux-ci aux utilisateurs.

Pour contourner cette impasse, un algorithme (Voir Figure 3.1) fortement inspirée d'un article de Duelen et Schroer (1991) met en œuvre la cinématique inverse calculée selon les paramètres optimisés du robot ($f_{\text{optimisé}}^{-1}$) et la cinématique directe selon les paramètres nominaux (f_{nominal}). Il permet de calculer les positions dans l'espace cartésien $\tilde{\mathbf{x}}_d$ à envoyer au robot pour qu'il se rende à la position réelle désirée \mathbf{x}_d . De cette façon les positions envoyées au robot dans l'espace cartésien tiendront compte de l'erreur de modélisation et le robot se positionnera à la position désirée.

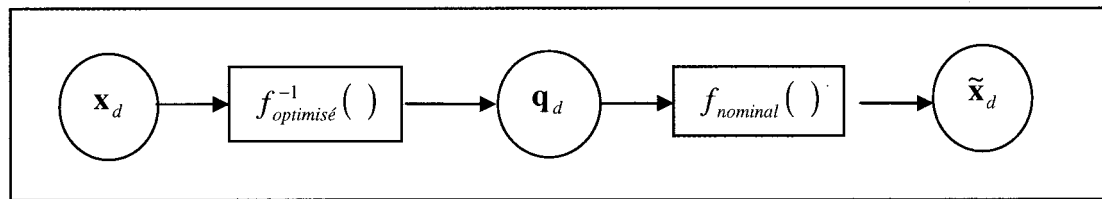


Figure 3.1 Algorithme de calcul des poses à envoyer au robot selon les poses désirées et les erreurs de modélisation.

Cet algorithme pourrait être mis en œuvre dans le *post-processor* de l'outil de programmation hors-ligne ou encore comme une fonction dans le contrôleur du robot. Dans ce dernier cas, toutes les instructions ordonnant un déplacement à un point en coordonnées cartésiennes dans l'espace devront être redéfinies pour inclure l'algorithme.

D'autre part, si la commande doit être envoyée dans l'espace articulaire du robot, seules les erreurs angulaires devront être tenues en compte. Il est à noter qu'il sera préférable d'utiliser l'approche avec les coordonnées cartésiennes si l'option de la programmation hors-ligne est retenue puisque les trajectoires sont généralement générées à partir de points dans l'espace cartésien et que le modèle du robot hors-ligne correspond normalement au modèle du robot utilisé dans le contrôleur, soit le modèle nominal.

3.4.2 Sphère moyenne

Cette méthode permet de retrouver de nouvelles dimensions pour le TCP de l'outil calibré. Généralement, les contrôleurs permettent de définir différents TCP d'outils selon les longueurs et orientations du TCP, masses et inerties de l'outil, etc. Comme les déplacements du robot tiendront compte de ces dimensions et qu'elles sont facilement modifiables, il ne suffit que d'utiliser les dimensions optimisées du TCP de l'outil. Dans le cas qui nous intéresse, c'est le palpeur qui sera calibré et seules les grandeurs en $\begin{bmatrix} {}^E x_T, {}^E y_T, {}^E z_T \end{bmatrix}^T$ seront modifiées dans la définition du TCP de l'outil.

3.5 Conclusion

Les développements mathématiques nécessaires à la simulation et la mise en œuvre des différentes techniques de calibration sont présentés dans ce chapitre. Chacune de ces techniques sera d'abord simulée, avant d'en envisager la mise en œuvre. Les résultats des simulations seront présentés dans le prochain chapitre. La technique offrant les meilleures performances pourra être utilisée dans l'application d'ébavurage robotisé.

CHAPITRE 4

SIMULATIONS

Ce chapitre présente l'ensemble des résultats obtenus selon les différentes techniques de calibrations testées. Les résultats de simulation seront présentés et ceux de la mise en œuvre de la technique de calibration du TCP d'outil seront comparés à ceux obtenus avec d'autres techniques. Les techniques devront être évaluées selon leur capacité à améliorer la précision du robot utilisé dans l'application d'ébavurage robotisé. La technique offrant les meilleures performances sera utilisée pour la mise en œuvre de l'application.

4.1 Conditions générales de simulation

Tel que mentionné au chapitre 2, les simulations seront faites avec le logiciel MATLAB® et la librairie robotique permettra de mettre en œuvre le modèle développé dans ce même chapitre. Sauf en cas d'avis contraire, le vecteur d'erreur de modélisation à retrouver, $\Delta \mathbf{p}$, sera composé de 20 paramètres, soit tous ceux du Tableau 2.1, excluant les paramètres α_6 , a_1 , d_1 et d_2 pour des raisons qui seront expliquées dans la prochaine section de ce chapitre. Lorsqu'un modèle réel sera simulé, l'erreur au niveau des longueurs sera limitée à ± 1.5 mm et celle des angles à $\pm 0.015^\circ$. Cette amplitude d'erreur est conforme à ce qu'on retrouve dans la littérature.

Lorsqu'il sera question de touches faites sur un plan par le robot, des touches non colinéaires, faites avec des orientations différentes sur une surface d'une planéité parfaite doivent être considérées. Un coin du plan se situe aux coordonnées (400, -100, 290) (mm) par rapport à la base du robot et ses dimensions sont de 300 mm par 300 mm. La position des touches est générée avec une composante aléatoire et l'orientation de ces touches est calculée à partir de valeurs d'angles complètement aléatoires bornées pour éviter d'obtenir des configurations articulaires physiquement impossibles (où le robot serait en collision avec le plan par exemple). L'objectif est d'imiter ce qui pourrait être mis en pratique et de maximiser la richesse des données pour améliorer les conditions d'optimisation. Lorsque les touches sont

faites sur une sphère pour la dernière technique de calibration, on considère les mêmes conditions de touches et une sphère parfaitement sphérique.

L'optimisation sera effectuée par des fonctions disponibles dans MATLAB® et le facteur de pondération R du terme de régulation sera spécifié pour chaque simulation puisqu'il varie selon les performances observées.

La transformée de l'outil qui sera simulé au bout du robot s'exprime par l'équation (4.1). Ces dimensions seront supposées idéales pour simplifier les tests.

$${}^E_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

Finalement, la qualité des solutions obtenues lors de la calibration du robot sera évaluée selon l'erreur globale moyenne (Voir Équation (4.2)). Elle représente la moyenne de la distance entre la position où le robot se trouve réellement et celle où il croit qu'il est. C'est cette valeur qui doit être améliorée après optimisation pour considérer que la précision du robot a été améliorée pour un certain volume d'opération.

$$ErreurGlobaleMoyenne = \frac{\sum_{i=1}^t \|P_i - \tilde{P}_i\|}{t} \quad (4.2)$$

Où P_i indique la position calculée selon la cinématique réelle du robot et \tilde{P}_i indique la position calculée selon la cinématique nominale ou retrouvée par optimisation.

Pour évaluer la meilleure solution lors de la calibration du TCP, 15 touches seront faites sur une plaque de verre de planéité de 50.8 nm. Un plan moyen sera construit en fonction de la position de chacune des touches (Voir Équations (3.13) à (3.15)), évaluées selon les différents TCP trouvés. La somme des distances entre chaque touche et le plan sera calculée

(Voir Équation (3.16)) et le TCP admettant la plus petite valeur sera considéré comme le meilleur.

4.2 Méthode des produits vectoriels simplifiés (Zhong, Lewis et L.N.-Nagy, 1996)

Cette première technique de calibration qui sera mise en application réfère à la fonction objective présentée à l'équation (3.11). C'est donc dire qu'à partir de touches faites sur une surface plane, la position du TCP de l'outil sera exprimée en fonction de l'erreur de modélisation et de la matrice jacobienne spéciale présentée au chapitre précédent (Voir Équation (3.3)). Des vecteurs sont construits à partir de l'expression de ces touches et l'expression du double produit vectoriel de ces vecteurs est simplifié pour en arriver à un système linéaire d'équation qui pourra être solutionné de différente façon.

Pour cette première méthode, les angles des articulations (θ) ont été considérés idéaux pour simplifier la problématique et la totalité des 18 paramètres (Voir Tableau 2.1) restants ont été considérés comme erronés. Malgré cette simplification, les tests, dont quelques résultats sont présentés au Tableau 4.1, indiquent que la matrice \mathbf{M} a un rang maximal de 14 et un conditionnement très élevé. Le conditionnement indique la sensibilité d'un système linéaire aux variations d'entrées. Plus cette valeur est près de 1, plus le système donnera une solution fiable en solutionnant l'équation linéaire. Dans notre cas, c'est aussi un bon indicateur de la solvabilité du système par optimisation.

Les résultats présentés ici ont été générés avec 25 touches. Les valeurs de rang et de conditionnement ont été trouvées en utilisant les fonctions *rank* et *cond* de MATLAB®.

Tableau 4.1 Rang et conditionnement de la matrice \mathbf{M} selon différentes configurations d'erreurs à retrouver

Nombre de paramètres	Paramètres erronés	Rang (\mathbf{M})	Conditionnement (\mathbf{M})
18	α_1 à α_6 a_1 à a_6 d_1 à d_6	14	2.2436×10^{23}
14	α_1 à α_5 a_2 à a_6 d_3 à d_6	14	7.4672×10^8
7	a_2 à a_5 d_3 à d_5	7	7.5166×10^3

Ces résultats ont permis de mettre en évidence le fait que les paramètres α_6 , a_1 , d_1 et d_2 n'ont aucun impact sur la solution du système linéaire dans cette situation. En effet, le paramètre d_2 caractérise deux articulations parallèles sur le robot utilisé. Tel que démontré par Hayati et Mirmirani (1985), il peut donc être fixé à zéro. Ces derniers proposent d'ailleurs l'ajout du paramètre β , ce qui ne sera pas fait dans le cas présent par soucis de simplification. Les paramètres a_1 et d_1 sont quant à eux sans impact sur la matrice \mathbf{M} puisque leur influence sera égale sur chacune des touches, donc annulée lors de la construction des premiers vecteurs selon l'équation (3.5). Finalement, le paramètre α_6 n'a aucune influence étant donné que les dimensions de l'outil mis à l'extrémité du robot sont supposées idéales et que le TCP de l'outil ne comporte qu'une composante en Z par rapport au référentiel $\{E\}$. La matrice \mathbf{M} est donc de plein rang lorsque ces quatre paramètres sont supposés idéaux et exclus du vecteur $\Delta \mathbf{p} = [\Delta \boldsymbol{\alpha}^T, \Delta \mathbf{a}^T, \Delta \mathbf{d}^T, \Delta \boldsymbol{\theta}^T]^T$. Comme indiqué dans la première section du chapitre, les paramètres α_6 , a_1 , d_1 et d_2 seront donc considérés idéaux pour la suite des tests.

D'autre part, on remarque que peu importe le nombre de paramètres qui sont considérés comme idéaux, le conditionnement reste très élevé. L'équation présentera donc une très forte variation des sorties pour une faible variation du vecteur d'entrée.

Dans cette situation, l'optimisation et la résolution de l'équation par matrice inverse ne laissent pas présager de bons résultats. C'est d'ailleurs ce que les tests ont démontrés : les erreurs de modélisation retrouvées sont loin d'être celles induites et n'améliorent pas la précision du robot. Plusieurs tentatives ont été faites, avec des erreurs induites de différentes amplitudes, pour des résultats semblables, quelque soit la méthode de résolution. Les tableaux 4.2 à 4.4 présentent un exemple de résolution de l'équation linéaire où les erreurs de modélisation sont retrouvées selon trois méthodes différentes : résolution par l'inverse de la matrice \mathbf{M} , avec *lsqnonlin* (algorithme *Levenberg-Marquardt*) et avec *fminsearch*. Dans ce cas-ci, 36 touches ont été simulées et le facteur de pondération R a été fixé à 0.01. Pour ces trois techniques l'optimisation est faite sur les 14 paramètres possiblement erronés.

Tableau 4.2 Erreurs de modélisation induites et retrouvées par matrice inverse

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	0 (1×10^{-7})	0 (0)	0 (0)	0 (0)
2 (induite)	0 (0.001)	-17.5078 (0.1)	0 (0)	0 (0)
3 (induite)	0 (1×10^{-7})	-90.0399 (1×10^{-7})	0 (1×10^{-7})	0 (0)
4 (induite)	0 (1×10^{-7})	0 (1×10^{-7})	-95.0421 (-0.1)	0 (0)
5 (induite)	0 (1×10^{-7})	0 (0.1)	0 (1×10^{-7})	0 (0)
6 (induite)	0 (0)	0 (1×10^{-7})	-41.2683 (1×10^{-7})	0 (0)
Erreur globale moyenne nominale (4.2) : 0.0834 mm				
Erreur globale moyenne optimisée (4.2) : 157.6725 mm				

Tableau 4.3 Erreurs de modélisation induites et retrouvées par optimisation, en utilisant la fonction *lsqnonlin* et l'algorithme *Levenberg-Marquardt*

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	-0.0001 (1×10^{-7})	0 (0)	0 (0)	0 (0)
2 (induite)	0.0006 (0.001)	-0.0003 (0.1)	0 (0)	0 (0)
3 (induite)	0.0004 (1×10^{-7})	0.0164 (1×10^{-7})	0 (1×10^{-7})	0 (0)
4 (induite)	0.0002 (1×10^{-7})	0.0029 (1×10^{-7})	-0.0402 (-0.1)	0 (0)
5 (induite)	0 (1×10^{-7})	-0.0075 (0.1)	-0.0005 (1×10^{-7})	0 (0)
6 (induite)	0 (0)	0.0131 (1×10^{-7})	0.0224 (1×10^{-7})	0 (0)
Erreur globale moyenne nominale (4.2) : 0.0834 mm				
Erreur globale moyenne optimisée (4.2) : 0.2273 mm				

Tableau 4.4 Erreurs de modélisation induites et retrouvées par optimisation, en utilisant la fonction *fminsearch*

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	-0.0003 (1×10^{-7})	0 (0)	0 (0)	0 (0)
2 (induite)	0.0008 (0.001)	0.0142 (0.1)	0 (0)	0 (0)
3 (induite)	0.0003 (1×10^{-7})	0.1051 (1×10^{-7})	-0.0060 (1×10^{-7})	0 (0)
4 (induite)	0.0002 (1×10^{-7})	-0.0350 (1×10^{-7})	0.0137 (-0.1)	0 (0)
5 (induite)	0 (1×10^{-7})	-0.0053 (0.1)	0.0171 (1×10^{-7})	0 (0)
6 (induite)	0 (0)	-0.0030 (1×10^{-7})	0.0441 (1×10^{-7})	0 (0)
Erreur globale moyenne nominale (4.2) : 0.0834 mm				
Erreur globale moyenne optimisée (4.2) : 0.1951 mm				

On constate donc que les différentes techniques pour retrouver les erreurs de modélisation ne donnent pas des résultats consistants, ni même près d'être semblables d'une technique à l'autre ou d'un test à l'autre. De plus, dans la majeure partie des cas, l'erreur globale moyenne optimisée n'est pas réduite par rapport à la valeur de l'erreur globale nominale. Il est arrivé que ce soit le cas, selon l'amplitude de l'erreur et les paramètres erronés, mais rien dans les résultats ne montrait de la constance, cependant les meilleurs résultats étaient généralement obtenus avec la fonction *lsqnonlin* (algorithme *Levenberg-Marquardt*).

On confirme donc le problème de conditionnement de ce système linéaire, ce qui concorde avec les observations d'autres chercheurs (Hanqi, Motaghedi et Roth, 1999; Ikits et

Hollerbach, 1997). Pour améliorer les résultats, l'estimation des positions robot doit être améliorée. Les prochaines simulations ne comporteront pas les simplifications qui ont menés à la création de la matrice \mathbf{M} et l'impact de la linéarisation sera étudié.

4.3 Produits vectoriels non simplifiés

Les développements du chapitre 3 mènent vers l'utilisation d'une meilleure approximation de la position du robot en ne simplifiant pas les termes quadratiques de l'erreur lors des constructions vectorielles et en utilisant un degré plus élevé de linéarisation. Il sera donc possible d'évaluer l'impact des deux premiers degrés de linéarisation sur la qualité de la solution obtenue par l'optimisation de l'équation objective (3.12).

Les tableaux suivants présentent l'impact de la linéarisation de premier et second ordre sur les vecteurs créés à partir des positions estimées de 16 touches faites sur une surface parfaitement plane. Le vecteur d'erreur induite $\Delta\mathbf{p} = [\Delta\mathbf{a}^T, \Delta\mathbf{a}^T, \Delta\mathbf{d}^T, \Delta\boldsymbol{\theta}^T]^T$ est présenté au Tableau 4.5. Le Tableau 4.6 présente l'impact de la linéarisation sur les vecteurs coplanaires (*Voir* Équation (3.5)) qui sont créés à partir des équations (3.3), puis (3.4). La première colonne du tableau présente la norme de l'erreur entre la position réelle du robot et celle évaluée selon les dimensions nominales du robot pour chacun des vecteurs. La seconde colonne utilise l'équation (3.3) pour évaluer chacune des positions robot à partir desquelles les vecteurs seront calculés, alors que la dernière colonne utilise l'équation (3.4). Les tableaux suivants (*Voir* Tableau 4.7 et Tableau 4.8) présentent les mêmes comparaisons par rapport aux normes calculées (3.6), puis aux produits vectoriels de ces normes utilisés dans l'équation (3.11).

Tableau 4.5 Erreurs de modélisation induites pour tester l'influence de la linéarisation sur les positions estimées du robot

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1	-0.0001	0.0000	0.0000	0.0000
2	0.0005	0.1400	0.0000	0.0000
3	-0.0002	-0.6700	-0.9870	0.0000
4	0.0006	-0.4310	0.5630	0.0000
5	-0.0009	0.2100	-0.7520	0.0000
6	0.0000	0.0000	0.0000	0.0000

Tableau 4.6 Normes de l'erreur induite par la linéarisation de premier et second ordre dans l'équation (3.5)

Norme de l'erreur avant linéarisation (mm)	Norme de l'erreur de linéarisation selon (3.3) (mm)	Norme de l'erreur de linéarisation selon (3.4) (mm)
1.1729×10^{-1}	1.1388×10^{-8}	3.1056×10^{-14}
6.6616×10^{-1}	1.1989×10^{-7}	4.0723×10^{-13}
2.1862×10^{-2}	6.8967×10^{-9}	2.5822×10^{-15}
9.2367×10^{-2}	1.8561×10^{-7}	2.2547×10^{-13}
5.3956×10^{-2}	2.4348×10^{-8}	3.9662×10^{-14}
1.2287×10^{-3}	3.3110×10^{-8}	1.4902×10^{-14}
3.5644×10^{-1}	2.3357×10^{-8}	9.1364×10^{-14}
8.0224×10^{-1}	2.0450×10^{-7}	1.2606×10^{-13}
3.0659×10^{-2}	2.7193×10^{-8}	4.6890×10^{-15}
2.9397×10^{-1}	3.7379×10^{-8}	5.7513×10^{-14}
4.2981×10^{-2}	1.9416×10^{-8}	2.5646×10^{-14}
2.2926×10^{-1}	1.7887×10^{-7}	4.2730×10^{-14}
2.0607×10^{-2}	4.4583×10^{-9}	3.1236×10^{-15}
4.5638×10^{-1}	4.7551×10^{-8}	8.4777×10^{-14}
7.6301×10^{-2}	2.9234×10^{-8}	3.8988×10^{-14}

Tableau 4.7 Vecteurs (Voir Équation (3.5)) issus des positions réelles et estimées à partir du vecteur d'erreur initial et de la linéarisation de premier, puis de second ordre

Norme de l'erreur avant linéarisation (mm)	Norme de l'erreur de linéarisation selon (3.3) (mm)	Norme de l'erreur de linéarisation selon (3.4) (mm)
3.5660×10^{-4}	1.7413×10^{-11}	7.6041×10^{-17}
2.1030×10^{-4}	7.0675×10^{-12}	4.0127×10^{-17}
2.5594×10^{-5}	6.8836×10^{-14}	4.1947×10^{-18}
4.6377×10^{-5}	3.1298×10^{-11}	9.6273×10^{-18}
5.9833×10^{-5}	1.6591×10^{-11}	1.2181×10^{-17}
1.2640×10^{-4}	2.8826×10^{-12}	2.6921×10^{-17}
6.0201×10^{-5}	2.4526×10^{-12}	1.3375×10^{-17}
1.5537×10^{-5}	1.1760×10^{-11}	2.9568×10^{-18}
3.7359×10^{-5}	8.6134×10^{-12}	7.5016×10^{-18}
1.2228×10^{-4}	1.3529×10^{-11}	2.4674×10^{-17}
4.9052×10^{-5}	3.1654×10^{-12}	9.4334×10^{-18}
8.1835×10^{-6}	1.1273×10^{-12}	1.2118×10^{-18}
7.7824×10^{-5}	4.2306×10^{-13}	1.5871×10^{-17}
2.3344×10^{-4}	1.2960×10^{-11}	4.7056×10^{-17}

Tableau 4.8 Normales (Voir Équation (3.6)) issues des positions réelles et estimées à partir du vecteur d'erreur initial et de la linéarisation de premier, puis de second ordre

Norme de l'erreur avant linéarisation (mm)	Norme de l'erreur de linéarisation selon (3.3) (mm)	Norme de l'erreur de linéarisation selon (3.4) (mm)
2.5621×10^{-5}	2.9672×10^{-12}	7.5807×10^{-18}
9.2117×10^{-5}	6.1202×10^{-12}	1.8788×10^{-17}
1.3337×10^{-4}	2.8515×10^{-11}	2.4859×10^{-17}
1.4567×10^{-6}	3.8465×10^{-12}	2.5882×10^{-19}
3.6010×10^{-4}	6.1513×10^{-12}	7.5304×10^{-17}
2.8078×10^{-5}	1.5325×10^{-13}	5.0408×10^{-18}
5.5547×10^{-5}	4.5520×10^{-12}	1.4573×10^{-17}
6.8500×10^{-6}	1.1928×10^{-12}	1.3267×10^{-18}
5.4629×10^{-5}	3.5945×10^{-11}	1.0008×10^{-17}
1.7014×10^{-5}	4.5236×10^{-12}	3.7750×10^{-18}
7.0504×10^{-5}	7.8564×10^{-12}	1.3288×10^{-17}
7.5585×10^{-5}	2.4129×10^{-12}	1.5864×10^{-17}
7.0248×10^{-5}	1.0950×10^{-11}	1.3293×10^{-17}

Ces tableaux montrent que malgré l'utilisation du vecteur d'erreur induit dans les équations (3.3) et (3.4), la norme de l'erreur des produits vectoriels des normales est très faible, mais non nulle. Comme l'optimisation se fait en tentant de minimiser ces valeurs, l'erreur que l'on retrouve ici pourrait biaiser les résultats. Malgré tout, on constate une diminution d'un facteur de 10^{-6} de l'erreur lorsqu'on tient compte de la linéarisation de premier ordre (deuxième colonne de chacun des tableaux) et encore une diminution de l'erreur du même facteur lorsque la matrice hessienne est utilisée (dernière colonne de chacun des tableaux).

Cette amélioration permet d'ailleurs de mettre à l'épreuve un tout nouvel algorithme de calibration impliquant le calcul des positions robot estimées à l'aide de l'équation (3.4) et de

la fonction objective présentée à l'équation (3.12). Un test avec 36 touches et R mis à 0.01 permet de démontrer la minimisation de l'erreur globale moyenne d'un robot (Voir Tableau 4.9). Dans ce cas, l'optimisation est faite pour les paramètres erronés seulement.

Tableau 4.9 Exemple d'amélioration de la précision par optimisation de l'équation (3.12)

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	0 (0.002)	0 (0)	0 (0)	0 (0)
2 (induite)	-0.005 (-0.005)	0 (0)	0 (0)	0 (0)
3 (induite)	0 (0)	0 (0.1)	0 (-0.6)	0.0008 (0.001)
4 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
5 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
6 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
Erreur globale moyenne nominale (4.2) : 1.1515 mm				
Erreur globale moyenne optimisée (4.2) : 0.7246 mm				

Même si l'erreur de modélisation n'est pas exactement retrouvée, l'erreur globale moyenne est diminuée. En supposant que la calibration est faite autour de la position où le robot sera amené à travailler, sa précision sera améliorée (Voir Figure 4.1).

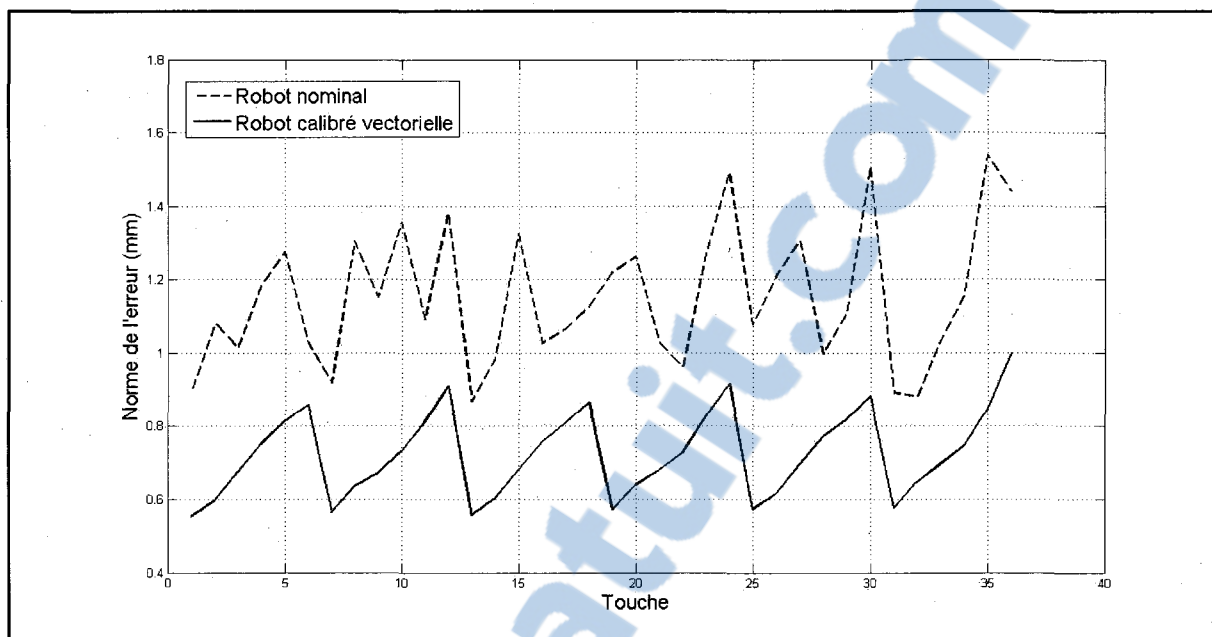


Figure 4.1 Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.12).

Par contre avec un vecteur d'erreur plus important, plus près de ce qui pourrait être retrouvé en environnement industriel, les performances de l'algorithme diminuent drastiquement. En effet, l'erreur moyenne globale n'est pas diminuée (Voir Tableau 4.10 et Figure 4.2), même si l'optimisation n'est faite qu'en fonction des paramètres erronés.

Tableau 4.10 Exemple d'échec de l'algorithme
d'optimisation avec l'équation (3.12)

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	0 (0.0002)	0 (0)	0 (0)	0 (1 x 10 ⁻⁷)
2 (induite)	-0.0010 (-0.005)	0 (0.1)	0 (0)	0.0002 (1 x 10 ⁻⁷)
3 (induite)	0.0002 (0.004)	0 (-0.9)	0 (-0.6)	0.0001 (0.001)
4 (induite)	-0.0012 (1 x 10 ⁻⁷)	0 (1 x 10 ⁻⁷)	0 (1 x 10 ⁻⁷)	-0.0005 (1 x 10 ⁻⁷)
5 (induite)	0.0018 (1 x 10 ⁻⁷)	0 (1 x 10 ⁻⁷)	0 (1 x 10 ⁻⁷)	-0.0002 (1 x 10 ⁻⁷)
6 (induite)	0 (0)	0 (1 x 10 ⁻⁷)	0 (-0.02)	0 (1 x 10 ⁻⁷)
Erreur globale moyenne nominale (4.2) : 1.5703 mm				
Erreur globale moyenne optimisée (4.2) : 1.9945 mm				

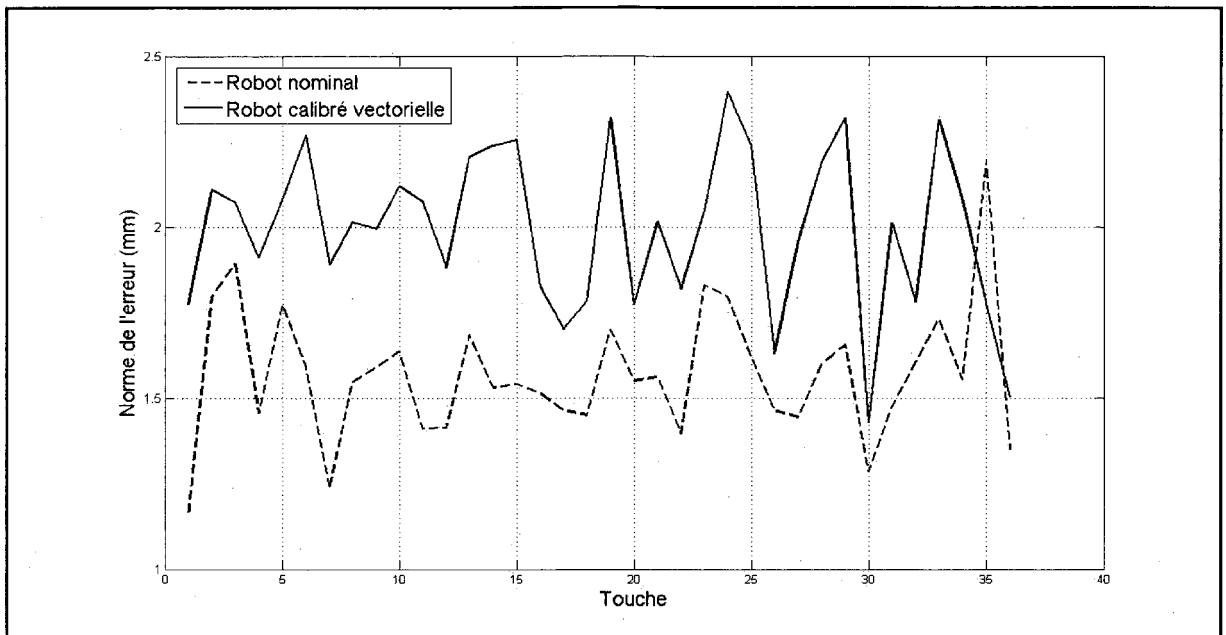


Figure 4.2 Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.12).

À la lumière de ces résultats, on observe que même si l'erreur induite est faible par rapport à ce qui se retrouve en industrie, l'optimisation ne permet pas d'améliorer la précision du robot. Il apparaît donc que cette technique ne pourra pas être mise en œuvre dans l'application d'ébavurage robotisé étant donné les limites de ses performances.

4.4 Calcul du plan moyen

Tel que présenté au chapitre 3, il est possible de bâtir une équation objective en construisant mathématiquement un plan moyen autour des touches estimées. On minimise ensuite la somme des distances entre les points et le plan moyen en optimisant selon l'erreur de modélisation.

Par contre, les résultats d'optimisation avec la fonction objective présentée à l'équation (3.17), où les positions robot sont estimées à partir de l'équation (3.4) ne démontrent pas plus de potentiel que la technique explorée précédemment. Avec des erreurs de faible amplitude imputées à quelques paramètres bien choisis, pour lesquels l'optimisation est faite, les

résultats sont intéressants, mais lorsque l'optimisation se fait sur un nombre croissant de paramètres, les résultats deviennent rapidement hasardeux. Le Tableau 4.11 montre les résultats lorsque cette technique est utilisée dans les mêmes conditions que pour le test présenté au Tableau 4.9.

Tableau 4.11 Exemple d'amélioration de la précision
par optimisation de l'équation (3.17)

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	0.0001 (0.002)	0 (0)	0 (0)	0 (0)
2 (induite)	-0.0049 (-0.005)	0 (0)	0 (0)	0 (0)
3 (induite)	0 (0)	-0.0789 (0.1)	0.0001 (-0.6)	-0.0002 (0.001)
4 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
5 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
6 (induite)	0 (0)	0 (0)	0 (0)	0 (0)
Erreur globale moyenne nominale (4.2) : 1.1515 mm				
Erreur globale moyenne optimisée (4.2) : 0.9543 mm				

L'erreur globale moyenne est améliorée (*Voir* Figure 4.3), malgré l'identification plus ou moins exacte de l'amplitude des erreurs de modélisation, tout comme le permettait la technique précédente.

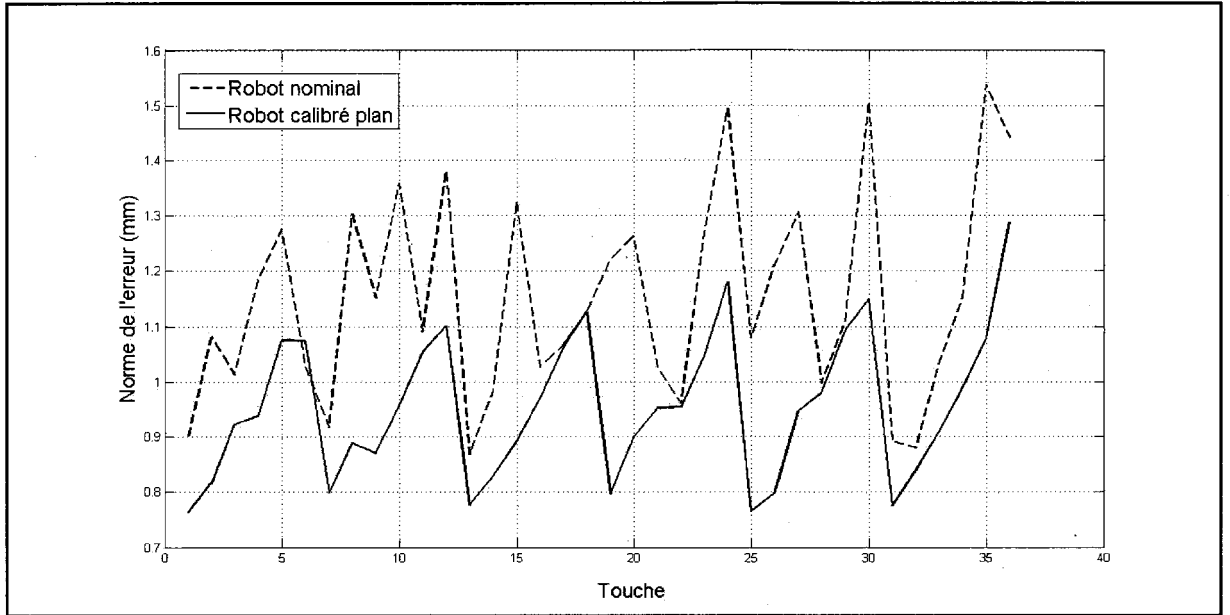


Figure 4.3 Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.17).

Le Tableau 4.12 montre les résultats obtenus lorsque le test présenté au Tableau 4.10 est repris avec cette nouvelle approche de calibration.

Tableau 4.12 Exemple d'échec de l'algorithme
d'optimisation avec l'équation (3.17)

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1 (induite)	-0.0001 (0.0002)	0 (0)	0 (0)	0.0002 (10^{-7})
2 (induite)	-0.0153 (-0.005)	0 (0.1)	0 (0)	0.0003 (10^{-7})
3 (induite)	0.0063 (0.004)	-0.0036 (-0.9)	0 (-0.6)	-0.0006 (0.001)
4 (induite)	0.0073 (10^{-7})	-0.0011 (10^{-7})	0.0031 (10^{-7})	0.0064 (10^{-7})
5 (induite)	0.0006 (10^{-7})	0.0032 (10^{-7})	-0.0002 (10^{-7})	-0.0001 (10^{-7})
6 (induite)	0 (0)	0 (10^{-7})	0.0002 (-0.02)	0 (10^{-7})
Erreur globale moyenne nominale (4.2) : 1.5703 mm				
Erreur globale moyenne optimisée (4.2) : 5.3338 mm				

Dans ce cas, l'erreur globale moyenne est plus importante après l'optimisation qu'avant ce qui n'est évidemment pas désiré (Voir Figure 4.4). On constate donc que cette technique ne permet pas de meilleurs résultats que la technique présentée précédemment.

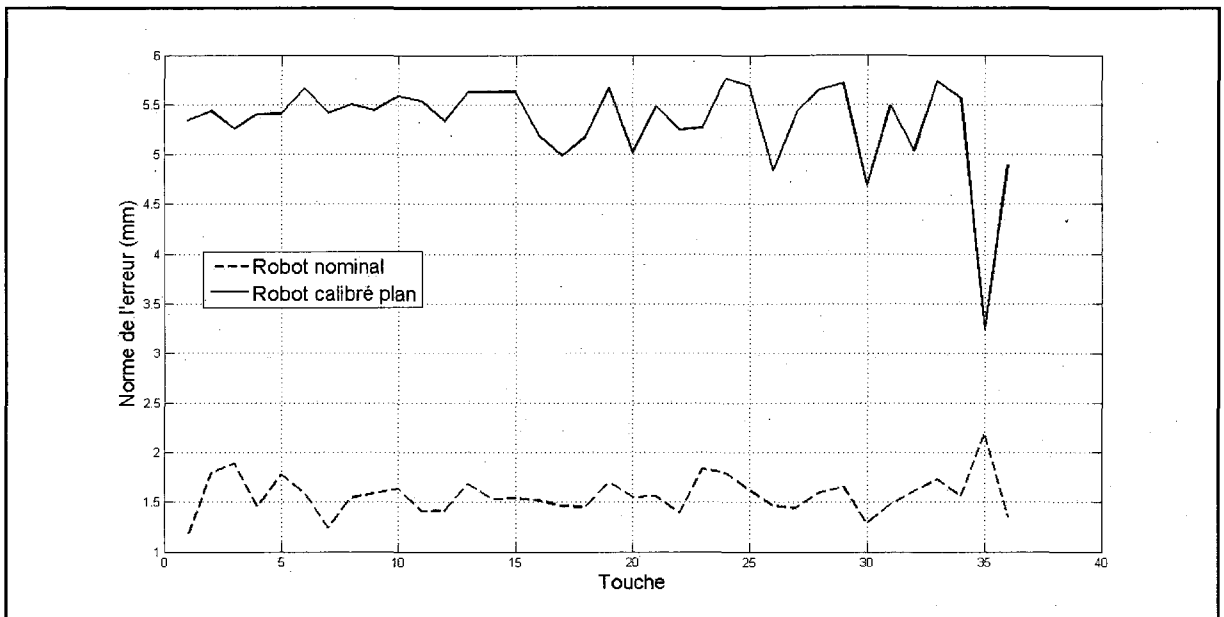


Figure 4.4 Comparaison de l'erreur de positionnement avant et après optimisation selon l'équation (3.17).

En fait, si les deux dernières techniques sont comparées, celle présentée précédemment offre généralement de meilleures performances. Bien que des tests comparatifs complets n'aient pas été conduits, la Figure 4.5 montre la norme de l'erreur entre la position réelle du robot et celle avant et après optimisation, selon la technique de calibration choisie. L'erreur de modélisation utilisée pour obtenir ces valeurs est celle présentée aux Tableau 4.9 et au Tableau 4.11.

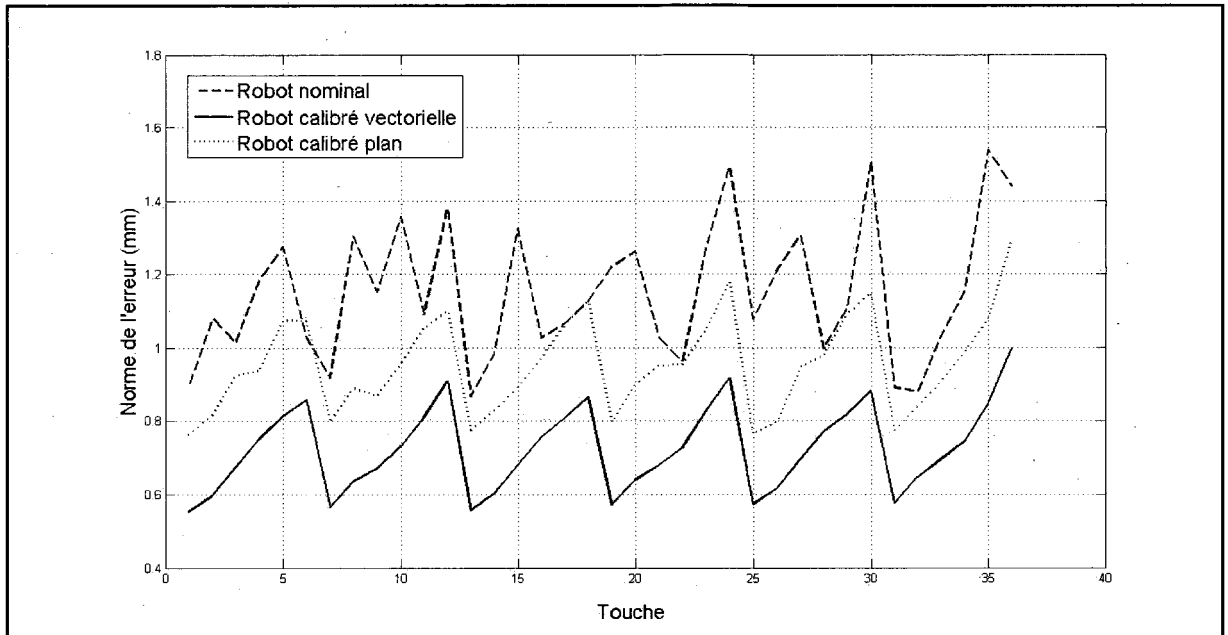


Figure 4.5 Norme de l'erreur entre la position réelle du robot et celle estimée selon trois configurations différentes du robot.

En ce qui concerne la technique utilisant la construction d'un plan moyen, différentes situations ont aussi été explorées : position du plan inconnue ou connue et utilisation de trois plans perpendiculaires dont les positions sont connues ou inconnues. Comme l'objectif final est de mettre en œuvre la technique en industrie, peu d'accent a été mis sur les techniques avec plusieurs plans étant donné qu'il n'était pas trivial d'avoir accès à ce genre de structure, avec la précision requise. Il a tout de même été possible d'apprécier l'amélioration générale de l'observabilité des paramètres de modélisation lorsque plusieurs plans étaient utilisés. Par contre, certains paramètres, comme α_3 et α_4 , dont l'identification était très bonne avec un seul plan, souffraient de l'ajout de deux plans puisqu'ils biaisaient l'optimisation. Finalement, lorsque la position des plans est connue, l'identification des paramètres en est grandement améliorée, ce qui était le résultat attendu. Les paramètres qui n'étaient pas observables avec un seul plan le sont toutefois demeurés.

Dans un autre ordre d'idées, les analyses faites ont permis de remarquer l'impact constant de certains paramètres de longueurs sur l'erreur en position des touches. Par exemple, pour une

erreur donnée sur un seul paramètre a_i ou d_i du modèle, la différence entre la position calculée à partir du robot réel et celle calculée à partir du robot nominal sera constante et égale à l'erreur de modélisation (Voir Figure 4.6).

Tableau 4.13 Exemple d'erreur sur un seul paramètre de longueur

i	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000
3	0.0000	1.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000

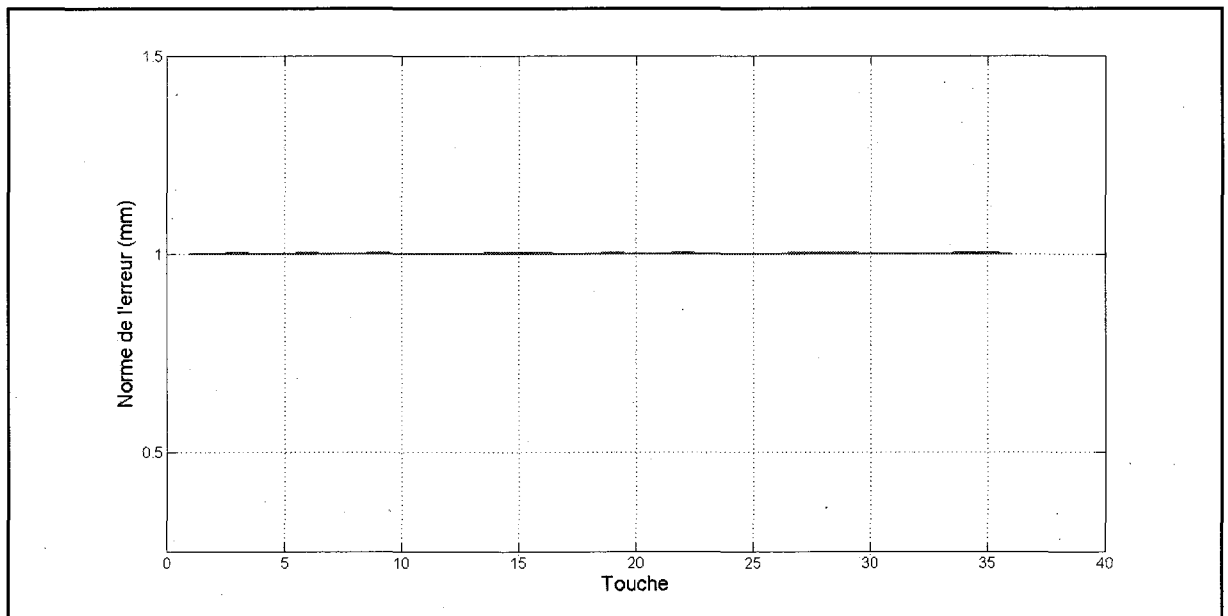


Figure 4.6 Norme de l'erreur entre la position réelle du robot et celle estimée selon le modèle nominal pour l'erreur de modélisation présentée au Tableau 4.13.

En définitive, aucune des deux techniques ne présentent des résultats assez prometteurs pour qu'elles soient mises en pratique et intégrées à l'application d'ébavurage robotisé. En dernier recours, une comparaison entre différents algorithmes de calibration de TCP d'outil sera tentée pour valider la précision relative d'un des outils qui sera utilisé dans l'application.

4.5 Technique de calibration du TCP de l'outil

La dernière technique de calibration présentée au chapitre 3 propose de ne calibrer que les dimensions du TCP de l'outil par des touches effectuées sur une sphère. L'équation d'une sphère moyenne est trouvée selon la position du TCP de l'outil et la somme des distances entre les touches et la sphère moyenne est minimisée en fonction des dimensions de l'outil.

4.5.1 Résultats de simulation

Les équations développées au chapitre précédent ont été mises en œuvre dans différentes simulations pour des résultats très concluants. Le Tableau 4.14 présente les résultats d'un de ces tests avec 12 touches simulées sur la sphère (*Voir Figure 4.7*).

Tableau 4.14 Exemple d'erreur sur un seul paramètre de longueur

	X (mm)	Y (mm)	Z (mm)
Dimensions réelles	1.834	2.040	299.423
Dimensions nominales	0	0	300
Dimensions retrouvées	1.384	2.061	299.987

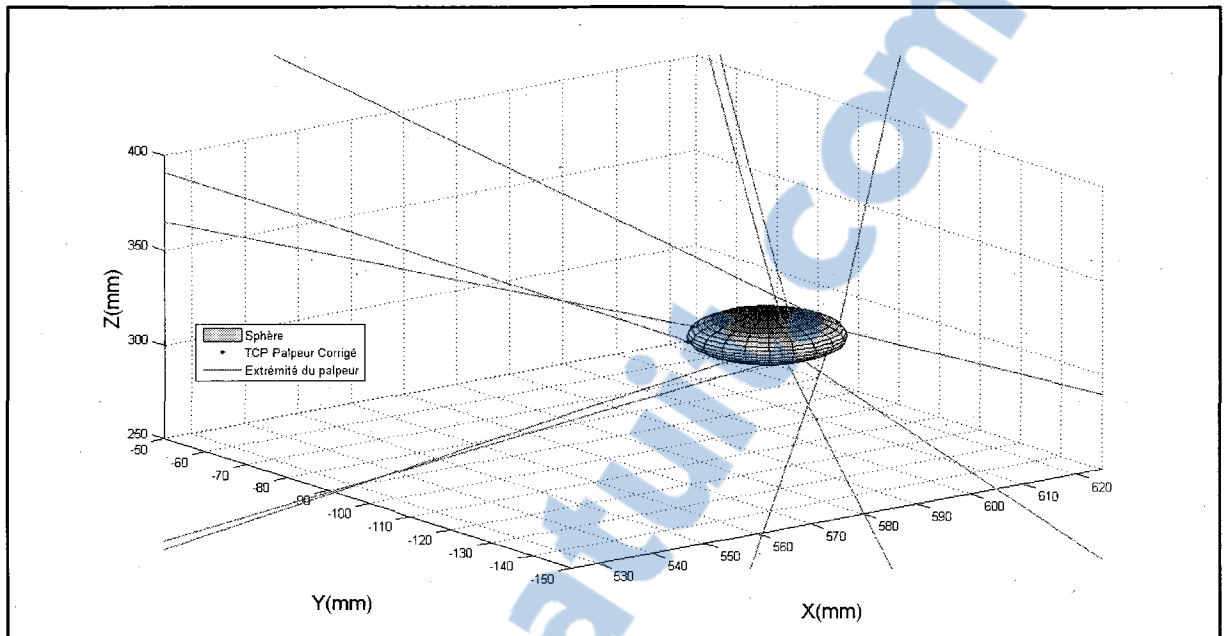


Figure 4.7 Représentation des touches simulées sur la sphère.

Lorsque des touches sont simulées sur une surface plane située près de la sphère, on constate que la précision du robot a été grandement améliorée (*Voir Figure 4.8*), **malgré le fait que le robot simulé contienne des erreurs de modélisation**. Il est à noter que pour obtenir de bons résultats, l'ampleur maximale des erreurs de modélisation du robot a due être diminuée d'un facteur dix.

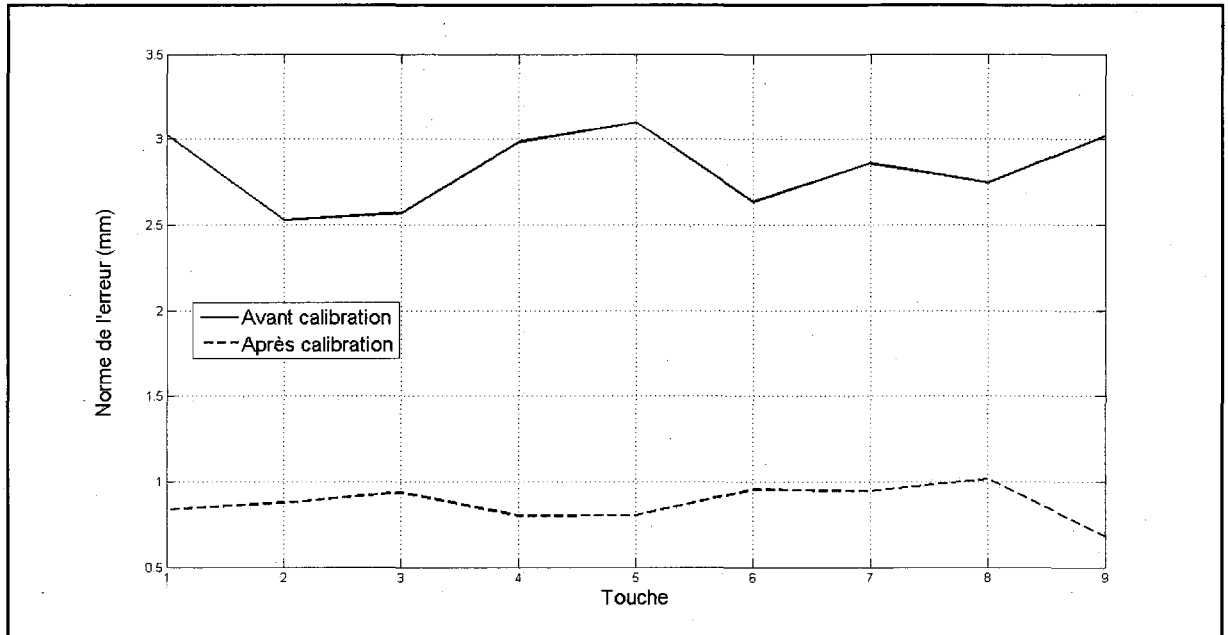


Figure 4.8 Norme de l'erreur de positionnement sur une surface plane selon les dimensions nominales et les dimensions optimisées de l'outil.

4.5.2 Résultats de la mise en œuvre

Le robot d'ABB qui est utilisé pour l'application d'ébavurage robotisé propose sa propre procédure pour calibrer les TCP des outils qui seront attachés à l'extrémité {E} du robot. Elle consiste à installer l'outil au bout du robot, puis à positionner son TCP au même endroit avec différentes orientations. L'algorithme utilisé n'est pas dévoilé par la compagnie, mais nécessite de quatre à neuf lectures différentes, selon la précision désirée.

Un palpeur Renishaw déjà à notre disposition sera l'outil à calibrer (*Voir Figure 4.9*).

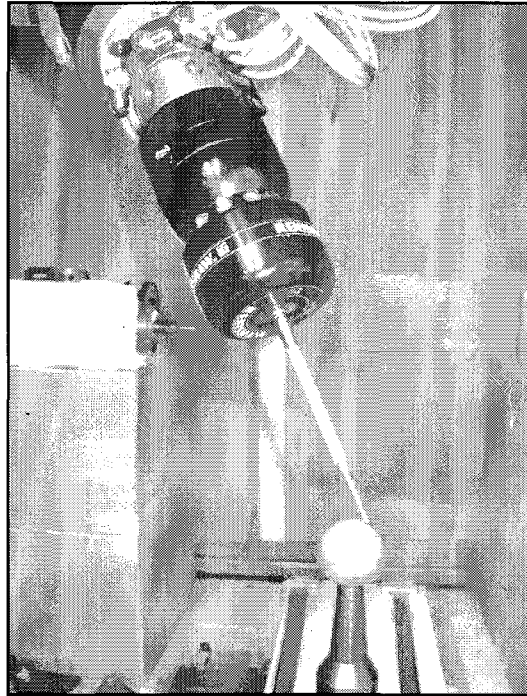


Figure 4.9 Palpeur lors de la calibration.

La valeur du TCP trouvée selon la technique présentée au chapitre 3 varie légèrement selon le nombre de points utilisés, leur position et orientation. La Figure 4.10 montre la variation de la valeur de chacune des trois composantes (X, Y, Z) du TCP en fonction du nombre de touches faites. Un TCP a été calculé avec 7, 8, 9, ..., 20 et 21 touches. Chacune des touches a été faite avec une orientation différente pour enrichir les données.

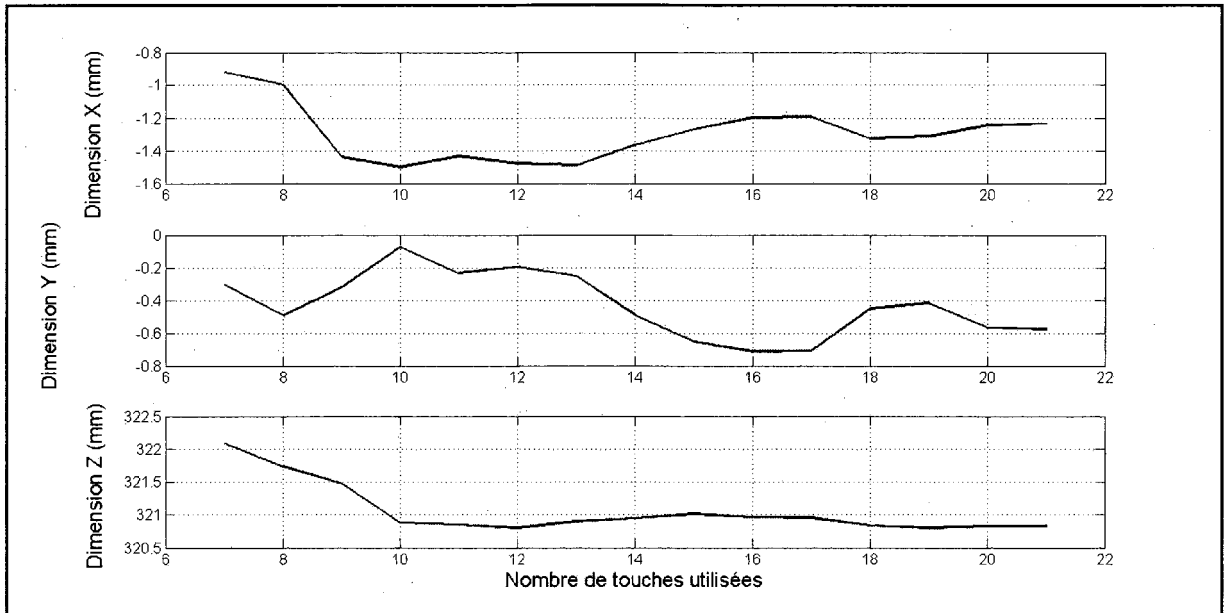


Figure 4.10 Évolution des dimensions du TCP selon le nombre de touches utilisées pour l'algorithme de calibration.

D'abord, on remarque une convergence des dimensions du TCP vers une valeur finale. Les touches 1 à 12 ont été prises en paires de positions le plus diamétralement opposées possible. Par contre, les touches 13 à 17 ont été prises sur la même hémisphère, puis les touches suivantes ont été prises sur l'autre pour compenser. On remarque d'ailleurs une transition plus marquée entre les touches 17 et 18 qui découle de ces manipulations.

Comme plusieurs TCP sont calculés, ils seront tous comparés au TCP obtenu par la calibration d'ABB et à celui obtenu en n'utilisant que les dimensions nominales des composantes du palpeur. On utilise la plaque de verre dont il est question à la première section (*Voir* Figure 4.11) de ce chapitre pour évaluer la meilleure solution.

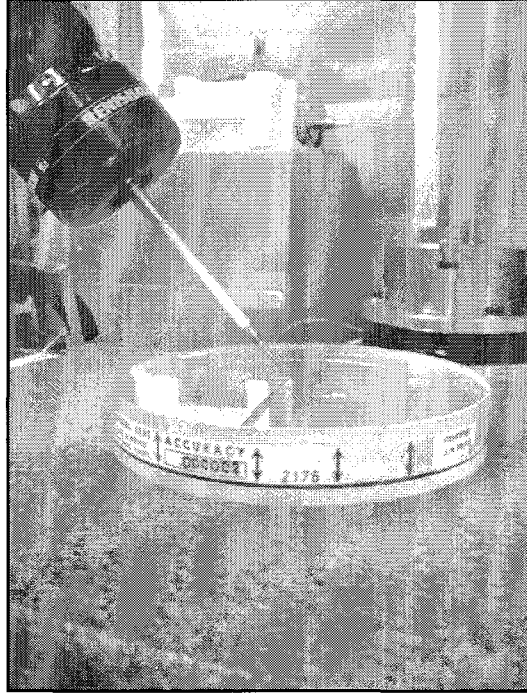


Figure 4.11 Plaque de verre utilisée pour l'évaluation des TCP.

La Figure 4.12 permet de comparer les résultats selon les différents TCP trouvés.

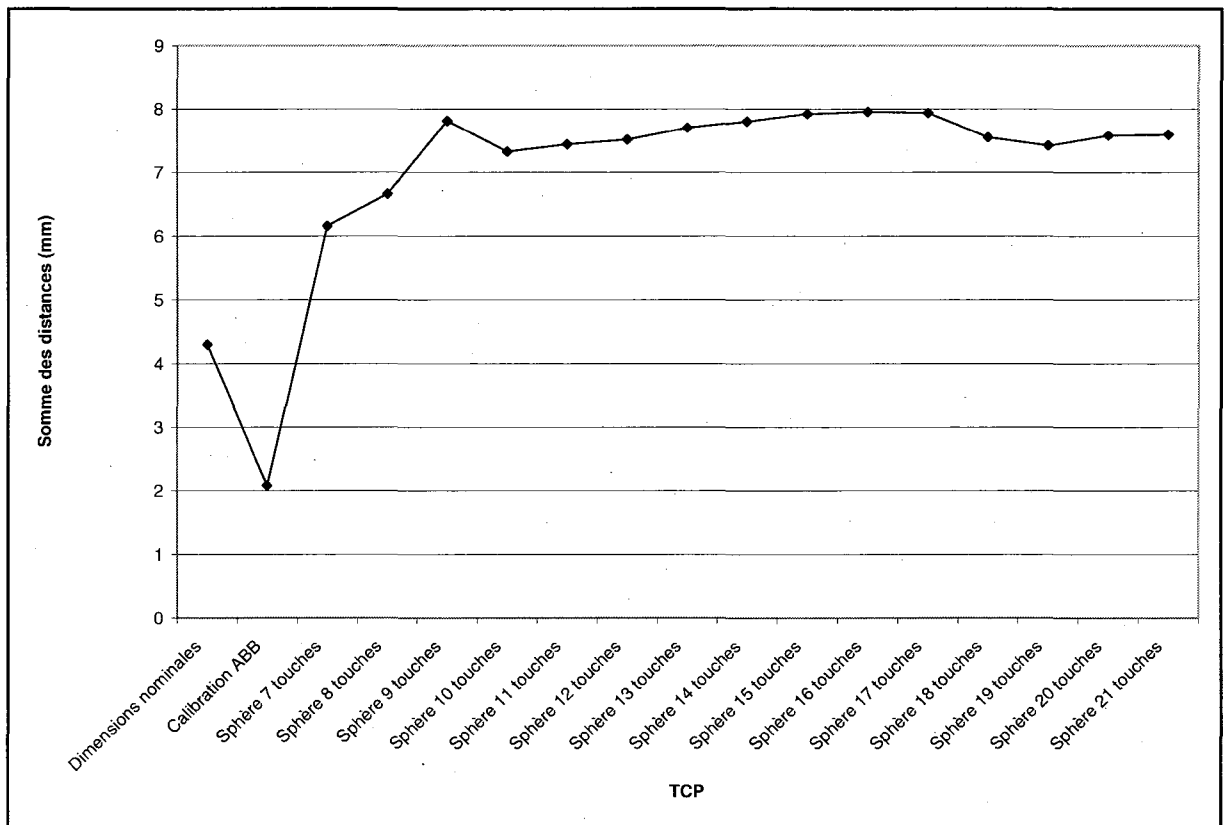


Figure 4.12 Évolution des dimensions du TCP selon le nombre de touches utilisées pour l'algorithme de calibration.

Il est donc clair que la calibration faite avec la technique d'ABB donne les meilleurs résultats avec une erreur totale de près de 2 mm sur les 15 touches faites sur la plaque de verre, soit une moyenne de 0.1333 mm d'erreur par touche, par rapport au plan moyen calculé. On constate aussi que le résultat obtenu avec les dimensions nominales est meilleur que ceux obtenus avec chacun des TCP retrouvés par optimisation. Il faut voir que la position du centre de la sphère dans l'algorithme est complètement estimée à partir du TCP de l'outil. Il est possible que cette estimation fasse dévier les solutions obtenues et entraîne les performances observées. Si la position du centre de la sphère était connue et utilisée dans l'algorithme, les performances seraient probablement bien meilleures. Lors du développement de l'application d'ébavurage robotisé, c'est donc la calibration selon la technique d'ABB qui devra être utilisée.

4.6 Conclusion

Tel qu'expliqué dans ce chapitre, aucune des techniques de calibration du modèle du robot n'offre des résultats assez stables et fiables pour être utilisée dans l'application d'ébavurage robotisé. De manière générale, il apparaît que l'effet des non-linéarités du modèle du robot est trop important pour permettre de retrouver les erreurs de modélisation selon les techniques simulées. Nous avons cependant un comparatif intéressant positionnant la technique utilisant les constructions vectorielles comme étant plus performante que celle utilisant un plan moyen. Ceci peut s'expliquer par le fait que le plan moyen est construit à partir de tous les points à la fois. Ce sont ces mêmes points qui sont ensuite réutilisés dans la fonction objective. L'impact de l'erreur de modélisation en est réduit. La calibration de l'outil selon la technique d'ABB sera donc le moyen le plus efficace pour améliorer la précision de l'opération parmi ceux présentés dans ce chapitre.

La problématique de précision demeure donc entière et devra être solutionnée par les différents dispositifs qui seront introduits dans la cellule robotisée. Entre autres, la caméra et le capteur de force devront être utilisés de façon à contourner l'imprécision du robot et à éliminer le contrôle en position lors de certaines séquences d'opérations.

CHAPITRE 5

CARACTÉRISATION DES COMPOSANTES DE L'ÉTUDE DE CAS

Dans ce chapitre, les éléments qui font partie de l'application d'ébavurage robotisé seront présentés et caractérisés. Le rôle de chacun des modules, leur répétabilité, leur précision, leur technique de calibration et les modes de communication respectifs seront expliqués. Telle que mentionnée au chapitre précédent, la technique de calibration fournie par ABB est celle retenue pour calibrer les TCP des outils. Elle sera mise en application pour tous les outils qui seront utilisés dans la cellule robotisée.

5.1 Robot IRB-140 d'ABB

Le robot utilisé dans l'application d'ébavurage robotisé avec contrôle de force est le IRB-140 d'ABB et a été présenté dans le second chapitre du présent document (*Voir* Figure 2.1 et Figure 2.2). Il a été choisi parmi l'ensemble des robots disponibles, entre autres, pour son poids (98 kg), sa répétabilité ($\pm 30 \mu\text{m}$), sa portée (810 mm) et la charge qu'il peut transporter (6 kg), ce qui correspond aux critères de sélection qui ont été établis en début de projet.

Un capteur de force est fixé à l'extrémité {E} du robot (*Voir* Figure 5.1). Il est connecté au contrôleur du robot et intégré directement dans la boucle de contrôle par ABB. Plus de détails sur les performances du capteur de force seront donnés dans le chapitre 6. Le changeur d'outil, permettant de changer automatiquement les outils, est fixé à l'extrémité du capteur de force. Ces dispositifs sont reliés entre eux et au robot par différentes plaques d'adaptation faites sur mesure.

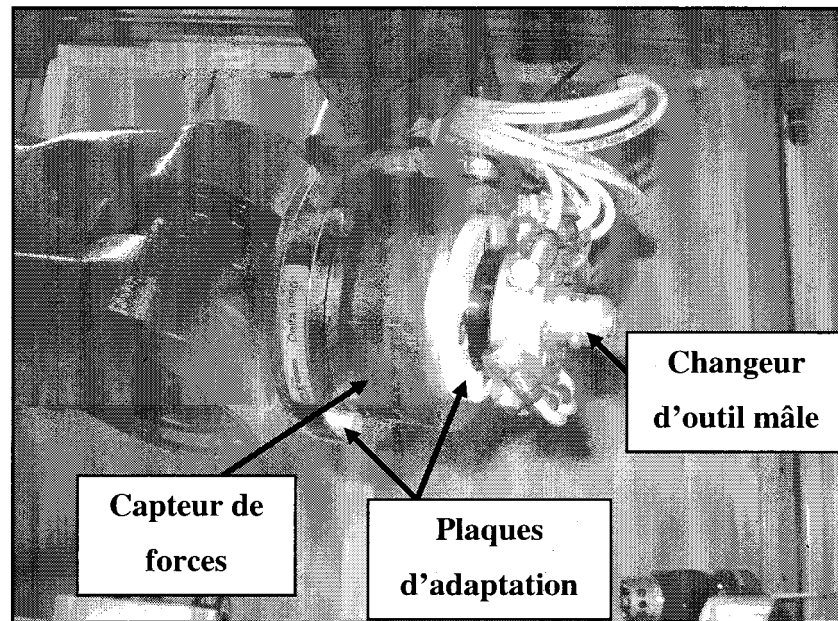


Figure 5.1 Assemblage du capteur de force et changeur d'outil.

Le robot est considéré comme le maître dans l'application et son contrôleur commande toutes les opérations qui doivent être faites. Sa programmation est réalisée via le logiciel d'ABB, RobotStudio®, en langage RAPID¹.

5.1.1 Répétabilité et précision

Selon les données du fabricant, le IRB-140 d'ABB a une répétabilité en position de $\pm 30 \mu\text{m}$. Cette valeur a été calculée selon la moyenne obtenue lors de l'exécution des tests définis dans la norme ISO9283 qui standardise les techniques pour calculer les performances des manipulateurs industriels.

Cette valeur a pu être corrélée avec différents tests dont un qui a été conduit avec le palpeur. Trois touches ont été faites avec le palpeur sur une pièce fixée dans la cellule. Ces mêmes

¹ RAPID est un langage de programmation propre à ABB, pour une référence à jour il faut contacter un vendeur ABB.

trois touches ont été faites à 20 reprises et la variation autour de la position moyenne obtenue est à l'intérieur de $\pm 30 \mu\text{m}$.

Il a été impossible de soumettre le robot utilisé à un appareil de mesure plus précis que lui-même pour évaluer sa précision. Il est commun d'estimer la précision à 10 fois la répétabilité du robot, ce qui correspond aux observations faites tout au long de l'utilisation de ce dernier. D'ailleurs, c'est cette imprécision qui justifie l'ajout de différents périphériques et différentes manipulations pour assurer une constance dans le travail qui doit être effectué par le robot.

En ce qui concerne le capteur de force, il est capable de lire des forces de $\pm 660 \text{ N}$. Sa résolution dépend du système utilisé pour lire le signal, ce qui est déjà intégré dans l'ensemble fourni par ABB. Selon toute vraisemblance, il s'agit d'une carte d'acquisition de données qui permet une résolution de 0.125 N , en X et Y, et de 0.25 N en Z. Ces valeurs n'ont pu être vérifiées puisque le bruit présent en tout temps lors de la lecture du capteur dépassait amplement la résolution.

Finalement, un jeu mécanique a été observé au niveau des interfaces du capteur de force et du changeur d'outil. Ce jeu génère un mouvement répétable d'une amplitude allant jusqu'à 2 mm par axe à l'extrémité de l'outil rotatif. L'application doit être développée en fonction de cette réalité qui force l'introduction d'une compensation expliquée à la section 6.2.3.

5.1.2 Technique de calibration

Ce type de robot utilise des *resolvers* comme capteur de position pour les articulations. Ce dispositif est une résistance variable qui varie selon la position de l'articulation. Par contre, pour un tour de l'articulation, le *resolver* fait plusieurs tours. Il faut donc utiliser un compteur de tours. Lors de la calibration, c'est ce compteur de tours qu'il faut remettre à zéro. Pour ce faire, des marques sur le robot permettent d'aligner les articulations selon la position initiale du robot. Cela étant fait, il ne suffit que d'exécuter une routine résidant dans le contrôleur et les compteurs sont remis à zéro. Finalement, lorsque le robot est à sa position initiale les

valeurs des *resolvers* sont connues et ne changent jamais étant donné la construction mécanique du robot. Ces valeurs sont emmagasinées dans un fichier qui accompagne le robot à sa livraison et sur sa plaque signalétique. Elles peuvent être rétablies dans le contrôleur si nécessaire.

5.1.3 Communication

Le contrôleur du robot permet plusieurs modes de communication : sériel (RS-232), *ethernet*, signaux analogiques entrants et sortants (AI/O), signaux numériques entrants et sortants (DIO), etc. Il sera donc possible de communiquer avec tous les périphériques qui sont présentés dans ce chapitre.

Le changeur d'outil est d'ailleurs contrôlé par un DIO, qui commande la valve pneumatique permettant la préhension ou le relâchement des différents outils.

5.2 Palpeur

Le palpeur utilisé dans la cellule est muni d'un adaptateur de 75.00 mm et d'un *styli* de 21.00 mm dont le rubis est de 1.00 mm de diamètre (*Voir Figure 5.2*). Il a comme caractéristique d'être sans fils. L'information est donc transmise du palpeur à un capteur qui est connecté au contrôleur du robot (*Voir Figure 5.3*). Le palpeur est fixé sur un changeur d'outil.

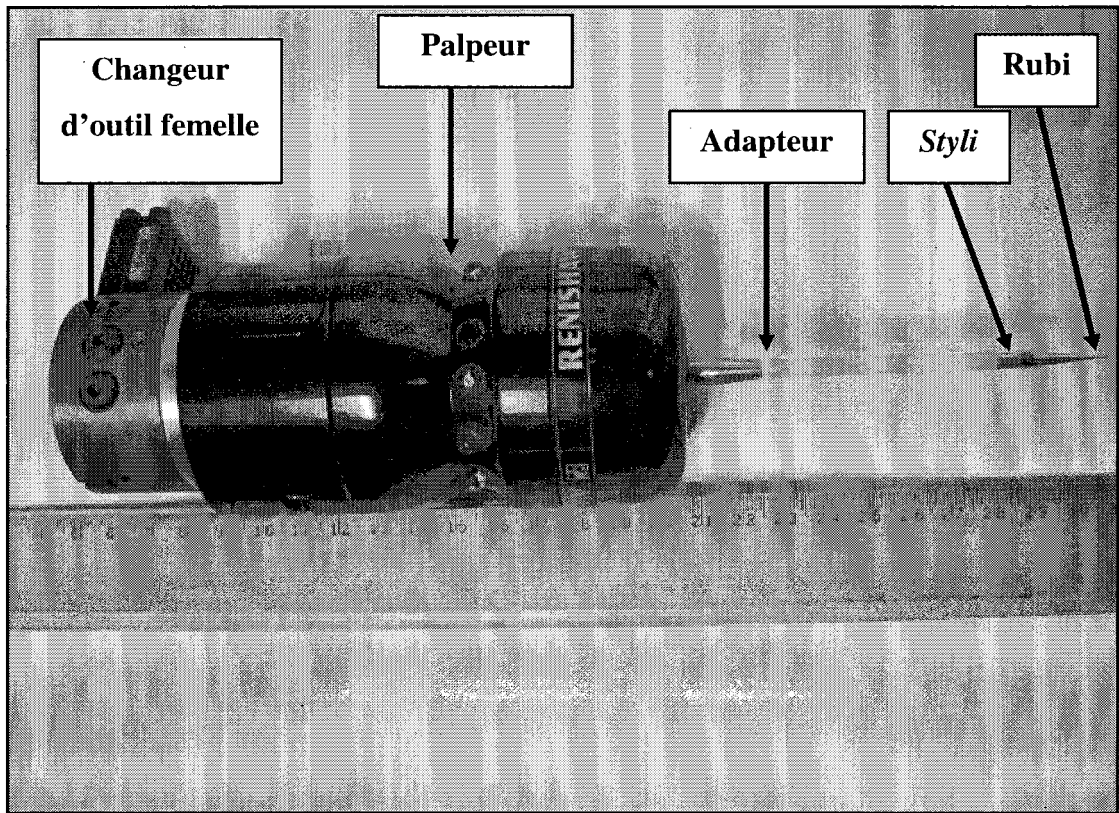


Figure 5.2 Assemblage du palpeur Renishaw, de l'adaptateur et du *styli*.



Figure 5.3 Capteur du palpeur.

Cet outil sera monté à l'extrémité du robot et permettra de faire certains tests de précision, de définir la position des objets dans la cellule et de définir le positionnement des pièces lors de l'exécution des routines d'ébavurage.

5.2.1 Répétabilité et précision

Le palpeur a une répétabilité de $\pm 1.75 \mu\text{m}$ dans la pire situation, c'est-à-dire lorsque le contact déplace le *styli* dans plusieurs axes en même temps (avec une longueur totale d'adaptateur et de *styli* de 100 mm). Telle que présentée au chapitre 3, une évaluation qualitative de la précision de l'ensemble robot-palpeur a été établie pour différentes valeurs de TCP du palpeur et celui trouvé à l'aide de la technique proposée par ABB donne les meilleurs résultats. Cependant, comme pour le robot, il n'a pas été possible de quantifier la précision de cet ensemble, faute d'un instrument de mesure assez précis.

5.2.2 Technique de calibration

Aucune calibration n'est nécessaire pour l'outil lui-même. Il est en fait considéré comme étant parfaitement précis par rapport à notre application et est lui-même utilisé comme référence lors de certains tests de précision et de répétabilité. Par contre, la calibration du TCP du palpeur se fera selon la technique de calibration de TCP proposée par ABB.

5.2.3 Communication

Cet outil transmet quelques informations par les DI/O. Les signaux utilisés sont l'activation ou non de l'outil, le signal indiquant un contact et le signal indiquant que la batterie est faible.

5.3 Axes auxiliaires

Deux axes auxiliaires sont disponibles dans la cellule robotisée. Le premier est linéaire, le second rotatif et il se déplace sur le chariot du premier. Ces derniers sont contrôlés par un contrôleur dédié à cette tâche.

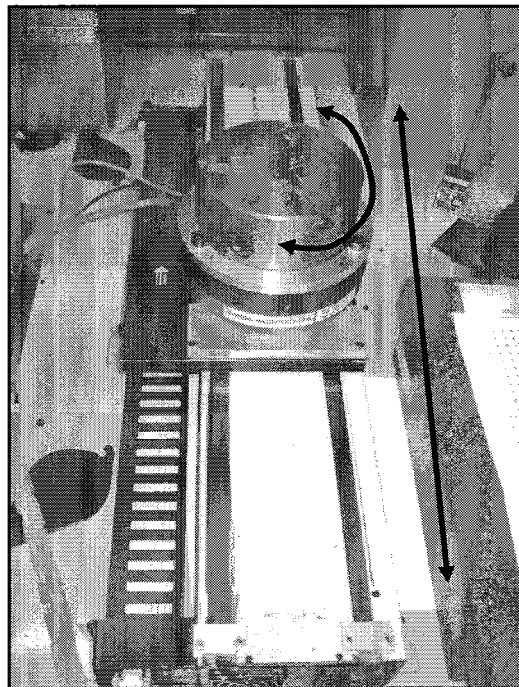


Figure 5.4 Montage des axes auxiliaires.

Ces axes ajoutent deux DDL supplémentaires au robot. Ils seront utilisés pour amener la pièce dans l'espace de travail du robot ainsi que pour minimiser les déplacements du robot et tirer avantage de l'aspect rotatif de chacune des pièces qui devront être ébavurées. En effet, au lieu de déplacer le robot autour de la pièce pour répéter une opération d'ébavurage sur une géométrie qui se retrouve à plusieurs endroits, ce sera l'axe rotatif qui sera mis à profit. De cette manière, l'impact de l'imprécision du robot est minimisé.

5.3.1 Répétabilité et précision

La répétabilité a été établie pour chacun des axes de manière indépendante.

Pour l'axe linéaire, la répétabilité de la position du zéro a d'abord été étudiée. Le zéro est établi lorsque le chariot vient frapper un interrupteur de fin de course et les moteurs s'arrêtent. La valeur lue aux encodeurs à ce moment est enregistrée et considérée comme étant le zéro de la course du chariot. Pour les besoins du test, le zéro de l'axe linéaire a été établi pour une première fois. Le chariot a ensuite été amené à 200 mm du zéro, puis ramené jusqu'à faire contact avec l'interrupteur de fin de course. Cet aller-retour est répété 50 fois et, pour chacune des deux positions, la valeur de l'encodeur est lue. La Figure 5.5 montre la variation de la position, selon les encodeurs, où le chariot se retrouve après avoir fait contact avec l'interrupteur de fin de course. On conclut donc que la répétabilité est de 0.5 mm étant donné que la position varie de -0.2 mm à 0.3 mm autour de la position moyenne.

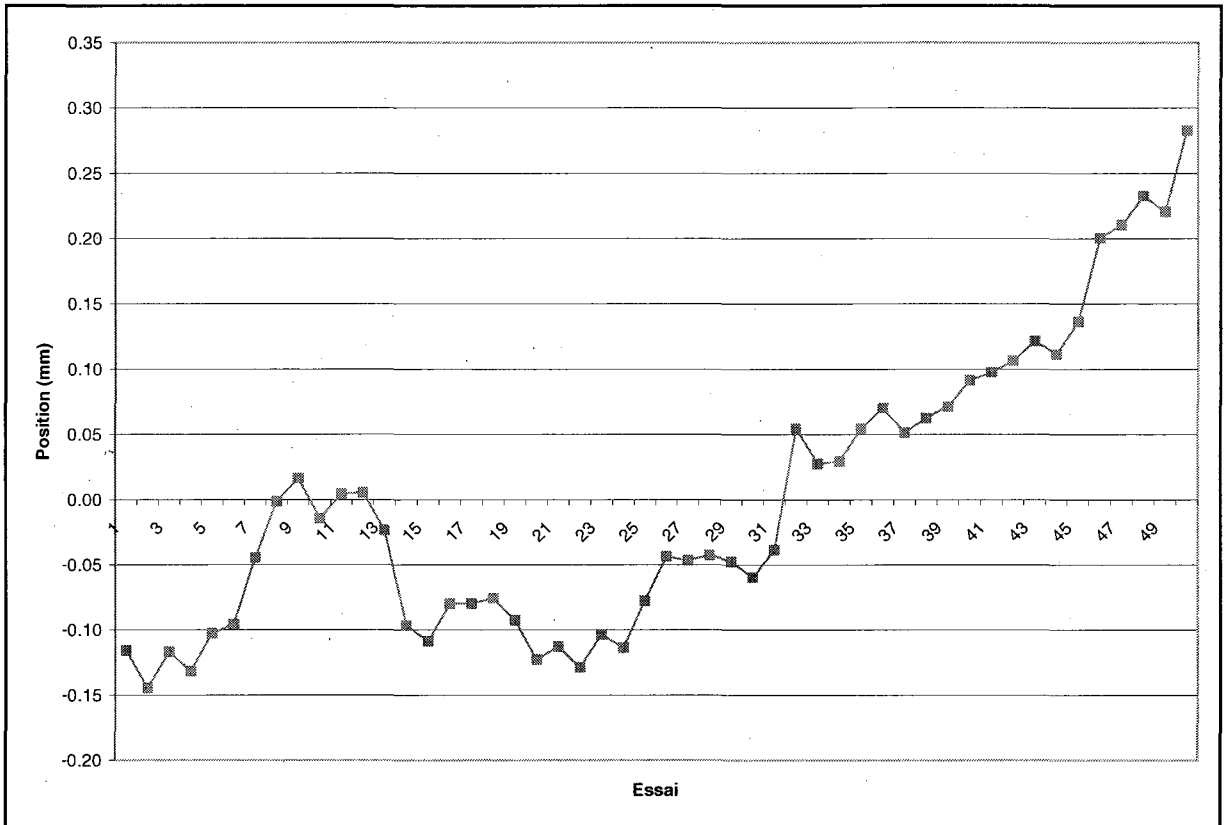


Figure 5.5 Variation de la position du zéro de l'axe linéaire autour de la valeur moyenne.

La répétabilité des déplacements a aussi pu être établie en évaluant la lecture de la position atteinte suite à la commande de déplacement de 200 mm (déplacement relatif) et la lecture de la position du zéro correspondant. La Figure 5.6 montre que le déplacement de 200 mm est constant à l'intérieur de $\pm 30 \mu\text{m}$.

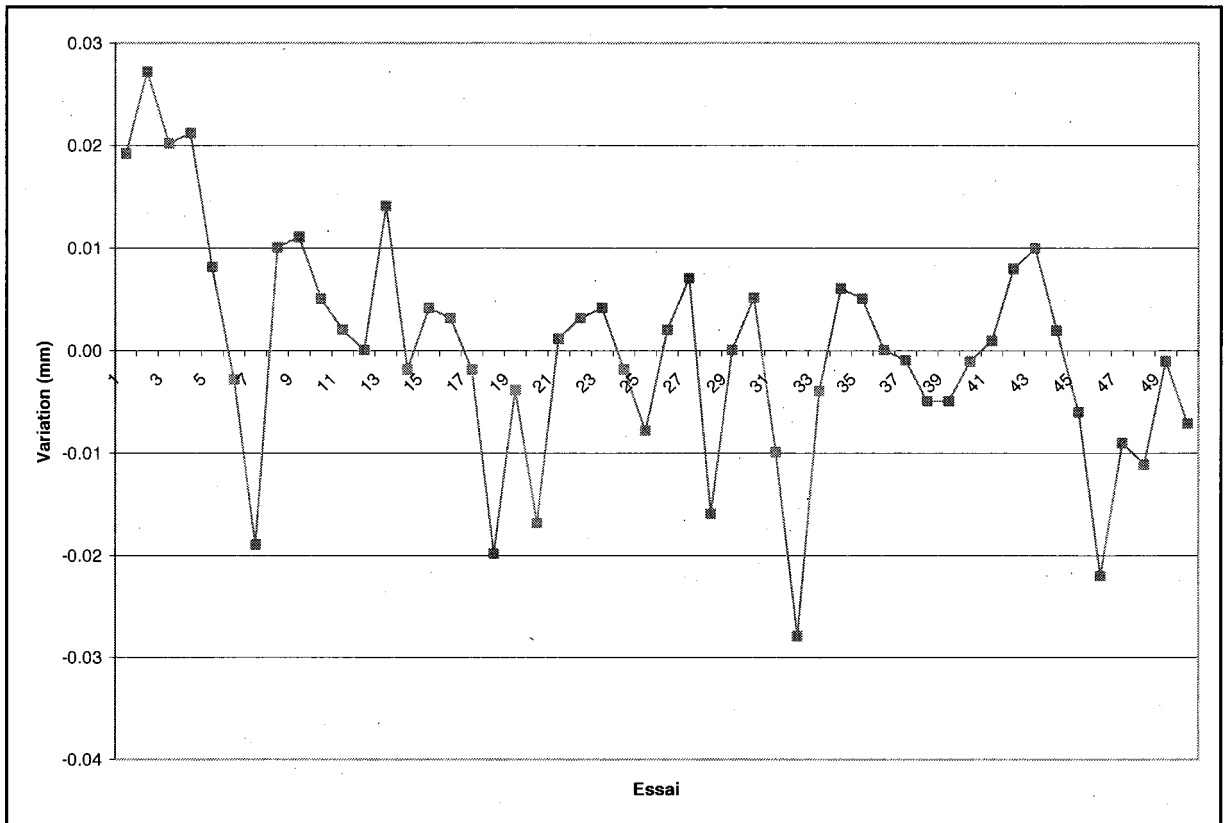


Figure 5.6 Variation du déplacement lors de la commande de déplacement relatif de 200 mm.

Finalement, la précision de l'axe linéaire a été évaluée en établissant le zéro de l'axe, en commandant un déplacement à cette même position zéro, puis à la position 150 mm. Contrairement à la situation précédente où le déplacement était d'une certaine distance (commande en déplacement relatif), on demande ici de déplacer l'axe linéaire à une position précise (commande en position absolue). Les déplacements aux positions zéro et 150 mm sont répétés 50 fois et les valeurs des encodeurs sont lues pour chacune de ces positions. (Voir Figure 5.7 et Figure 5.8). On constate un décalage de 0.3169 mm autour du zéro et de 0.2231 mm autour de 150 mm. Cette erreur de précision provient d'une combinaison de l'erreur mécanique et d'une erreur admise par la boucle de contrôle (PID) en régime permanent. Il est à noter que lors du premier essai, la position du zéro est juste puisque le chariot n'a pas été déplacé suite à l'établissement initial du zéro. On constate aussi que le premier déplacement à 150 mm n'est que légèrement erroné. On constate aussi que sur

l'ensemble des déplacements, la variation de la longueur du déplacement correspond à ce qui a été trouvé précédemment comme valeur de répétabilité.

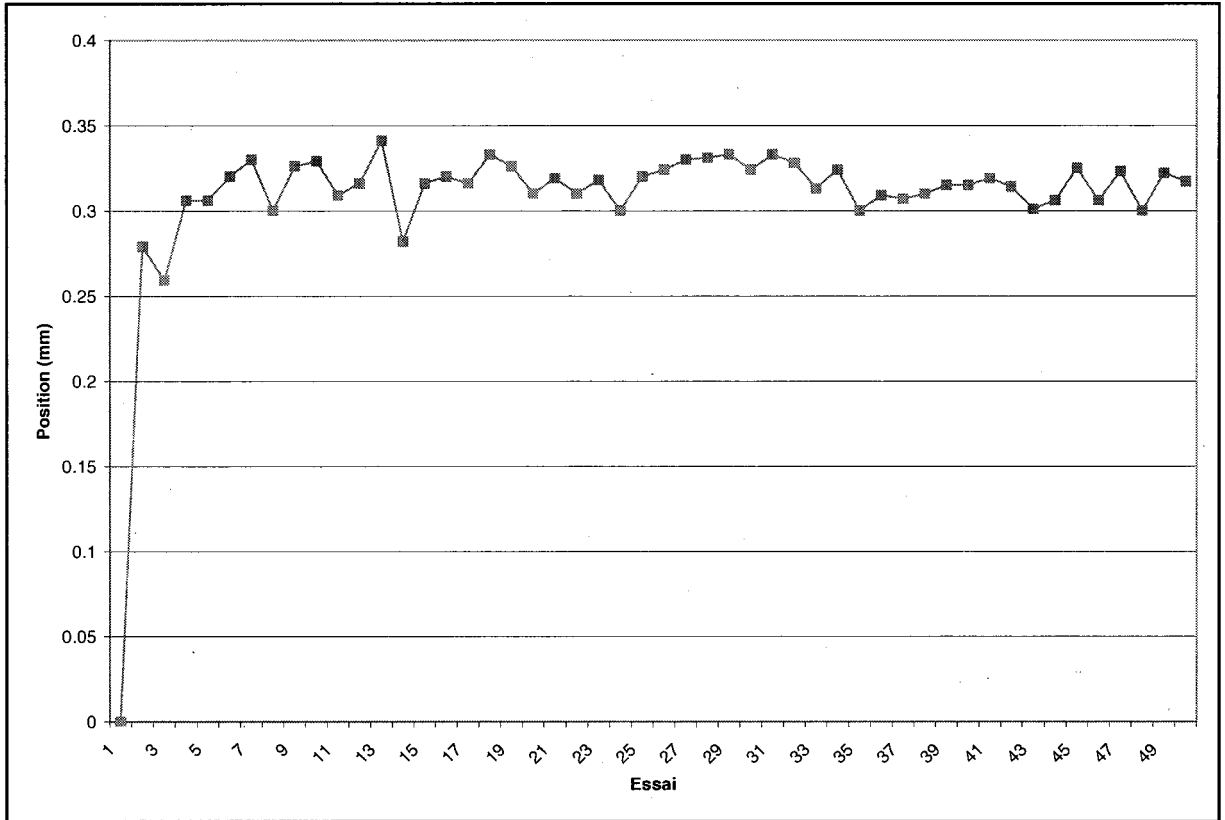


Figure 5.7 Positions du zéro de l'axe linéaire.

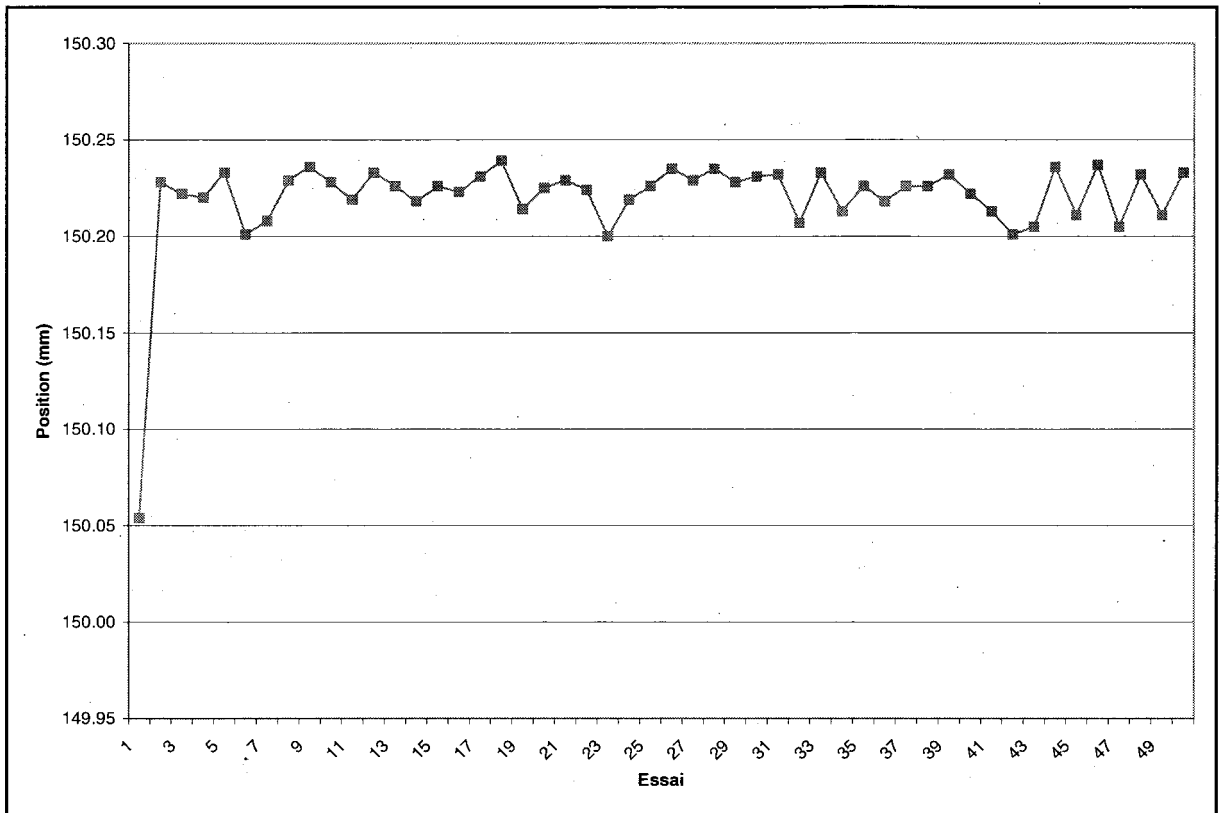


Figure 5.8 Positions après la commande de positionnement à 150 mm.

En ce qui concerne l'axe rotatif, sa répétabilité a été établie selon la même approche que celle utilisée pour l'axe linéaire, c'est-à-dire en retournant à la position zéro de l'encodeur et en commandant un déplacement de 25° . Une répétabilité de 0.0018° a été observée. Pour un cercle de 300 mm de diamètre cela correspond à un arc de cercle de $4.71 \mu\text{m}$.

La précision de l'axe rotatif a été évaluée en utilisant le robot et le palpeur, ce qui implique d'emblée une marge d'erreur attendue équivalente à la répétabilité du robot. Une pièce rotative avec une géométrie caractéristique (un trou) se répétant à tous les 9° a été fixée sur l'axe rotatif. Avec le robot et le palpeur, le centre d'un des trous a été trouvé, puis la pièce a été tournée de 9° (Voir Figure 5.9).

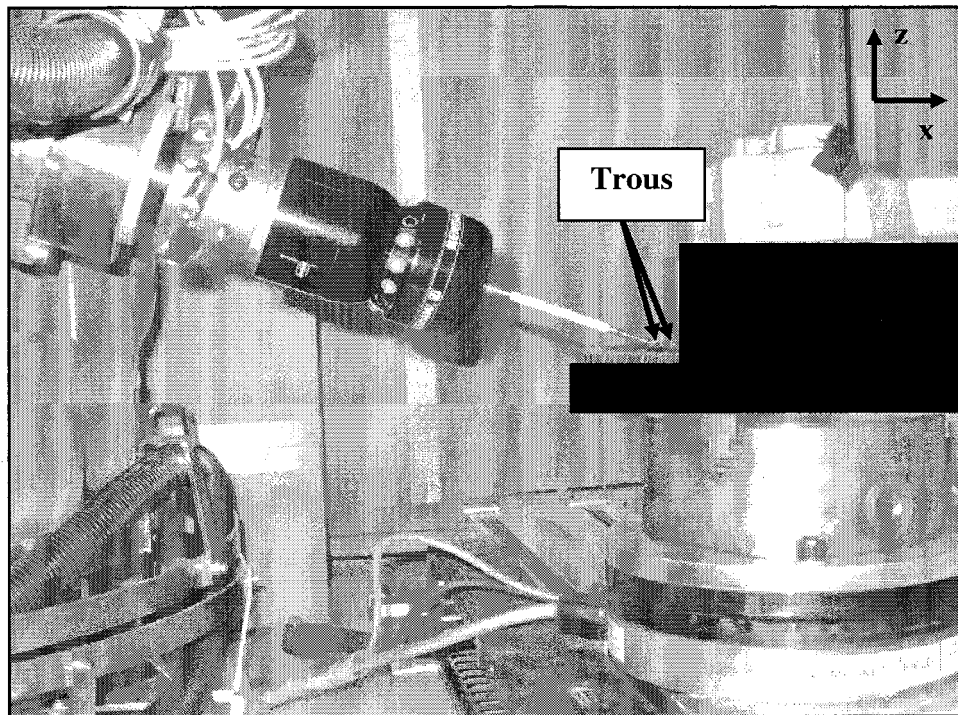


Figure 5.9 Lecture du centre des trous lors de l'étude de répétabilité de l'axe rotatif.

Cinq tours consécutifs ont été faits, donc 200 trous ont été évalués. La Figure 5.10 montre la variation des composantes X, Y et Z de la position du centre des trous. D'un trou à l'autre, lors d'un même tour, les positions varient à cause du décalage du centre de rotation de la pièce, des tolérances de fabrication, etc. Par contre, lorsque la position d'un même trou est comparée d'un tour à l'autre, elle varie de 0.030° , soit $79 \mu\text{m}$ pour un cercle de 300 mm de diamètre. Il faut noter que l'impact de l'erreur venant du robot doit être considéré dans une certaine limite puisque les mouvements étaient très restreints tout au long de la prise de mesure. Aussi, on note que les variations les plus importantes sont en Y et Z (Voir le référentiel sur la Figure 5.9), ce qui correspond aux axes les plus affectés par le mouvement de la pièce. La variation en X indique la variation de la profondeur du trou. Les valeurs obtenues dans cette direction n'ont que peu de significations étant donné que les touches se faisaient sur la paroi intérieure du trou uniquement, l'impact de l'erreur de positionnement n'est pas aussi direct sur cette composante.

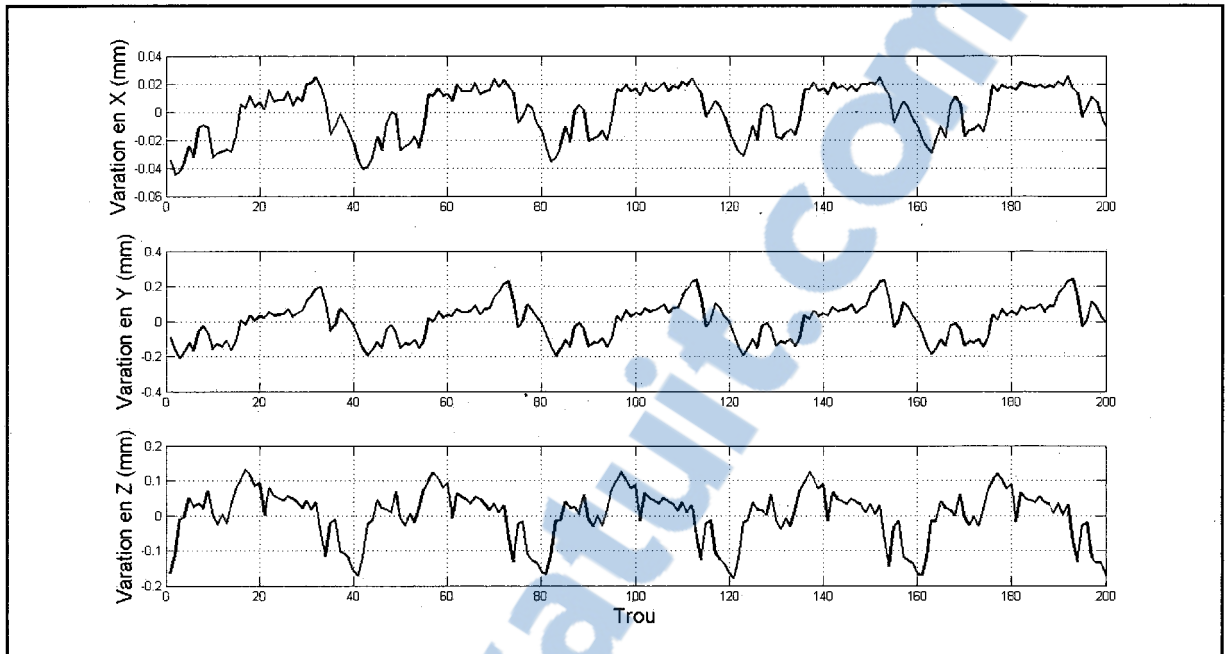


Figure 5.10 Variation de la position du centre des trous.

5.3.2 Technique de calibration

La technique de calibration pour l'axe linéaire consiste à faire une réinitialisation du zéro. Selon la répétabilité que nous avons évaluée, cette opération devrait se faire le moins souvent possible. Par contre, il faut savoir qu'à chaque coupure d'alimentation du contrôleur des axes auxiliaires, la réinitialisation du zéro doit être refaite sinon la position où se trouve le chariot est considérée comme le nouveau zéro.

Pour l'axe rotatif, aucune réinitialisation ne doit être faite étant donné qu'il sera utilisé exclusivement en déplacements relatifs.

5.3.3 Communication

Les axes auxiliaires sont connectés par communication sérielle avec le contrôleur (protocole RS-232). Bien que ces axes permettent deux DDL de plus, ils ne sont pas contrôlés

directement par le contrôleur du robot. La coordination des mouvements robot-axes auxiliaires n'est donc pas envisageable dans les circonstances.

5.4 Système de vision

Le système de vision utilisé est composé d'une caméra intelligente et d'un éclairage de type *ring light* blanc. Un assemblage rigide des deux composantes assure une configuration fixe de ce sous-système.

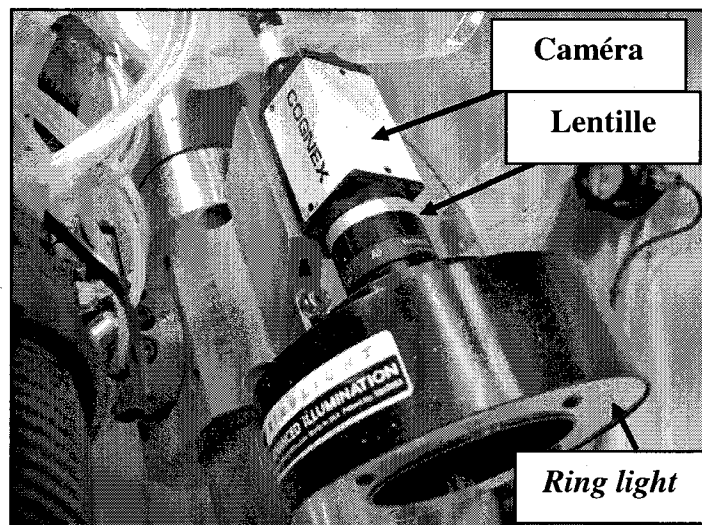


Figure 5.11 Assemblage caméra et éclairage.

Cet ensemble permet au robot de localiser la géométrie qui doit être ébavurée et établit la correction à apporter à la trajectoire définie hors-ligne. Le robot lit la correction et l'applique lors de l'exécution de la tâche. La caméra permet, entre autres, de pallier au manque de précision des axes auxiliaires.

Il est à noter que l'objectif de l'opération faite avec la caméra est de corriger l'erreur en position entre les trajectoires réalisées hors-ligne pour arriver à respecter les spécifications de fini de surface présentées au chapitre 6. D'ailleurs, l'assemblage de la caméra et des lumières est fixé sur celui de l'outil rotatif. De cette façon, le robot n'a pas à déposer l'outil rotatif,

prendre la caméra, effectuer la mesure, déposer la caméra et reprendre l'outil rotatif à chaque fois qu'une nouvelle géométrie doit être ébavurée. Les déplacements faits entre les opérations nécessitant la caméra et celles nécessitant l'outil rotatif sont limités au minimum. L'impact des non-linéarités du robot sur la qualité et la constance du résultat final est donc limité et corrigée par les compensations présentées à la section 6.2.1.

5.4.1 Répétabilité et précision

La caméra a une résolution de 1600 pixels par 1200 pixels, la distance de travail sera d'environ 300 mm et une zone d'environ 100 mm par 100 mm doit être inspectée. Pour répondre à ces critères une lentille de distance focale de 16 mm avec les caractéristiques compatibles au type de capteur de la caméra a été choisie. Les dimensions de la surface visible au point focal avec ce montage sont de 133 mm par 100 mm, ce qui correspond à ce qui était désiré.

Théoriquement, cet assemblage permet une précision de $83.125 \mu\text{m}/\text{pixel}$. Par contre, la précision de l'information qui sera obtenue lorsque l'éclairage, les non-linéarités de la lentille et du capteur de la caméra ainsi le traitement de l'image seront pris en compte peut grandement varier.

Dans cet optique, le système robot-caméra-éclairage a été soumis à quelques tests de répétabilité. Ces tests visaient à définir dans quelles conditions il serait possible d'obtenir la meilleure constance au niveau du positionnement d'une certaine géométrie sur une pièce et quelle est la variation dans la lecture.

Connaissant la répétabilité du robot, celle de la lecture faite par le système caméra-éclairage a été étudiée en positionnant la caméra vis-à-vis un trou sur une pièce. La coordonnée du centre du trou est l'information recherchée. Différents types d'éclairage ont été testés et les résultats obtenus sont très similaires d'une fois à l'autre. Les paramètres lors de la prise de l'image, comme le temps d'obturation, ont été ajustés à chaque fois pour obtenir une lecture

optimale. De plus, l'analyse de l'image s'est faite en utilisant les outils disponibles dans le logiciel fourni avec la caméra et ont été les mêmes pour toutes les conditions d'éclairage.

Pour 20 images prises avec la caméra immobile devant le trou, la répétabilité est d'environ 70.00 μm , soit un peu mieux que la précision de la caméra. Ceci peut s'expliquer par l'algorithme utilisé pour retrouver les coordonnées du centre du trou qui utilise plusieurs points sur le diamètre de ce trou pour calculer le point centre. L'erreur de précision sur la lecture de chacun des points tend donc à être atténuée.²

Maintenant, la répétabilité du système robot-caméra-éclairage doit aussi être évaluée puisqu'en situation de production la caméra devra prendre une image suite au déplacement du robot, ce qui n'était pas le cas dans le test précédent. Le robot effectuera donc un déplacement entre sa position au repos (*home*), une position intermédiaire et la position permettant d'avoir le trou à la distance focale et dans le centre de la caméra. L'opération est répétée 10 fois et une image est prise chaque fois.

La variation observée est très grande lorsqu'une seule image est prise pour calculer la position du trou, soit 210 μm . Par contre, lorsque plusieurs images sont prises et que la moyenne de ces lectures sont faites, il est possible d'améliorer le niveau de répétabilité, comme le montre la Figure 5.12. Les données lignes et colonnes font références aux pixels du capteur de la caméra et sont converties en mm selon la calibration de la caméra.

² Ces résultats ont aussi été corrélés avec ceux obtenus en utilisant les images prises par la caméra et un autre logiciel de traitement d'image.

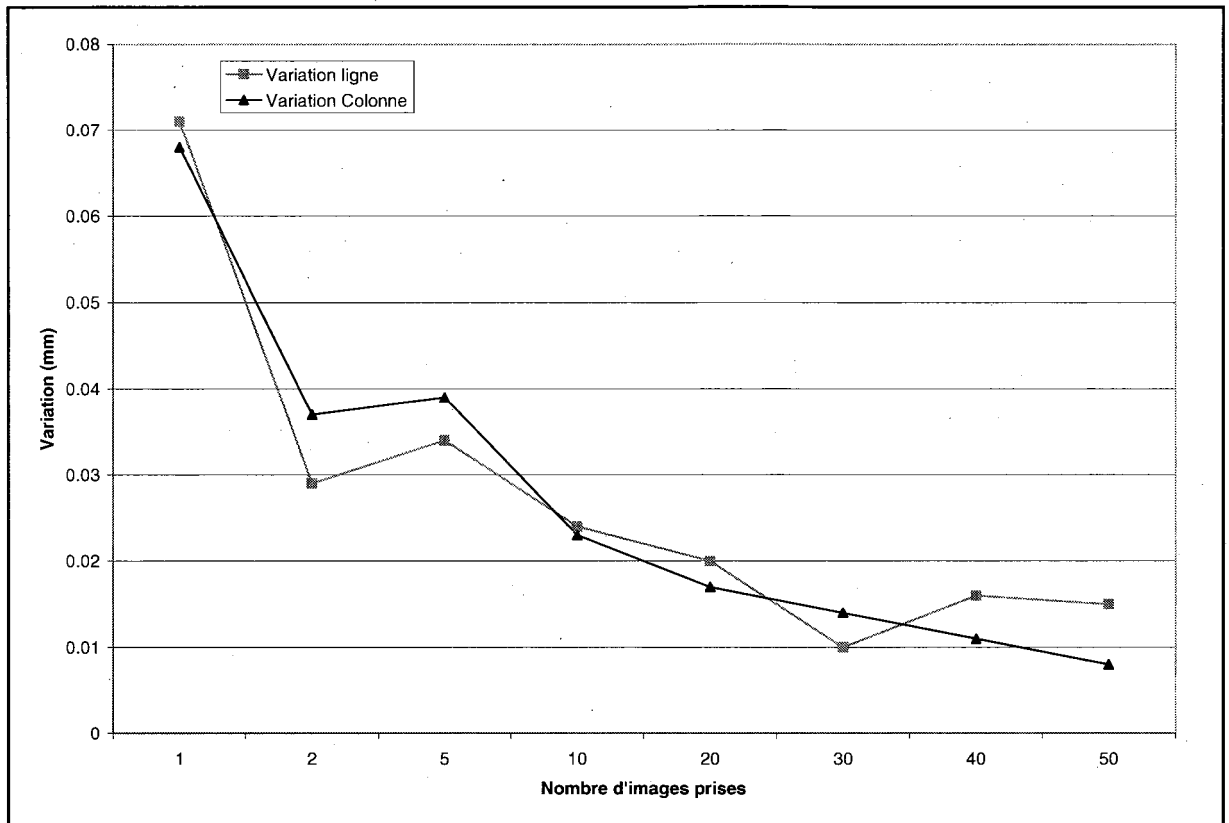


Figure 5.12 Répétabilité du système robot-caméra-éclairage selon le nombre d'images prises.

Un compromis entre temps d'exécution et répétabilité doit être fait, c'est donc 20 images de la géométrie qui seront prises pour une répétabilité de 20 μm . Ce choix est justifié par la précision désirée de 25 μm ou mieux et la réduction du temps de cycle³.

5.4.2 Technique de calibration

La calibration du système de vision est essentielle à son bon fonctionnement. Il faut d'abord calibrer les paramètres intrinsèques de la caméra : distorsion focale, correspondance pixel/mm, etc. Puis, il faut calibrer le TCP de la caméra de telle sorte que la position d'une

³ La durée de l'opération est abordée ici à titre discriminatoire et permet de justifier le choix fait. Bien que cet aspect soit primordial lors de la mise en production d'un système, il n'est pas prioritaire dans la phase de recherche et développement.

géométrie puisse être transmise au robot en terme de coordonnées cartésiennes par rapport au référentiel du robot.

Dans le premier cas, la caméra offre d'emblée quelques techniques de calibrations plus ou moins automatisées. Une de ces techniques met à profit une grille de calibration pointillée (*Voir ANNEXE I*) qui est utilisée pour calculer les paramètres internes de la caméra. Cette technique est éprouvée et a déjà été utilisée pour d'autres caméras semblables, elle sera donc celle choisie pour calibrer la caméra dans la cellule. Son désavantage principal est la quantité de manipulations qui doivent être faites manuellement, mais les fabricants de la caméra sont présentement en démarches pour automatiser ce genre de procédés.

Dans un second cas, il faut calibrer le TCP de la caméra. Pour ce faire, ABB propose une routine utilisant une sphère, comme celle utilisée au chapitre 3 lors de la calibration du TCP du palpeur. Par contre, cette procédure présente plusieurs obstacles avant d'être automatisée. De plus, elle ne permet pas de mettre l'emphase sur la relation entre le TCP de l'outil rotatif et celui de la caméra. Une autre approche développée spécialement pour le projet est facilement automatisable, utilise la même grille de calibration que celle utilisée pour calibrer les paramètres internes de la caméra et assure une bonne connaissance de la position relative du TCP de l'outil rotatif par rapport à celui de la caméra.

Cette technique demande d'avoir un TCP approximatif de la caméra et de positionner la grille de calibration de sorte qu'un axe principal de la grille de calibration et l'axe des X du référentiel du robot soient parallèles. De plus, il faut s'assurer que la grille soit positionnée sur une surface plane.

La première étape est de faire une touche sur le centre de la grille avec l'outil rotatif et un abrasif pointu conçu à cet effet (*Voir Figure 5.13*). De cette façon, on vient lire la position du centre de la grille selon le TCP de l'outil rotatif. Ensuite, on calibre les différents DDL de la caméra un à un.

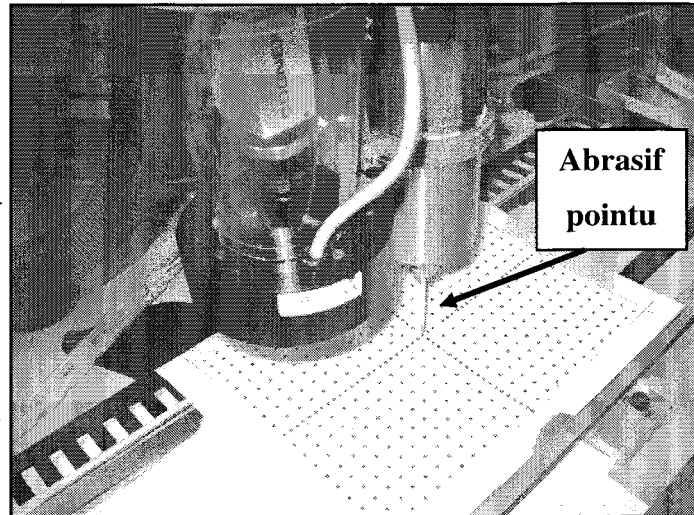


Figure 5.13 Identification du point central de la grille de calibration du TCP de la caméra.

D'abord, la mise au point (*focus* - axe Z) de la caméra est faite en positionnant le TCP approximatif de la caméra au centre de la grille, tel que lu par le TCP de l'outil rotatif. La caméra est reculée de 150 mm (distance choisit selon des tests préalables) selon son axe des Z. Elle est ensuite bougée par bond de 0.5 mm sur une distance de 250 mm (il faut préalablement prendre les précautions nécessaires pour éviter tout contact lors de cette manœuvre) et une image est prise pour chaque bond (Voir Figure 5.14 à Figure 5.16). Pour chacune des images la netteté de la transition entre les points noirs et le fond blanc est évaluée. La position pour laquelle l'image est la plus nette est considérée comme la position permettant d'obtenir la distance focale optimale. Le TCP est ensuite mis à jour connaissant le point centre de la grille, déjà identifié avec l'outil rotatif et l'abrasif pointu.

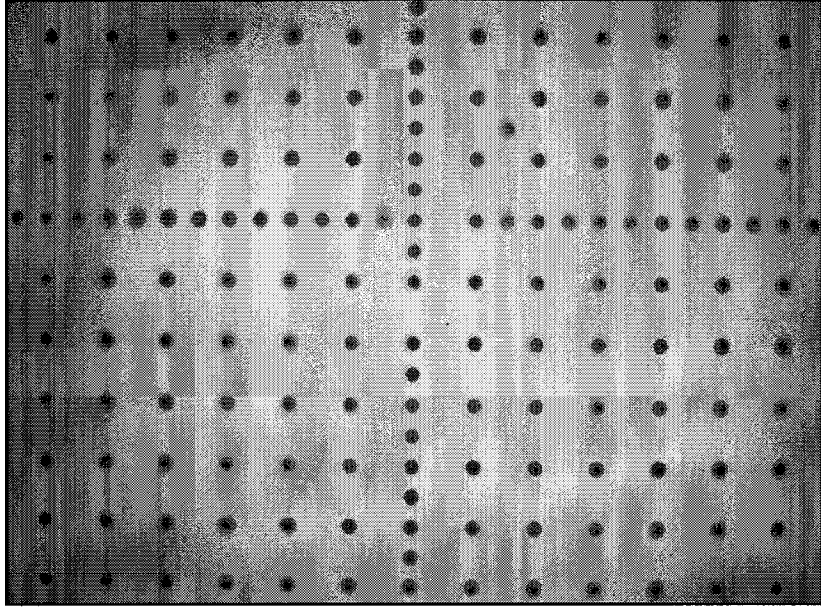


Figure 5.14 Calibration du focus de la caméra,
image prise avant l'atteinte du point focal.

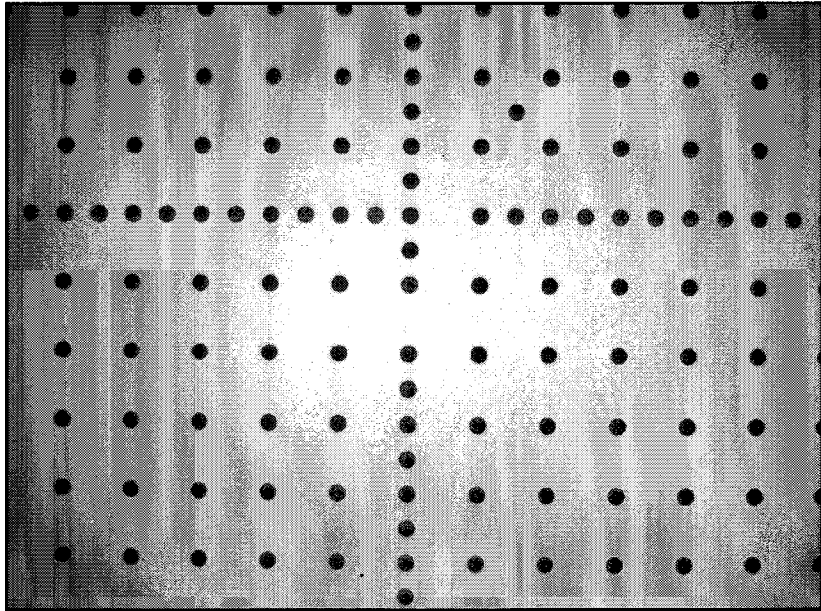
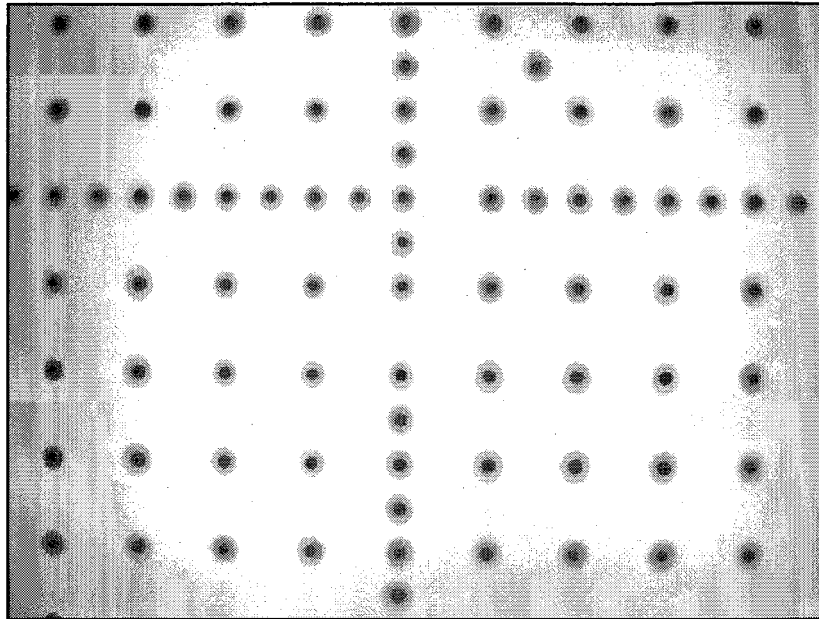


Figure 5.15 Calibration du focus de la caméra,
image prise très près du point focal.



**Figure 5.16 Calibration du focus de la caméra,
image prise après l'atteinte du point focal.**

Maintenant, la caméra doit être recentrée par rapport au centre de la grille. Le point centre de la grille est repéré selon une configuration caractéristique des points et la distance entre ce dernier et le centre de la lentille (*Voir* Figure 5.17 et Figure 5.18) est transmise au robot. Le déplacement correctif est fait et l'ajustement est repris jusqu'à ce que la distance soit à l'intérieur d'une tolérance de $50\ \mu\text{m}$. À ce moment, le TCP de la caméra est ajusté comme lors de l'étape précédente.

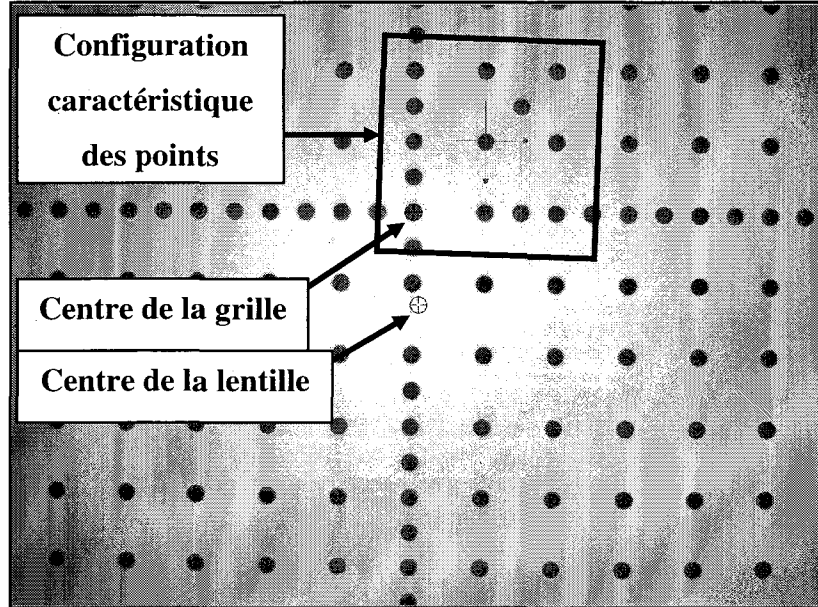


Figure 5.17 Calibration du centre de la caméra, image avant la calibration.

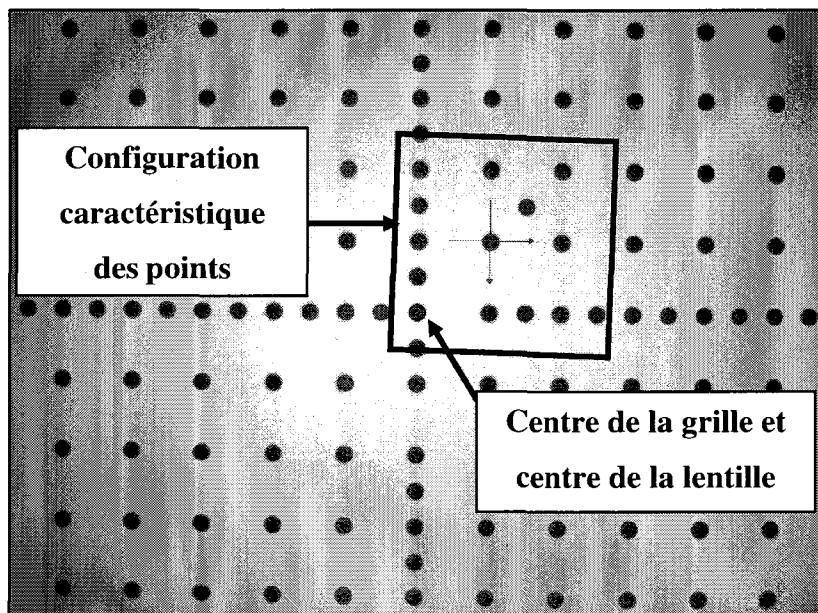


Figure 5.18 Calibration du centre de la caméra, image après la calibration.

La rotation autour de l'axe Z est corrigée en utilisant l'angle formé entre quatre droites et l'horizontale. Les droites sont formées à partir du point centre et d'un point sur chacun des

axes $X+$, $X-$, $Y+$ et $Y-$ (Voir Figure 5.19 et Figure 5.20). Une moyenne de l'erreur angulaire de chacune des quatre droites est faite et la correction à faire est envoyée au robot, le TCP de la caméra est finalement ajusté. Comme pour la procédure de recentrage, on reprend la correction jusqu'à ce que l'angle soit à l'intérieur des tolérances désirées, soit 0.02° .

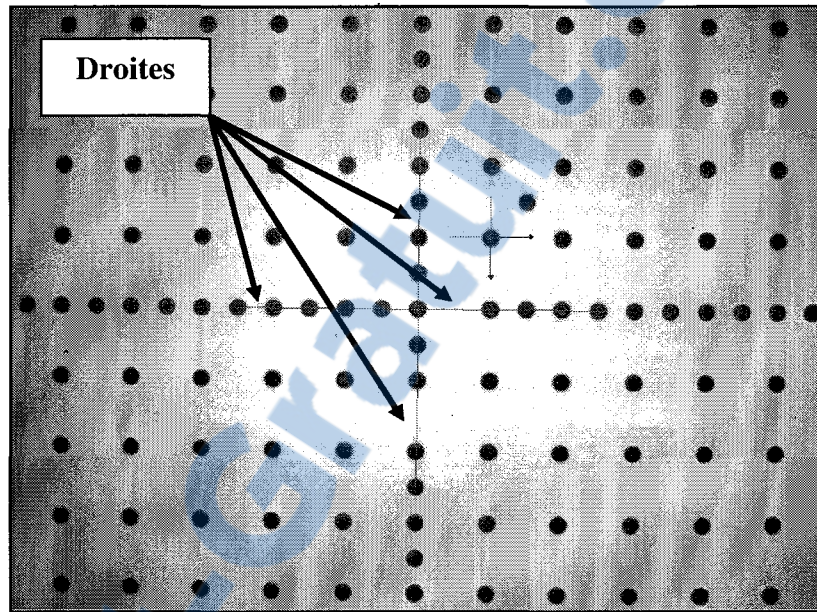


Figure 5.19 Calibration de l'angle autour de l'axe Z de la caméra, image avant la calibration.

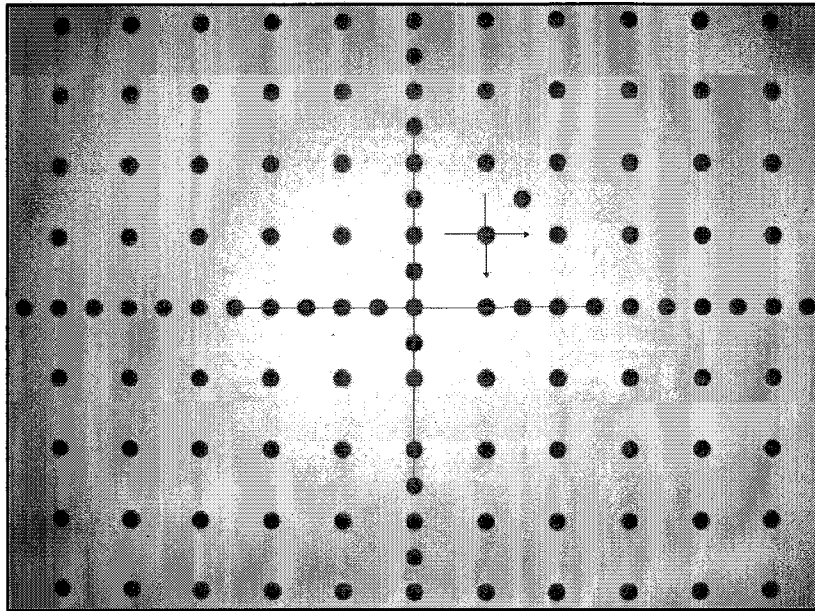


Figure 5.20 Calibration de l'angle autour de l'axe Z de la caméra, image après la calibration.

Finalement, la correction des angles autour des axes X et Y est faite. Bien que ces corrections soient faites consécutivement, elles se basent sur le même *modus operandi*. Par exemple, pour la calibration de l'angle autour de l'axe X, le TCP actuel de la caméra est posé sur le point centre de la grille avec un angle de 5° autour de l'axe X. L'angle désiré du TCP est modifié par bond de 0.5° autour de ce même axe jusqu'à ce qu'il atteigne -5° . À chaque bond, la distance entre un point en X+ et le centre est enregistrée, ainsi que la distance entre le point en X- correspondant et le centre (*Voir Figure 5.21*). Selon l'angle, les distances perçues varient et l'angle recherché est celui où les deux distances sont égales. Comme la prise de donnée n'est pas exhaustive, une régression linéaire est effectuée sur les deux suites de données et le point de croisement des deux droites obtenues nous permet d'interpoler la correction angulaire à donner pour que la caméra soit parfaitement perpendiculaire selon l'axe des X. Comme pour les procédures précédentes, l'opération est répétée jusqu'à ce que la correction à apporter soit à l'intérieur de 0.05° , puis le TCP de la caméra est mis à jour. La même procédure est répétée pour l'angle autour de l'axe Y de la caméra (*Voir Figure 5.22*).

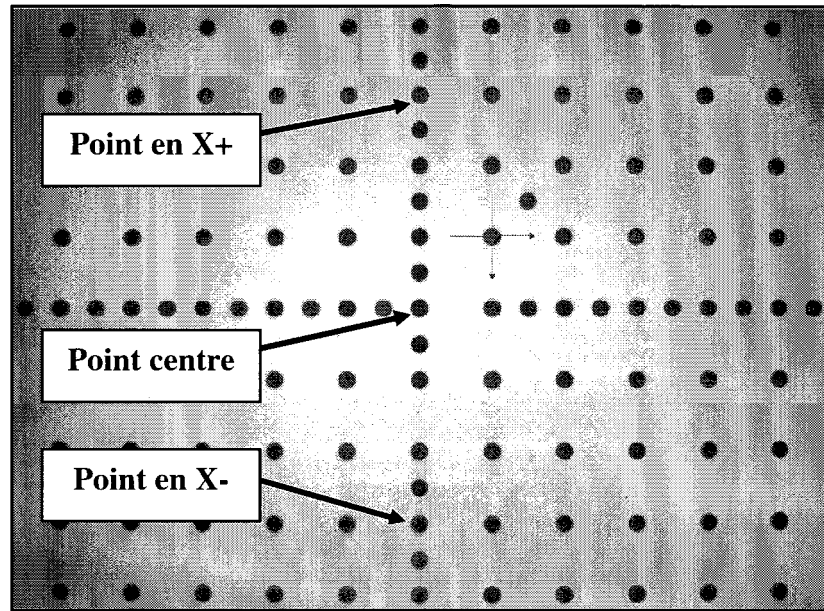


Figure 5.21 Calibration de l'angle autour de l'axe X de la caméra, image durant la calibration.

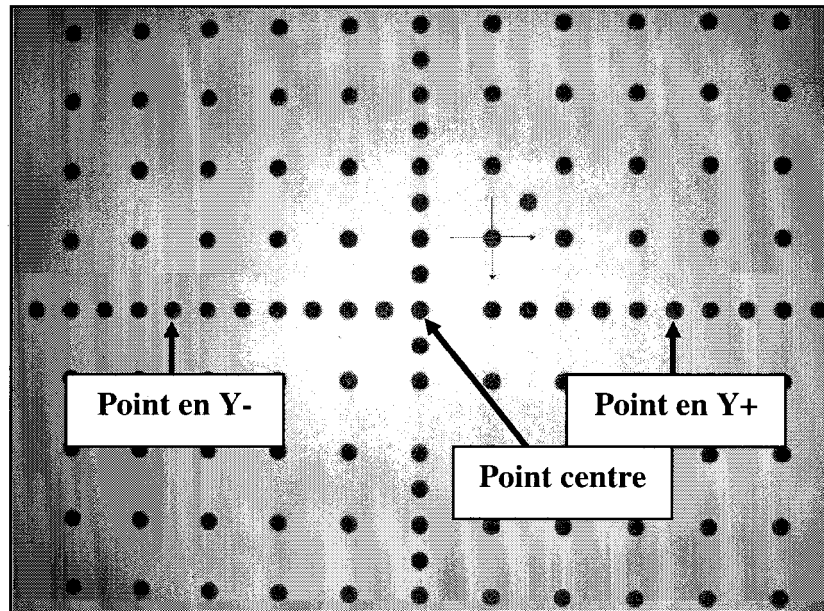


Figure 5.22 Calibration de l'angle autour de l'axe Y de la caméra, image durant la calibration.

Suite à ces calibrations, on répète une dernière fois les opérations d'alignement de l'angle autour de Z et de recentrage pour ajuster le plus finement possible le TCP de la caméra.

5.4.3 Communication

Le contrôleur de la caméra admet la communication *ethernet*, ce qui facilite les échanges de données avec le contrôleur du robot.

5.5 Outil rotatif

L'outil rotatif utilisé (Voir Figure 5.23) est commandé par un contrôleur fait sur mesure. Cet ensemble permet d'obtenir une vitesse variable de 5000 RPM à 22000 RPM, de contrôler le changement d'abrasif via une valve pneumatique et offre un faible battement de 3 μm . L'outil rotatif est fixé à l'ensemble caméra-éclairage et forme un tout qui se connecte au robot par un changeur d'outil femelle.

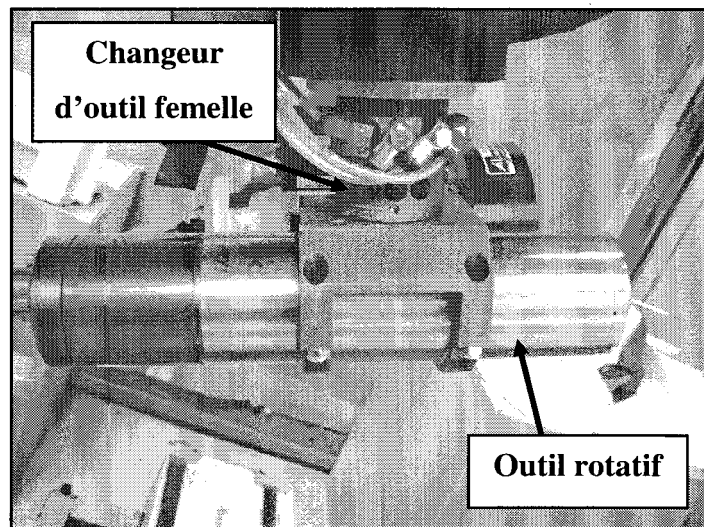


Figure 5.23 Outil rotatif.

Les abrasifs seront pris par l'outil rotatif et ce dernier permettra de contrôler le sens et la vitesse de rotation lors de l'opération d'ébavurage.

5.5.1 Répétabilité et précision

La précision de positionnement de l'outil rotatif dépend simplement de la précision du robot et du jeu dans son assemblage. Cette précision n'a pas pu être vérifiée par un appareil plus précis que le robot. Cependant, des résultats de tests d'intégration ont confirmé la capacité de cet ensemble à effectuer l'opération d'ébavurage selon les spécifications et seront présentés dans les prochains chapitres.

5.5.2 Technique de calibration

Le TCP de l'outil rotatif est trouvé par la technique de calibration proposée par ABB.

Le contrôle de la vitesse de l'outil rotatif a été calibré en utilisant un capteur de fréquence fait pour indiquer la vitesse de rotation d'objets. Le signal de contrôle venant du module AI/O a été ajusté pour maximiser la plage de tension disponible.

5.5.3 Communication

Les DI/O sont utilisés pour contrôler le sens de rotation, l'activation ou la désactivation, un signal d'erreur et la prise et le dépôt des abrasifs. Un signal 0-10V du module AI/O permet de contrôler la vitesse de rotation.

5.6 Abrasifs

Les abrasifs qui seront utilisés en production ne sont toujours pas définis. Des études sont présentement en cours pour établir quels seront les meilleurs abrasifs en fonction du travail à effectuer. Dans le cadre du présent travail, les abrasifs utilisés sont ceux qui sont déjà inscrits dans les directives d'ébavurage manuel des pièces (*Voir* Figure 5.24 et Figure 5.25) chez P&WC.

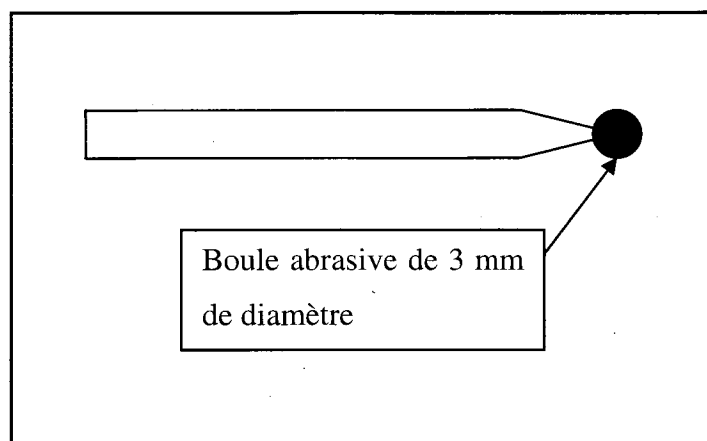


Figure 5.24 Abrasif 1.

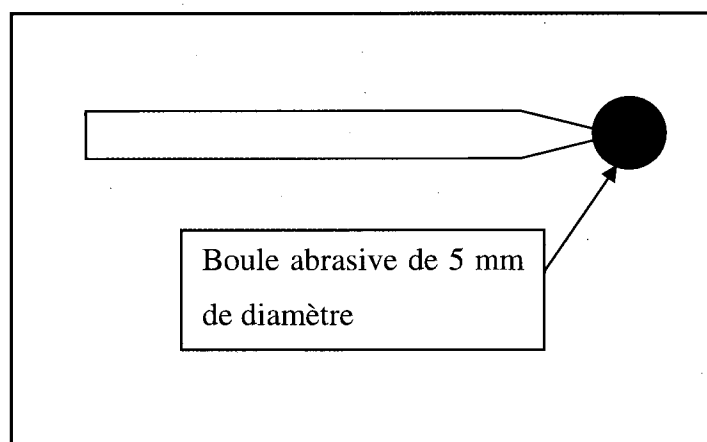


Figure 5.25 Abrasif 2.

Les abrasifs sont montés dans un adaptateur permettant la préhension facile de ceux-ci par l'outil rotatif (*Voir* Figure 5.26). Un équipement particulier devra être développé pour assurer une constance au niveau du positionnement des abrasifs ou un capteur devra être mis à profit pour valider les dimensions du TCP de l'abrasif et modifier le TCP outil en fonction de l'information trouvée.

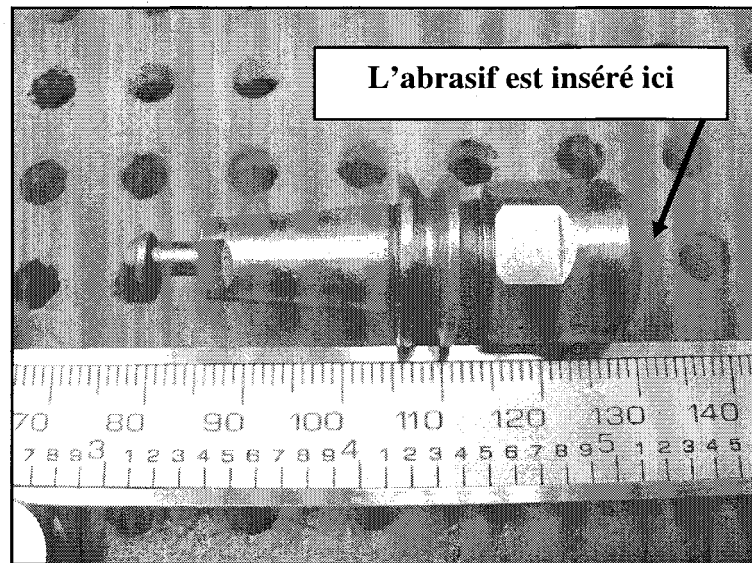


Figure 5.26 Adapteur pour abrasif.

Plusieurs paramètres liés aux abrasifs doivent encore être étudiés : l'usure, l'agressivité, le coût, les dimensions, les formes, etc. D'autres projets de recherche en cours actuellement chez P&WC se penchent d'ailleurs sur ces questions.

5.7 Conclusion

Ceci conclut la présentation et la caractérisation des différents dispositifs utilisés dans la cellule d'ébavurage robotisé. L'intégration de la cellule et l'aspect du contrôle de force seront développés dans le prochain chapitre, tout comme les résultats et l'analyse de l'étude de cas.

CHAPITRE 6

ÉTUDE DE CAS : ÉBAVURAGE ROBOTISÉ

Ce chapitre présentera d'abord une vue d'ensemble de la cellule d'ébavurage robotisé et l'architecture logicielle permettant l'intégration des différents intervenants. Les conditions générales dans lesquels les tests ont été effectués sont ensuite présentées. Finalement, la compensation de l'imprécision par la caméra et le capteur de force sera explorée. Les résultats obtenus seront optimisés jusqu'à trouver les paramètres optimaux pour arriver à rencontrer les exigences d'ébavurage pour la géométrie choisie.

6.1 Intégration

Les éléments présentés au chapitre 5 ont été intégrés dans une cellule de développement (Voir Figure 6.1). Tous les périphériques sont contrôlés par le robot, selon les différents modes de communication, aussi présentés dans le chapitre précédent.

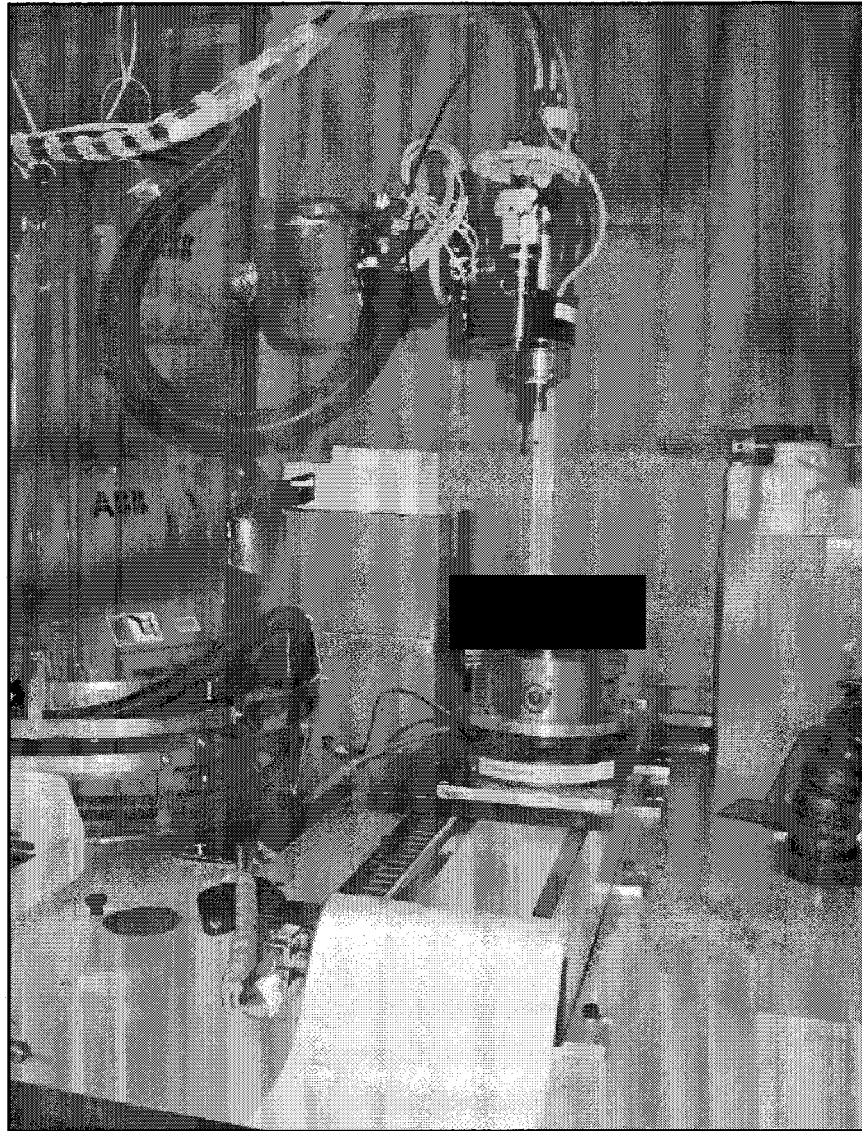


Figure 6.1 Cellule de développement pour procédés robotisés.

En terme de précision, le Tableau 6.1 dresse le portrait de la répétabilité des différents sous-systèmes.

Tableau 6.1 Sommaire des caractéristiques de répétabilité et précision des éléments de la cellule

Sous-systèmes	Technique de calibration	Précision ($\mu\text{m}/^\circ$)	Répétabilité ($\mu\text{m}/^\circ$)
Robot (en position)	ABB	N/D	± 30
Robot-Palpeur	ABB	N/D	± 30
Robot-Caméra-Éclairage	Développée sur mesure	N/D	20
Axe linéaire	Réinitialisation du zéro	500	± 30
Axe rotatif ($^\circ$)	Ne s'applique pas	0.03	0.0018

Les caractéristiques de l'ensemble robot-palpeur n'ont pas été explicitement définies dans le chapitre 5. Comme la précision du palpeur est très grande par rapport à celle du robot, on néglige son imprécision. Aussi, le sous-système robot-caméra-éclairage considéré est celui avec les paramètres optimaux lors de la prise d'image et l'utilisation de 20 images pour en faire la moyenne. Finalement, toutes les précisions impliquant le robot sont non déterminées (N/D) étant donné qu'aucun instrument de mesure avec une précision suffisante n'a pu être utilisé.

6.1.1 Architecture logicielle

L'architecture permet de mettre en relation tous les dispositifs dont nous avons parlé jusqu'à présent.

L'essentiel de la programmation tient dans le contrôleur du robot. Quelques routines, propres à l'axe linéaire et à la caméra, résident tout de même dans leur contrôleur respectif. En ce qui concerne l'axe linéaire, le langage utilisé est celui propre au contrôleur. C'est essentiellement la routine de calibration qui réside dans son contrôleur. D'autres routines y sont toujours, mais des changements ont été faits suivant la mentalité d'avoir le robot comme seul maître de la cellule. La suite de commandes nécessaires pour exécuter un déplacement relatif, un

déplacement en position absolue, une lecture des encodeurs, etc. est maintenant envoyée par le contrôleur du robot au contrôleur des axes auxiliaires, au lieu d'être appelée par une routine résidant dans le contrôleur des axes auxiliaires.

Au niveau de la caméra, la routine permettant la calibration, celle pour la transformation mm/pixels et celles permettant la localisation des différentes géométries sont dans la mémoire interne de la caméra. Elles sont programmées via l'interface de type « feuille de calcul » offerte par le logiciel accompagnant la caméra. Ces routines permettent donc d'effectuer la localisation de certaines formes ou de certaines caractéristiques, d'effectuer des calculs et des transformations de systèmes de coordonnées. Ce faisant, le contrôleur du robot est donc libéré de la charge de ces calculs, il ne fera qu'utiliser les résultats de ceux-ci.

Le robot a été programmé avec la plate-forme de développement RobotStudio®, fournie par ABB. Le langage de programmation propriétaire d'ABB est le RAPID. Il permet la création de différentes tâches, qui peuvent être exécutées en parallèles, de modules qui permettent surtout d'organiser la programmation et de procédures et fonctions⁴. Un schéma très général de la programmation du robot est présenté à la Figure 6.2.

⁴ Pour une référence RAPID à jour il faut contacter un vendeur ABB.

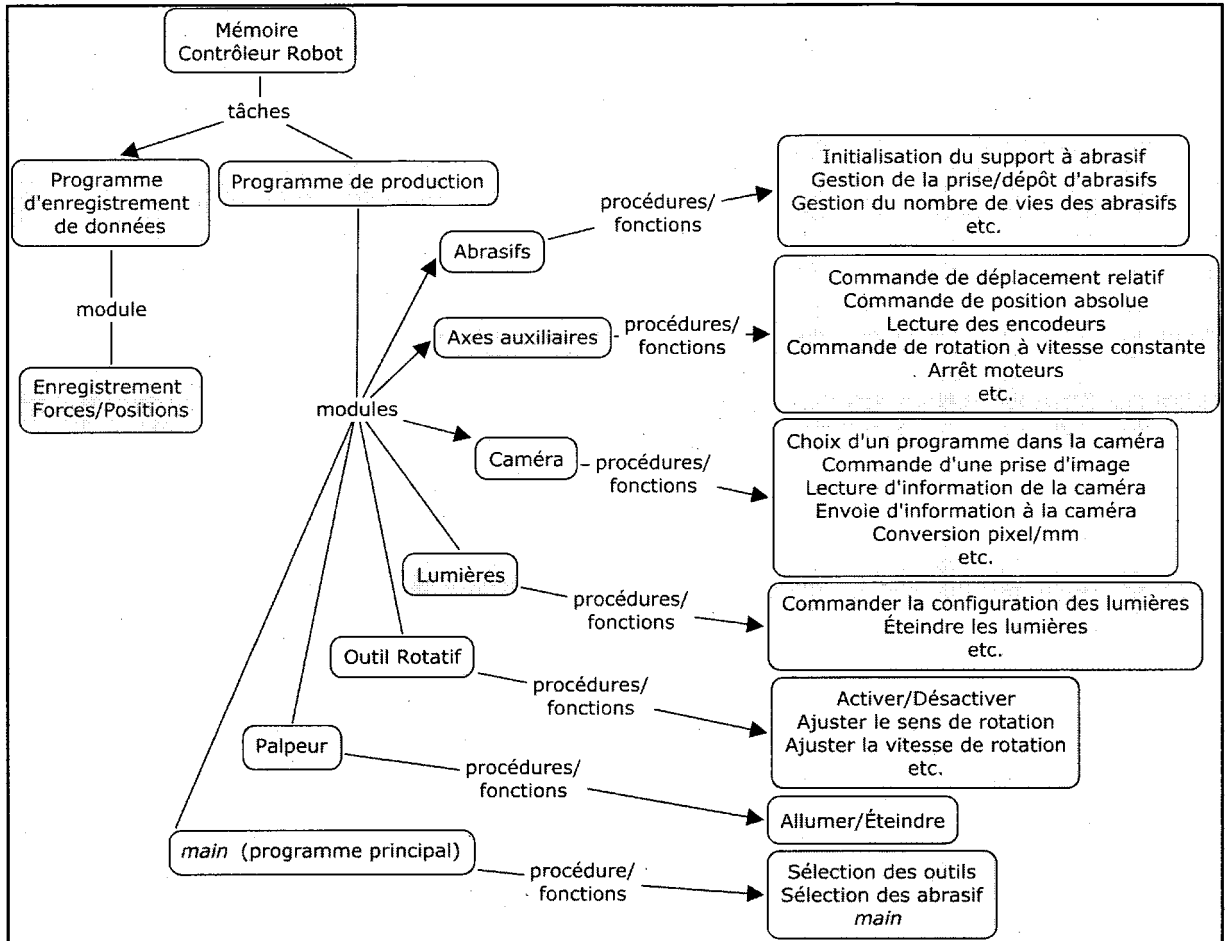


Figure 6.2 Schéma de la programmation du robot.

Dans une optique de production, une architecture logicielle plus large a aussi été développée. Elle admet l'interaction avec les opérateurs, les planificateurs et les développeurs, telle que promu par P&WC lors du développement de procédé manufacturier pour une pièce. Elle fait intervenir une panoplie de fichier permettant au planificateur d'utiliser la programmation hors-ligne avec RobotStudio®, à l'opérateur de commander l'exécution d'une pièce quelconque et aux développeurs d'ajouter des fonctions au système selon l'évolution des besoins.

6.2 Conditions générales de tests

Avec l'intégration faite des différents dispositifs et l'architecture logicielle qui permet de supporter plusieurs scénarios de tests, il est possible d'étudier les capacités du capteur de forces/moments et d'évaluer les résultats qui seront obtenus en production avec ce système.

6.2.1 Choix de la géométrie

Le mandat de développement pour ce genre de système est d'abord venu du département responsable de la production des disques de turbine chez P&WC. Près de 100 variétés de disques de turbines sont produites. Par contre, seulement une dizaine de types de géométrie sont recensés. Il est donc avantageux de développer un procédé paramétrique en fonction des géométries et non des pièces. La suite du présent travail se concentre donc sur la démonstration de la faisabilité de l'ébavurage robotisé avec contrôle de force sur une géométrie particulièrement répandue, soit les petits trous situés sur les *balancing rims* (Voir Figure 6.3). Bien que ces trous soient de mêmes dimensions d'une pièce à l'autre, une pièce en particulier a été choisie pour faire les tests. Le choix s'est fait en fonction de l'accessibilité de la géométrie et de la disponibilité de la pièce.

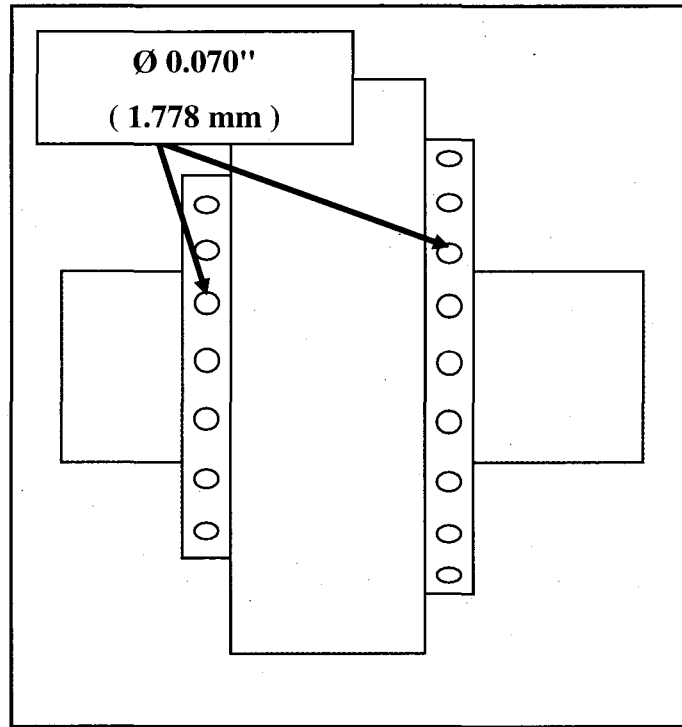


Figure 6.3 Petits trous sur les *balancing rims*.

Les exigences en terme de tolérances dimensionnelles pour cette géométrie demande de chanfreiner le coin d'une dimension horizontale et verticale de 76.2 μm à 381 μm . La variation entre ces limites est admise mais l'inspection est visuelle et comparée à l'historique de ce qui a été produit, ce qui n'est quantifié nulle part et variable d'un inspecteur à l'autre.

Les requis dimensionnels sont atteignables selon la répétabilité du robot, mais il faudra que ce dernier soit amené précisément dans l'axe du trou pour que l'opération puisse être réalisée avec succès.

6.2.2 Environnement

Les conditions de production tenteront d'être imitées le plus possible lors des tests. La pièce sera fixée dans l'axe rotatif et amenée à une position où le robot travaille le mieux. La cellule de test est située dans un environnement de production avec les conditions de bruit, vibration,

poussière, etc. auxquelles elle sera soumise lors de son déploiement (Voir Figure 6.4). Par contre, lorsque la caméra est utilisée, la cellule est isolée pour mieux contrôler l'éclairage (Voir Figure 6.5).

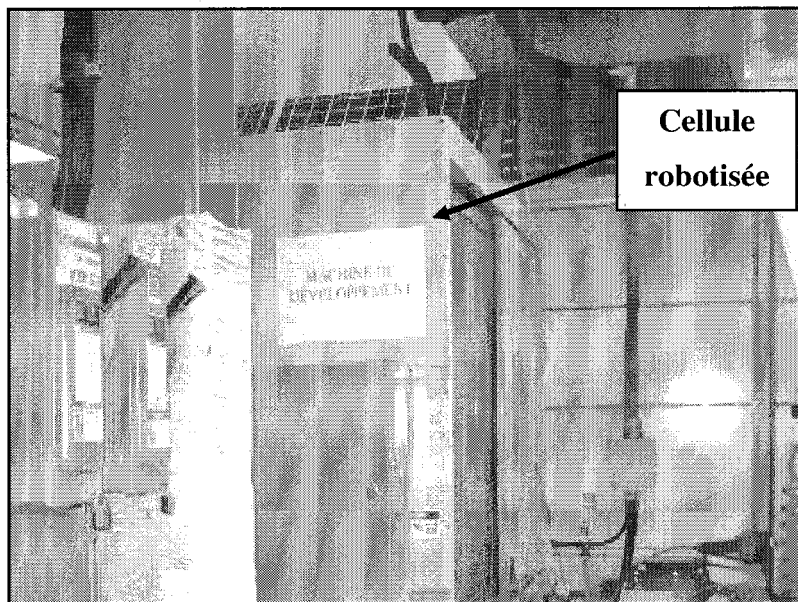


Figure 6.4 Cellule dans son environnement.

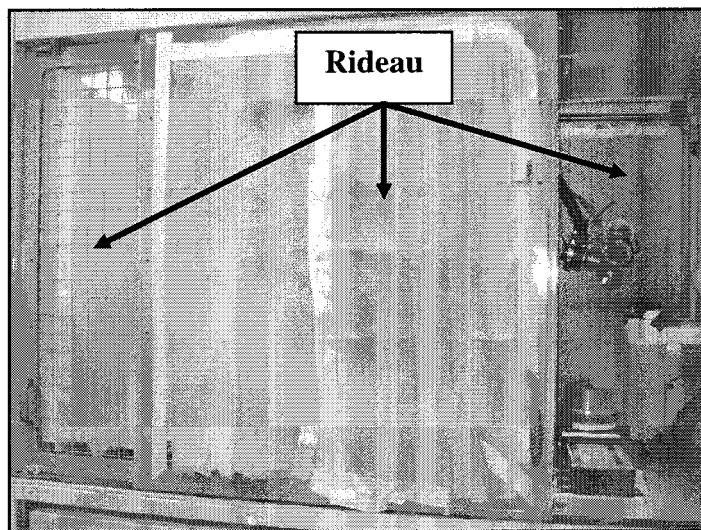


Figure 6.5 Rideaux pour le contrôle de l'éclairage.

6.2.3 Trajectoire

Pour effectuer l'ébavurage de la géométrie choisie, deux types de trajectoires ont été utilisées : l'approche horizontale (Voir Figure 6.6) et l'approche verticale (Voir Figure 6.7). Dans les deux cas, l'abrasif sera donc enfoncé dans le trou, en suivant l'axe du trou, jusqu'à l'atteinte de la force commandée (variant selon le test effectué), puis reviendra à sa position initiale. La première approche simplifie la lecture des forces à l'outil et la seconde est directement inspirée de ce qui est présentement fait manuellement.

Les lectures de la force et de la position seront faites par la tâche développée à cet effet (Voir Figure 6.2) dans le contrôleur du robot et permettront l'analyse des performances de chaque test. Il est à noter que les lectures de la force obtenues sont directement celles du capteur projeté au TCP de l'abrasif (Voir $F_{measured}$ à la Figure 6.8).

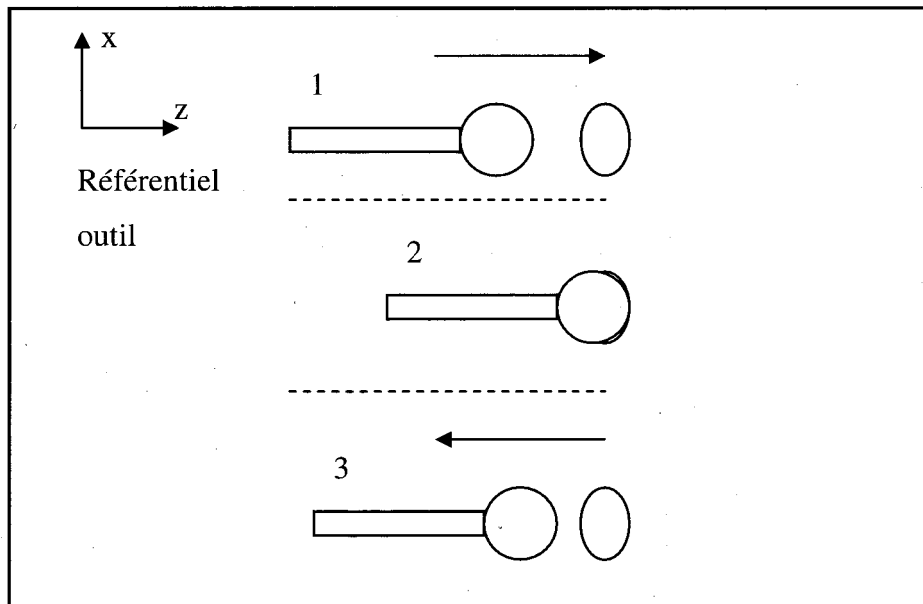


Figure 6.6 Illustration des étapes pour l'ébavurage de petits trous sur les *balancing rims*, approche horizontale.

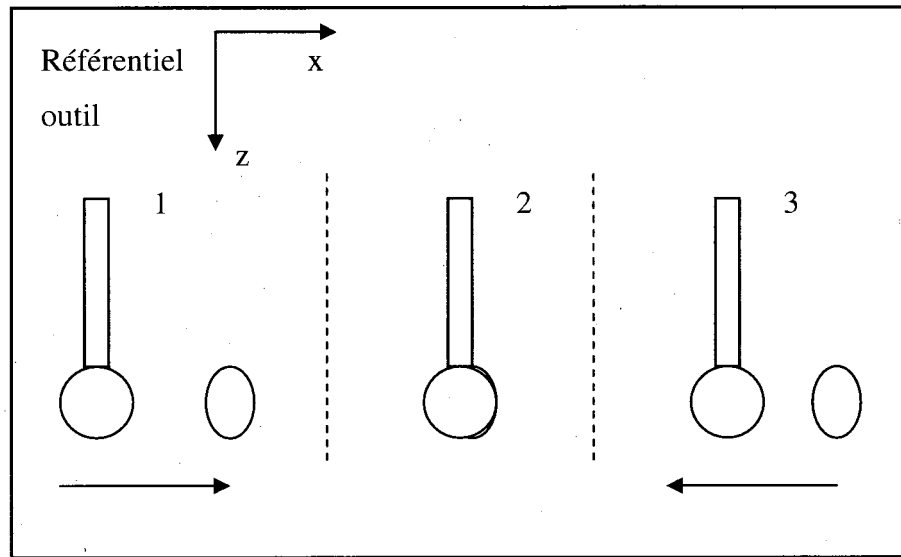


Figure 6.7 Illustration des étapes pour l'ébavurage de petits trous sur les *balancing rims*, approche verticale.

Une facette importante de l'opération d'ébavurage est l'alignement avec l'axe du trou. Pour la première série de tests, présentée dans la section suivante, la position du centre du trou sera apprise au robot. Ceci permet de mettre l'accent sur les paramètres du contrôle de force. Une fois ces paramètres définis, les derniers tests valideront le procédé complet en passant par le positionnement de la géométrie par la caméra. Comme une erreur de précision persiste entre le TCP de la caméra et celui de l'outil rotatif, donc de l'abrasif, il a fallu recourir à l'utilisation de compensations.

D'abord, comme la répétabilité du système est très bonne, son erreur en position est constante. Une compensation en X, Y et Z est trouvée de façon semi-automatisée. Elle est calculée suite à une séquence de commandes prévues dans l'architecture logicielle, permettant de déplacer le robot en le bougeant par le TCP de l'outil, grâce au capteur de force. Une fois le robot positionné à la position réelle désirée l'erreur de positionnement est calculée et enregistrée en tant que compensation, valide pour un type de géométrie, sur un modèle de pièce particulier. Pour la même géométrie sur un autre modèle de pièce, des vérifications devront être faites avant d'utiliser les mêmes compensations. C'est-à-dire que l'imprécision sera la même tant que la configuration pour le positionnement de la caméra et

celle pour le positionnement de l'outil rotatif sera semblable. Le volume pour lequel la compensation reste bonne n'a pas été vérifié lors de ce travail. Selon les manipulations faites, l'amplitude de la compensation peut aller jusqu'à 2 mm par axe (X, Y, Z), résultat de l'erreur en position et en orientation des composants de la cellule.

6.2.4 Variables

Les premiers tests permettent de définir la valeur à utiliser lors de l'ébavurage robotisé des petits trous avec contrôle de force pour un ensemble de paramètres. Ces différents paramètres sont :

- a) le type d'abrasif (**Abrasif**)
- b) la force à appliquer (**F**)
- c) la vitesse de déplacement (**V**)
- d) le taux d'amortissement (*damping ratio* **D**) (Voir Figure 6.8)
- e) la fréquence de coupure du filtre passe-bas (*Low Pass Filter* **LP**) (Voir Figure 6.8)
- f) le taux de changement de la force lors de l'approche (*Force Change Start* **FCS**)
- g) le taux de changement de la force lors du retrait (*Force Change End* **FCE**)
- h) la force limite avant de considérer qu'il n'y a plus de contact (*Zero Contact Force* **ZCF**)

Les tests seront définis par une plage de valeurs testées pour chacun des paramètres. Par exemple, le jeu de paramètres suivant impliquerait six tests, un pour chaque combinaison possible, dans cet ordre bien précis.

Jeu de paramètre :

- a) **F** = [1, 2] N
- b) **D** = [1, 2, 3]

Tests :

Test 1 : **F** = 1 N, **D** = 1

Test 2 : $F = 1 \text{ N}$, $D = 2$

Test 3 : $F = 1 \text{ N}$, $D = 3$

Test 4 : $F = 2 \text{ N}$, $D = 1$

Test 5 : $F = 2 \text{ N}$, $D = 2$

Test 6 : $F = 2 \text{ N}$, $D = 3$

Les abrasifs qui seront testés sont les sphères abrasives présentées au chapitre 5 (Voir Figure 5.24 et Figure 5.25).

Les premiers tests seront faits sans rotation de l'abrasif, lorsque la plage de paramètres à tester sera restreinte, des tests avec l'outil rotatif en marche seront faits. De plus, lorsqu'il sera question de définir la vitesse de rotation idéale et la force à appliquer, l'expertise d'autres personnes chez P&WC sera mise à contribution pour limiter l'envergure des tests à effectuer.

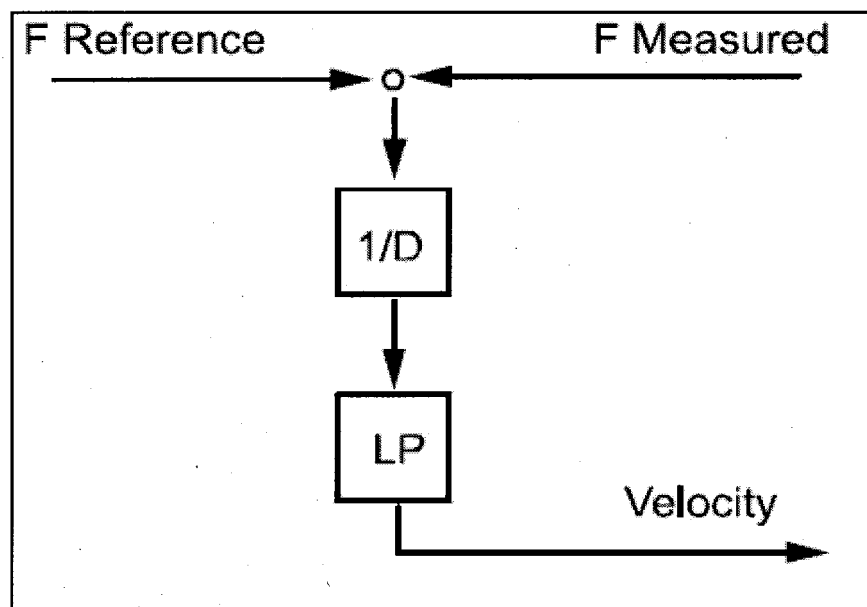


Figure 6.8 Effet du taux d'amortissement et de la fréquence de coupure du filtre passe-bas sur la commande.

Tirée de Application Manual,
Force Control for Machining (ABB, 2005, p. 38)

Finalement, deux types de fonctions pour le contrôle de force sont disponibles dans le contrôleur : suivi de trajectoire avec force constante et changement de vitesse pour suivre une trajectoire. Par contre, aucune de ces deux solutions n'est faite expressément pour répondre à la problématique actuelle. Il faudra feindre l'ébavurage sur une trajectoire avec force constante et interrompre son exécution prématurément pour réaliser la séquence présentée précédemment. Il demeure que les huit paramètres présentés au début de la section devront être ajustés pour réaliser l'opération.

6.3 Tests sans rotation de l'abrasif

L'objectif premier de cette série de tests est de confirmer la capacité du système robot-capteur de force à effectuer l'ébavurage des petits trous. Ce faisant, l'approche et les valeurs les plus susceptibles de donner un bon résultat doivent être cernées. Une fois ces valeurs trouvées, des tests avec l'outil rotatif en marche devront être effectués.

6.3.1 Évaluation de l'influence des paramètres **F**, **D** et **LP**

Voici les valeurs prises par chacun des paramètres lors de cette première série de tests (toutes les combinaisons n'ont pas été testées, certaines valeurs ont été rapidement éliminées des tests) :

- a) **Abrasif** = [1, 2]
- b) **F** = [2, 3, 5, 10, 15] N
- c) **V** = [12.5] mm/s
- d) **D** = [50, 100, 500, 1000]
- e) **LP** = [2, 5, 10, 50] Hz
- f) **FCS** = [1, 5, 10, 50] N/s
- g) **FCE** = [10, 50] N/s
- h) **ZCF** = [20] * **F**

Les valeurs de force (**F**) testées ont été inspirées de tests faits à la main, les valeurs de tests pour les autres paramètres viennent de recommandation d'experts chez ABB.

L'effet de chacun des paramètres sera mis en évidence par les premiers tests. La Figure 6.9 a été produite pour les valeurs suivantes. Une séquence avec l'approche horizontale est faite pour chaque combinaison possible (16 allers-retours dans un trou avec 16 combinaisons de **D** et **LP** différentes) :

- a) **Abrasif** = [2]
- b) **F** = [5] N
- c) **V** = [12.5] mm/s
- d) **D** = [50, 100, 500, 1000]
- e) **LP** = [50, 10, 5, 2] Hz *Notez l'ordre décroissant
- f) **FCS** = [10] N/s
- g) **FCE** = [50] N/s
- h) **ZCF** = [20] * **F**

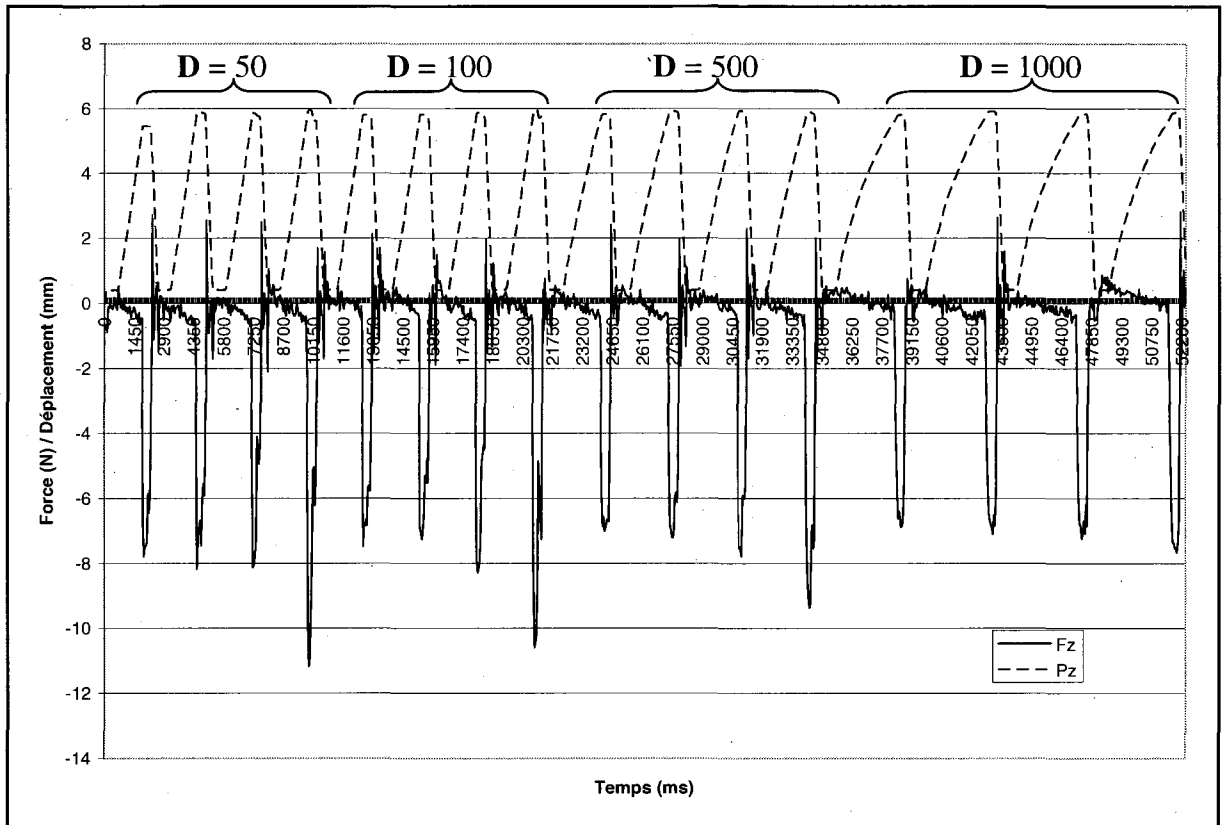


Figure 6.9 Force et position selon l'axe Z pour 16 combinaisons de D et LP, avec $F = 5$ N.

Premier constat, la force commandée est systématiquement dépassée (les valeurs de force lues ne sont pas filtrées ni amorties) peu importe la combinaison de **D** et **LP**. Elle l'est d'autant plus pour le quatrième aller-retour de chaque série, donc lorsque la fréquence de coupure du filtre passe-bas est faible. Ce qui concorde avec le fonctionnement du filtre, plus sa fréquence de coupure est basse, plus il y a d'information de perdue, moins le contrôleur est en mesure de respecter la contrainte de force. Malgré un dépassement de la force commandée allant jusqu'à un peu plus de 200%, les paramètres testés ne sont pas automatiquement rejetés puisque la force requise pour répondre aux spécifications n'est pas connue et doit aussi être déterminée. L'attention sera donc posée sur la constance et la qualité des courbes obtenues plutôt que sur la valeur de la force réelle. Des tests présentés plus loin permettront de déterminer la force qui doit être commandée pour obtenir un fini de surface acceptable selon les spécifications présentées à la section 6.2.1.

En ce qui concerne le déplacement, l'accroissement du taux d'amortissement ralentit le procédé et limite l'impact du filtre sur la force lue. Les dépassements de force sont donc plus constants (Voir la dernière série de quatre allers-retours sur la Figure 6.9). Un gros plan sur la quatrième séquence (Voir Figure 6.10) montre un comportement particulier, à la sortie du trou, qui est minimisé par un taux d'amortissement plus élevé.

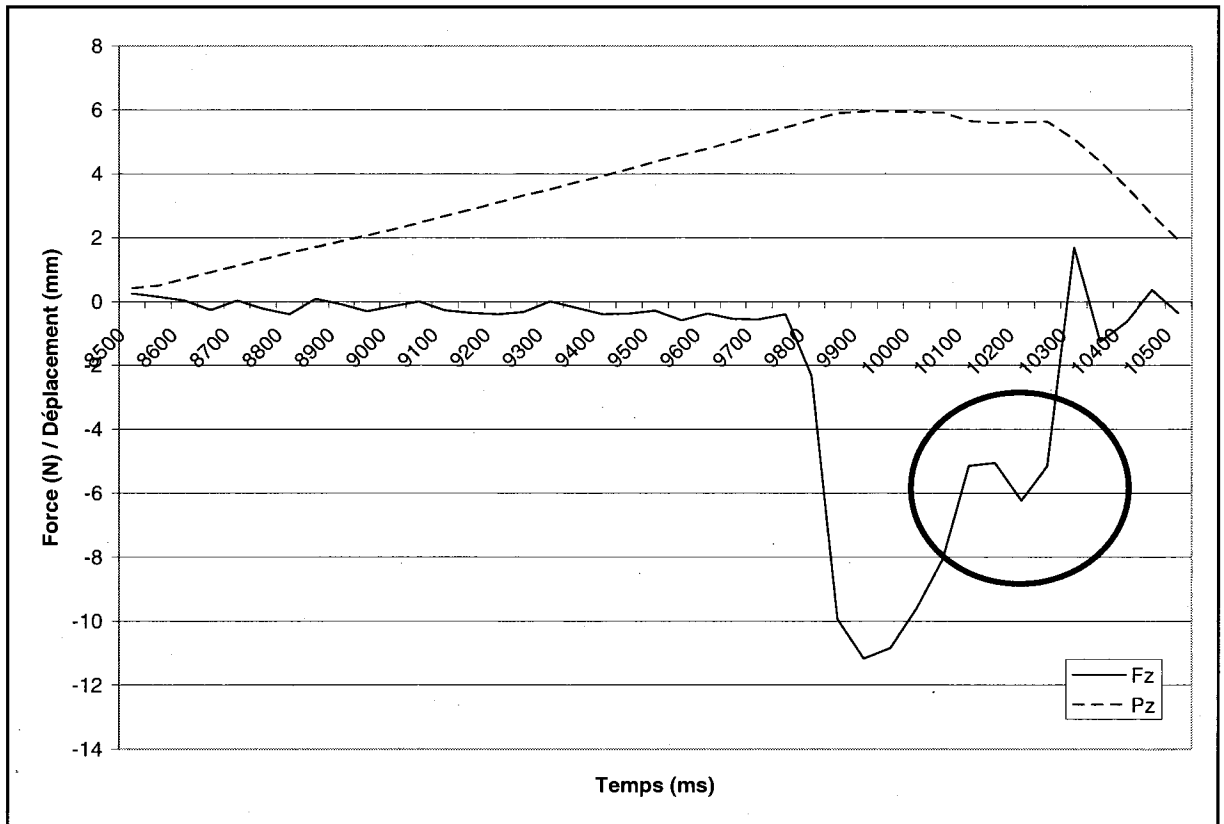
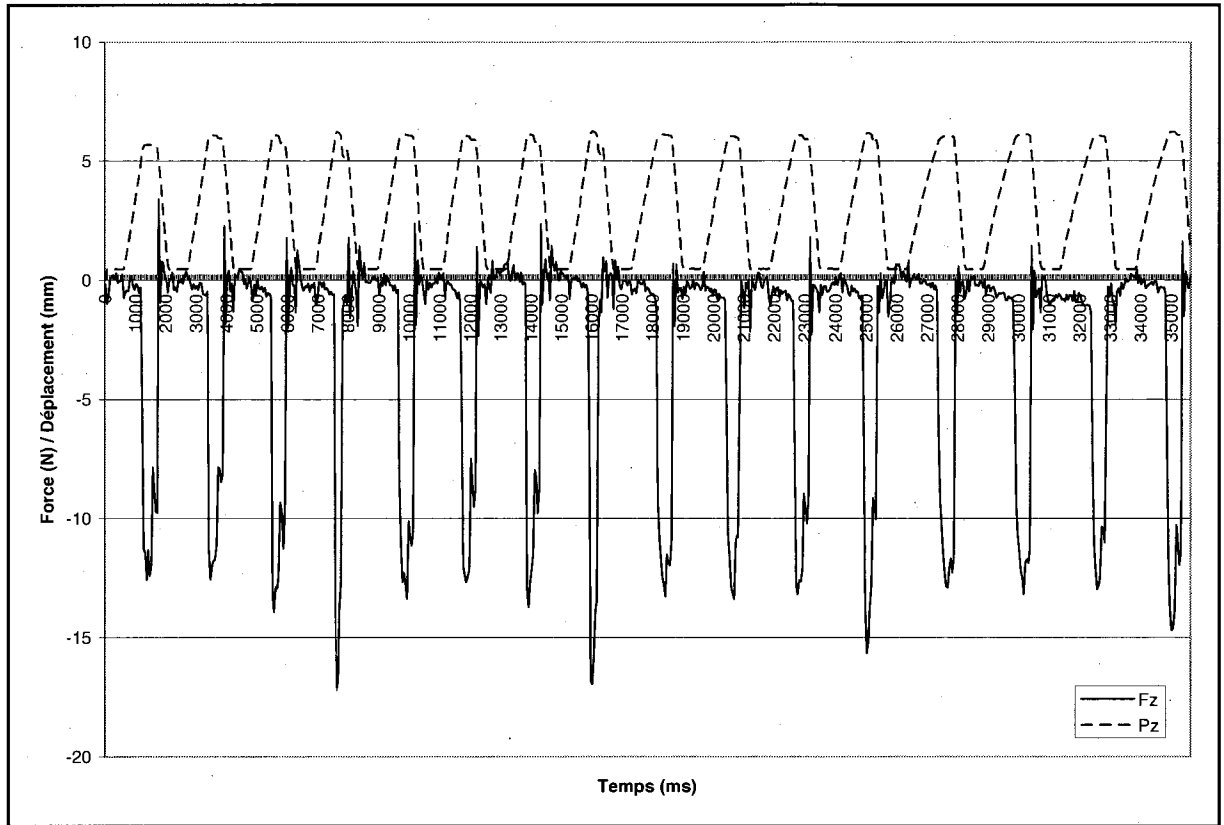


Figure 6.10 Comportement particulier lors d'un aller-retour.

Les mêmes tests ont été faits en variant la force à 10 N (Voir Figure 6.11) et 15 N (Voir Figure 6.12). La différence majeure est la diminution du dépassement de la commande en force. Ceci indique que la force commandée de 5 N est près de la limite de ce que le capteur nous permet de commander. Une étude plus exhaustive sera faite à la fin de la prochaine section pour déterminer quelle est la limite inférieure de la force qu'il est possible de commander.



**Figure 6.11 Force et position selon l'axe Z pour 16 combinaisons de D et LP.
avec $F = 10$ N.**

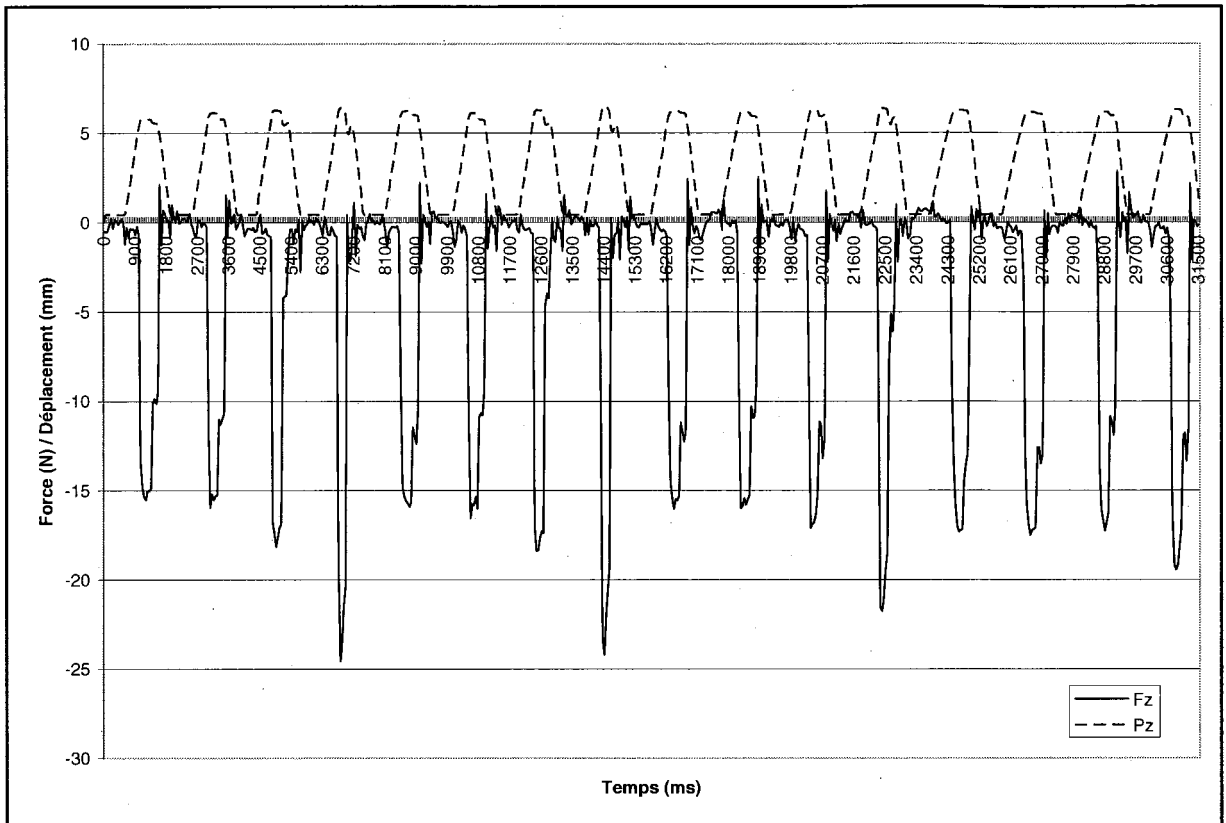


Figure 6.12 Force et position selon l'axe Z pour 16 combinaisons de D et LP, avec $F = 15$ N.

Le phénomène présenté à la Figure 6.10 est aussi présent sur les figures suivantes. On constate que le maximum de force atteint l'est pour une durée plus courte et qu'un effet de ressort qui ramène l'abrasif dans le trou est aussi notable. Si l'opération est comparée à ce qui se fait manuellement, il est désirable d'avoir une touche rapide telle qu'obtenue avec cette configuration de paramètres. Par contre, l'effet indésirable qui s'en suit doit être corrigé en ajustant les paramètres de retrait, ce qui sera aussi corrigé.

De manière générale, on remarque aussi une usure rapide et localisée de l'abrasif lorsque l'approche horizontale est utilisée, ce qui ne sera pas le cas avec une approche verticale. La même série de test a été faite pour l'abrasif 1. Les résultats étaient en tous points semblables à ceux obtenus précédemment. Puisque l'abrasif 2 est plus utilisé actuellement, il sera conservé pour la suite des tests.

Finalement, en ce qui concerne les configurations testées, c'est celle-ci qui offre la meilleure qualité de courbe (*Voir* Figure 6.13) :

- a) **Abrasif** = [2]
- b) **F** = [5] N
- c) **V** = [12.5] mm/s
- d) **D** = [1000]
- e) **LP** = [50] Hz
- f) **FCS** = [10] N/s
- g) **FCE** = [50] N/s
- h) **ZCF** = [20] * **F**

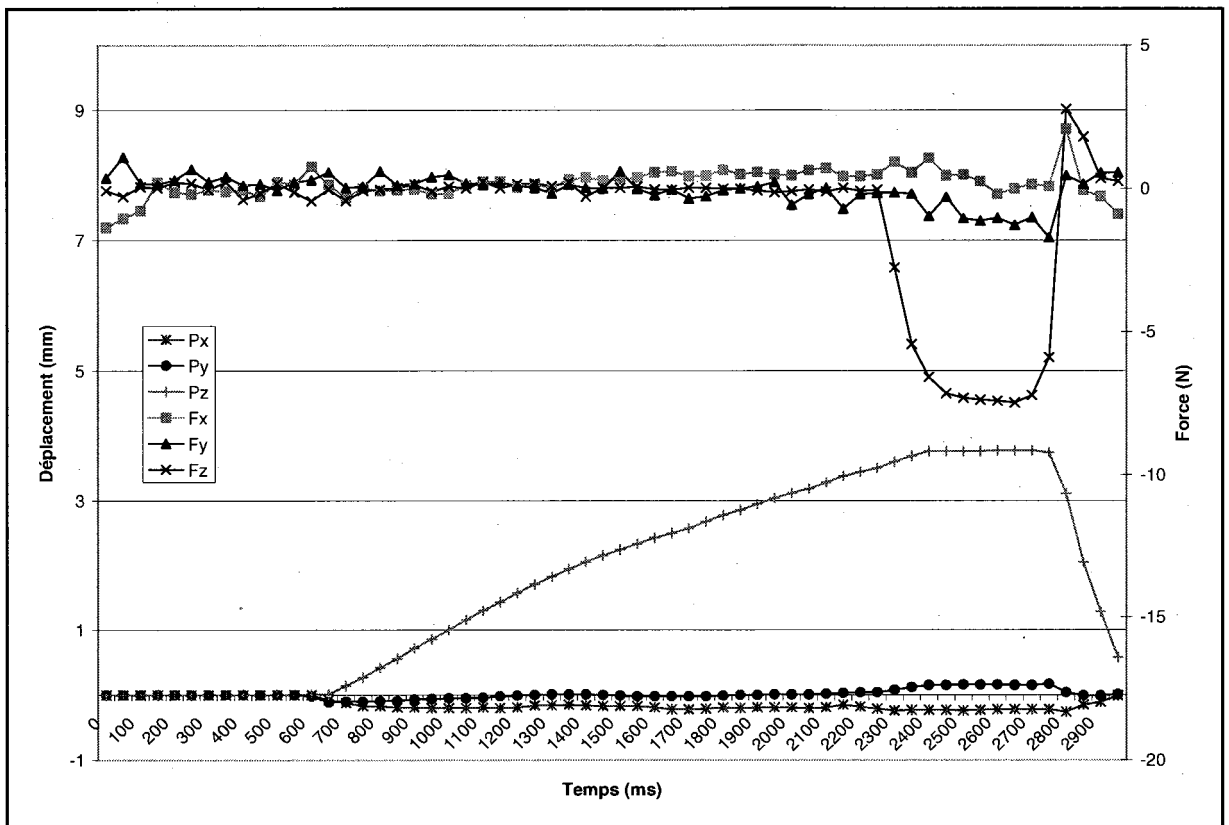


Figure 6.13 Force et position pour une combinaison optimale de D et LP, avec $F = 5$ N.

Lorsqu'une attention particulière est portée aux courbes P_x et P_y , on constate une erreur en position susceptible d'affecter la précision du travail (Voir Figure 6.14). À priori, la caméra serait la principale raison de l'imprécision lors de l'entrée dans le trou. Par contre, la compensation ajoutée à la position retournée par la caméra a démontré son efficacité lorsque des petits trous sont testés sur différentes pièces du même numéro de série.

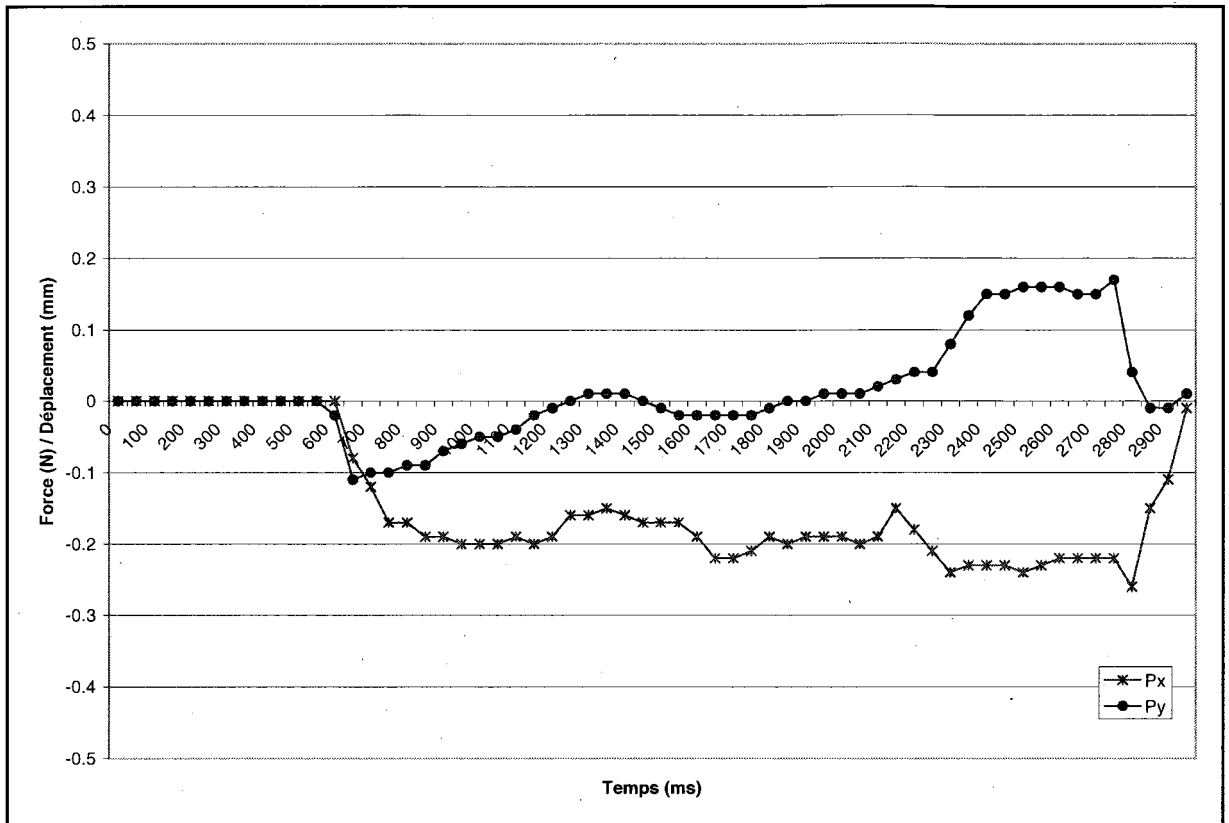


Figure 6.14 Erreur en position selon les axes X et Y.

Une série de tests destinés à évaluer l'impact des instructions impliquant le contrôle de force a permis d'apprécier le travail d'asservissement du contrôleur de mouvement et de définir la source de cette erreur de positionnement. Pour ce faire une approche horizontale et un abrasif rigide conique ont été utilisés. Le dernier de ces tests a été fait en fixant différemment un ensemble de câbles qui sont reliés à l'outil rotatif. Ces câbles créaient une force non désirée qui était en grande partie responsable de l'erreur de positionnement (Voir Figure 6.15). On constate que les seules erreurs de positionnement en P_x et P_y (l'axe Y de gauche) apparaissent lors des accélérations et décélérations du robot (P_z sur l'axe Y de droite).

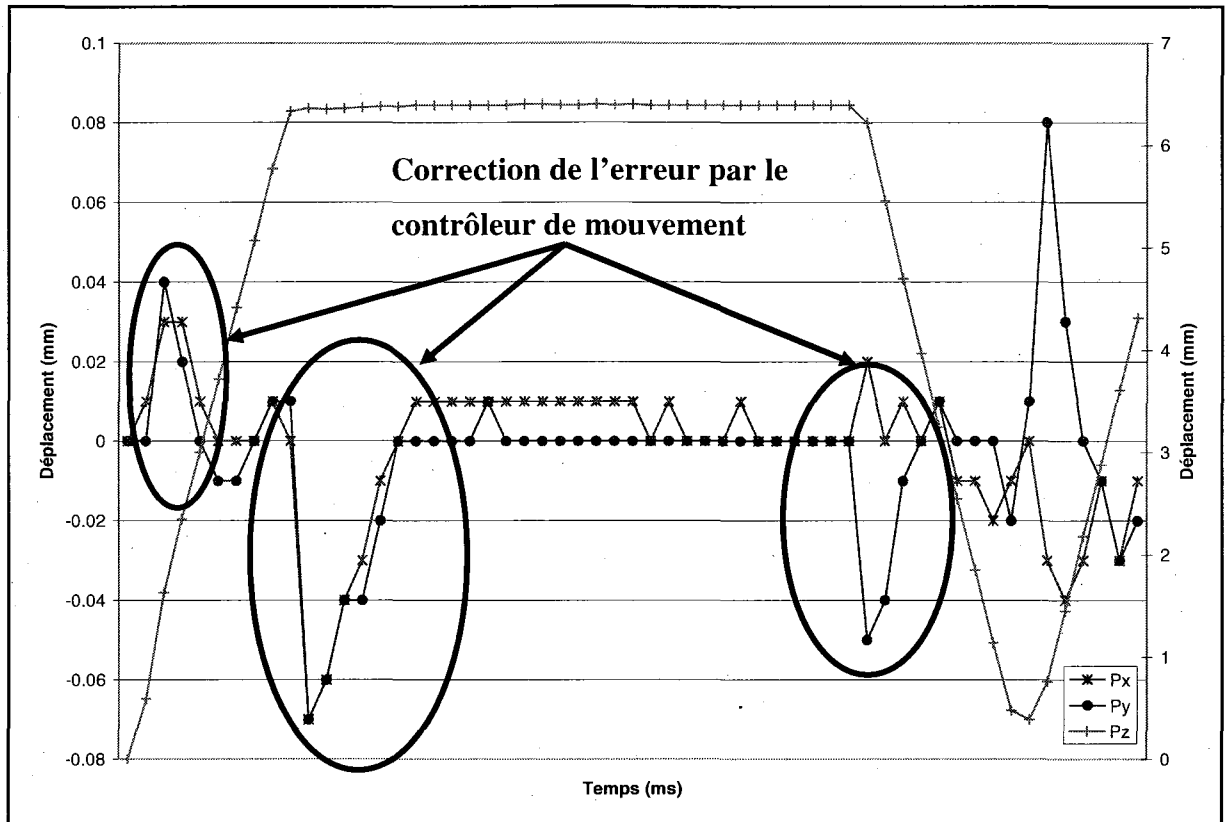


Figure 6.15 Correction de l'erreur en position lors de l'application de la force désirée.

Une fois cette interrogation levée et les paramètres optimaux trouvés, l'étape suivante est d'évaluer le comportement du système lorsque l'abrasif est en rotation.

6.4 Ébavurage robotisé

Pour les tests faits avec rotation de l'abrasif, l'outil rotatif tournera à une vitesse constante, fixée selon les recommandations du fournisseur de l'abrasif et de l'expertise présente chez P&WC.

Les tests faits sans rotation de l'abrasif ont confirmé l'ordre de grandeur des valeurs à utiliser. Voici donc les différentes valeurs qui ont été testées avec l'abrasif en rotation :

- a) **Abrasif** = [2]
- b) **F** = [5] N
- c) **V** = [1.25, 4, 12.5, 25] mm/s
- d) **D** = [50, 100, 500, 1000]
- e) **LP** = [2, 5, 10, 50] Hz
- f) **FCS** = [1, 5, 10, 50] N/s
- g) **FCE** = [10] N/s
- h) **ZCF** = [20] * **F**

Les courbes obtenues nous permettent de poser certaines conclusions :

- 1- La vitesse (**V**) influence la durée de l'opération et la stabilité si elle est trop élevée.
- 2- Le taux d'amortissement (**D**) permet de stabiliser le procédé, mais le ralentit lorsque le contrôle de force est en fonction.
- 3- La fréquence de coupure du filtre passe-bas (**LP**) n'a pas eu d'effet notable, mais les experts chez ABB nous ont fortement conseillé de la fixer à 2 Hz pour éviter les problèmes liés à la stabilité du robot.
- 4- Le taux de variation de la force lors de l'approche (**FCS**) permet d'accélérer l'attente de la force désirée, mais peut créer un dépassement lors du contact initial.
- 5- Le taux de variation de la force lors du retrait (**FCE**) contrôle la vitesse à laquelle le robot tentera d'atteindre une force de zéro à la fin de la séquence, ce paramètre ne devrait plus être utilisé dès que la force ressentie passe sous la force où l'on considère qu'il n'y a plus de contact (**ZCF**).
- 6- La force à partir de laquelle le contrôle de force est désactivé (**ZCF**) doit être à une valeur assez élevée pour que le contrôle de force soit désactivé tout de suite après le dernier point où l'ébavurage est fait.

Les meilleurs résultats ont été atteints avec la combinaison suivante (Voir Figure 6.16) :

- a) **Abrasif** = [2]
- b) **F** = [5] N
- c) **V** = [12.5] mm/s
- d) **D** = [500]
- e) **LP** = [2] Hz
- f) **FCS** = [10] N/s
- g) **FCE** = [10] N/s
- h) **ZCF** = [20] * F

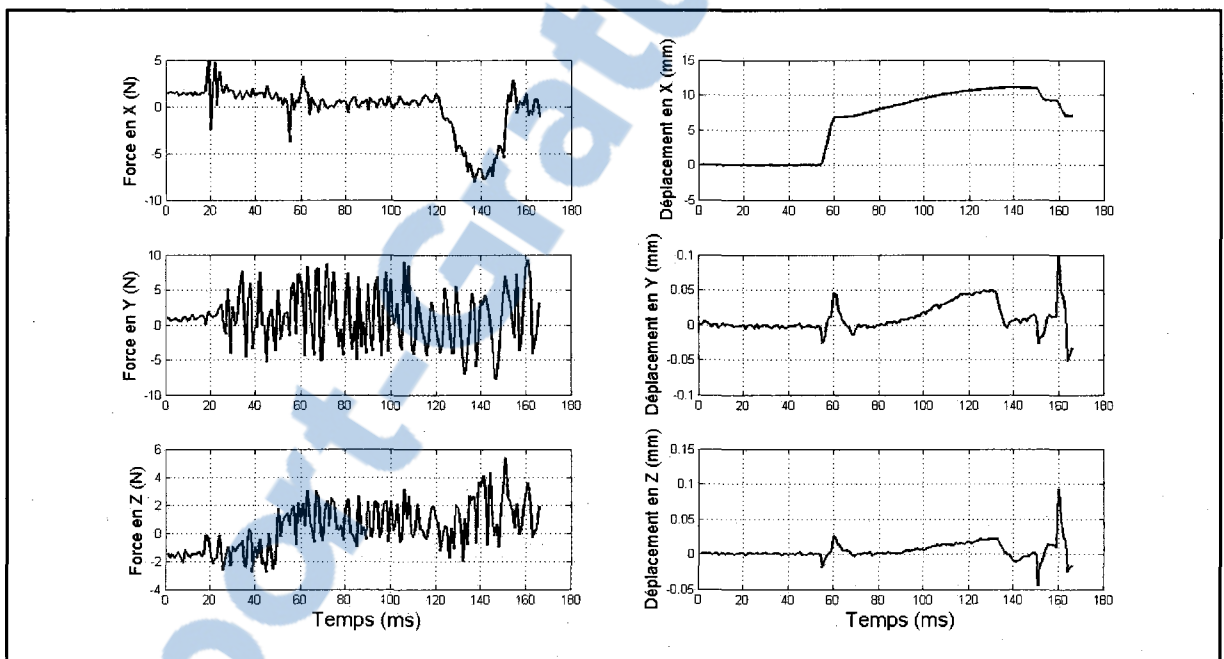


Figure 6.16 Résultats de l'ébavurage robotisé avec F = 5 N.

Ce résultat démontre l'effet important de l'activation de l'outil rotatif sur la force en Y. Heureusement le déplacement s'effectue en X dans ce cas-ci. De plus, les déplacements en Y et Z sont très près de zéro, le problème de déplacements non désirés est donc bel et bien réglé.

Malgré que la force commandée ne soit dépassée que de quelques Newtons, le fini de surface sur la pièce montre un ébavurage un peu trop agressif. La force de contact sera donc

diminuée à 3 N (Voir Figure 6.17). Aussi, lors du retrait de l'abrasif, on voit un plateau qui est le résultat d'un délai causé par le transfert du contrôle force/position à un contrôle uniquement en position. Malgré plusieurs valeurs testées, ce comportement était toujours présent. Il est possible qu'avec un ajustement plus fin des paramètres de sortie (**FCE** et **ZCF**) ce plateau puisse être éliminé.

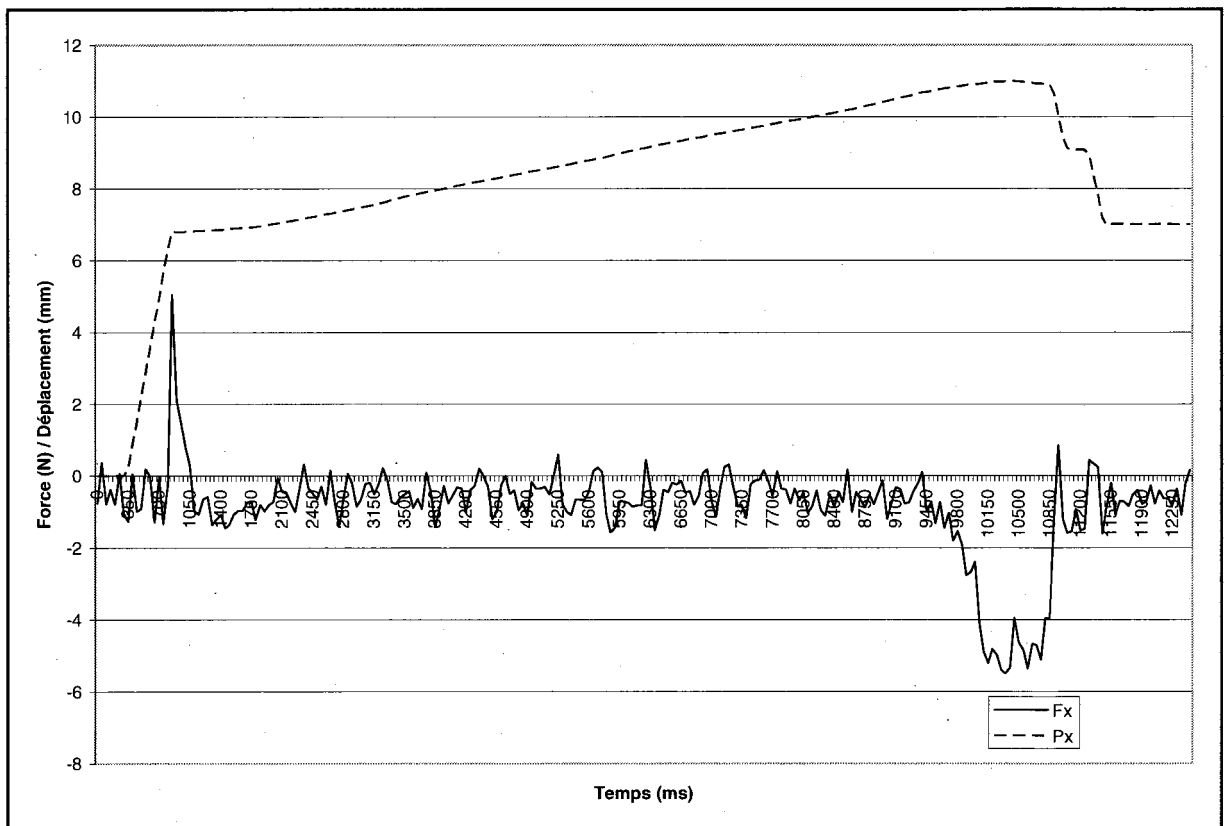


Figure 6.17 Résultats de l'ébavurage robotisé avec $F = 3$ N.

Les trous résultants répondent aux exigences énoncées à la section 6.2.1 et sont considérés comme bons selon le département d'inspection chez P&WC (Voir Figure 6.18).

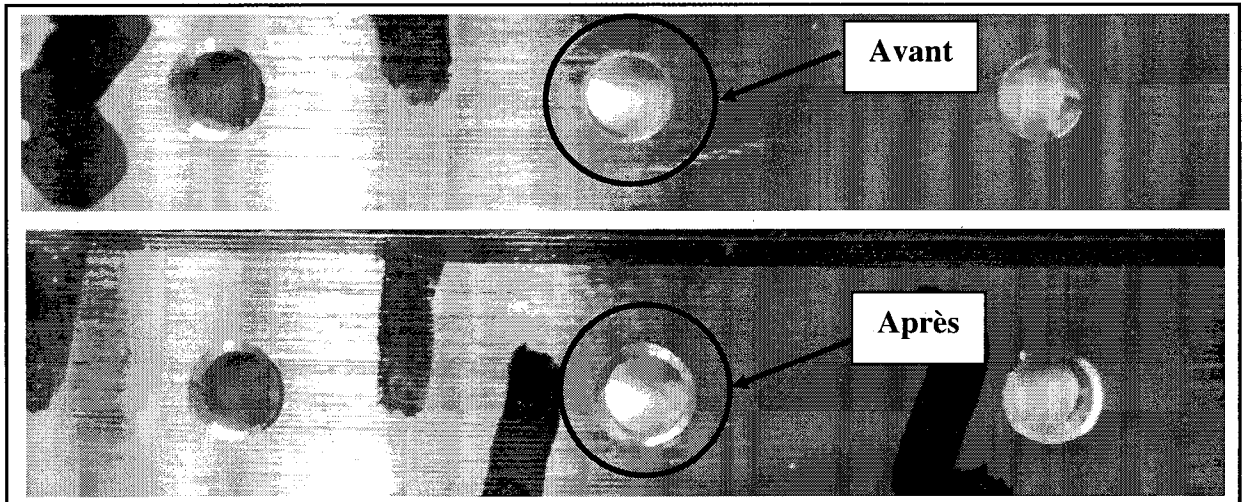


Figure 6.18 Image d'un trou avant et après ébavurage.

Finalement, quelques autres tests ont été faits pour évaluer la faisabilité de commander des forces de 2 N et 1.5 N. Comme ces forces sont faibles et près du bruit perçu par le capteur, les paramètres ont dû être minutieusement ajustés. Par exemple, voici la combinaison de paramètres à fournir pour une commande de 1.5 N

- a) **Abrasif** = [2]
- b) **F** = [1.5] N
- c) **V** = [12.5] mm/s
- d) **D** = [1000]
- e) **LP** = [2] Hz
- f) **FCS** = [1] N/s
- g) **FCE** = [10] N/s
- h) **ZCF** = [20] * F

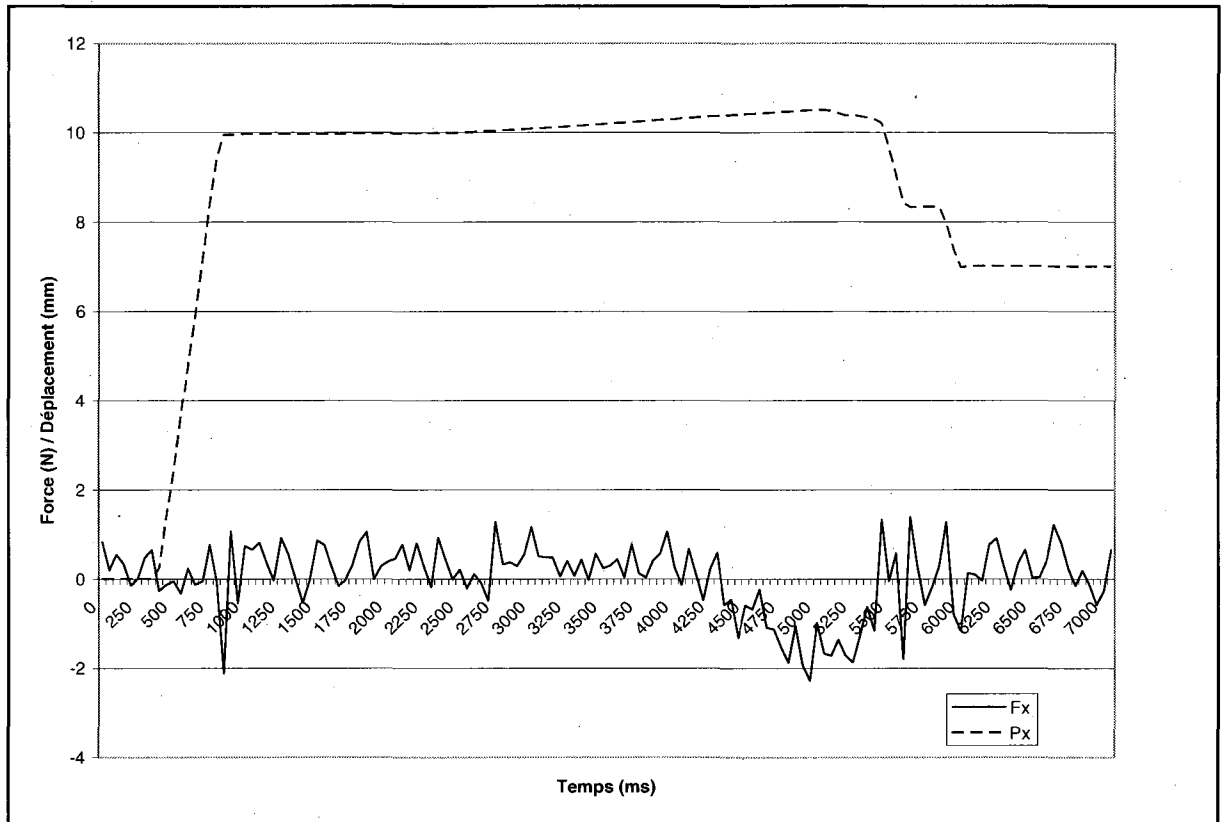


Figure 6.19 Résultats de l'ébavurage robotisé avec $F = 1.5$ N.

On note que la configuration nécessaire pour commander une force de 1.5 N ralentit énormément le procédé. De plus, une fois inspectés, les trous ébavurés avec cette configuration n'ont pas été à la hauteur de ceux faits avec une force de 3 N.

6.5 Conclusion

L'ébavurage robotisé avec contrôle de force et abrasifs rigides est donc rendu possible grâce aux différentes technologies développées et intégrées ici. De plus, la cellule robotisée offre une grande flexibilité permettant la création de trajectoires hors-ligne pour n'importe quelle autre géométrie et n'importe quelle autre pièce. L'architecture logicielle, la caméra et le capteur de force permettent d'utiliser le code généré hors-ligne et de le mettre en œuvre directement dans la cellule robotisée.

Plusieurs interrogations par rapport à l'influence de la variation des bavures avaient été soulevées au tout début du projet, mais les différents tests faits, avec plusieurs pièces différentes et quelques géométries différentes, n'ont jamais souffert de la variation des dimensions des bavures. On en conclut que la force nécessaire pour effectuer la tâche dans les tolérances requises permet de ne pas avoir à se soucier de cet aspect. Aucune quantification dimensionnelle des bavures n'a donc été faite. De plus, les tests présentés ici sont au niveau de l'extérieur des trous. L'intérieur des trous présente aussi des bavures, généralement plus importantes, et des résultats satisfaisant les exigences de l'inspection ont aussi été obtenus sans avoir à modifier l'approche pour tenir compte des variations de bavures.

Finalement, plusieurs autres projets touchant l'ébavurage sont en cours chez P&WC. L'infrastructure mise en place permettra d'accélérer les nombreux tests qui devront être faits.

CONCLUSION

Le présent travail portait donc sur l'amélioration de la précision des bras robotisé pour une application d'ébavurage robotisé. D'ailleurs, cette problématique est particulièrement présente dans le domaine de l'aéronautique. Les enjeux sont au niveau de la précision dimensionnelle des différentes géométries sur les pièces et de la capacité des robots à atteindre ces tolérances désirées.

L'amélioration de la précision robotique par la calibration des dimensions du robot et par la calibration du TCP des outils a été explorée. Des simulations ont été effectuées pour démontrer la qualité relative des solutions testées. Entre autres, il en ressort que le bruit causé par la linéarisation est trop important pour que la précision du robot puisse être améliorée en n'utilisant que les informations provenant du robot. Un comparatif est dressé et c'est la technique de calibration du TCP du fabricant du robot qui offre la meilleure précision dans le cas étudié.

Ensuite, les différents éléments utilisés pour arriver à réaliser l'opération d'ébavurage robotisé sont présentés et caractérisés. L'intégration de la cellule robotique et l'architecture logicielle développée pour l'application sont expliquées. L'évolution des différents tests est présentée pour expliquer comment l'ébavurage robotisé de petits trous a pu être possible. Les résultats présentés poussent à la conclusion que l'utilisation de la caméra et du capteur de force est nécessaire pour pallier à l'imprécision du robot et pour arriver à un résultat acceptable selon l'inspection faite chez P&WC.

L'accent est mis sur une seule géométrie alors qu'il en existe une dizaine, avec différentes contraintes d'accessibilité et tolérances, ce qui signifie que beaucoup de développement reste encore à faire. L'utilisation d'autres outils rigides est présentement en cours chez P&WC et présente déjà des résultats positifs. De plus, les infrastructures mises en place permettent d'accélérer les tests et faciliteront le passage de la phase développement à la phase de mise en production.

RECOMMANDATIONS

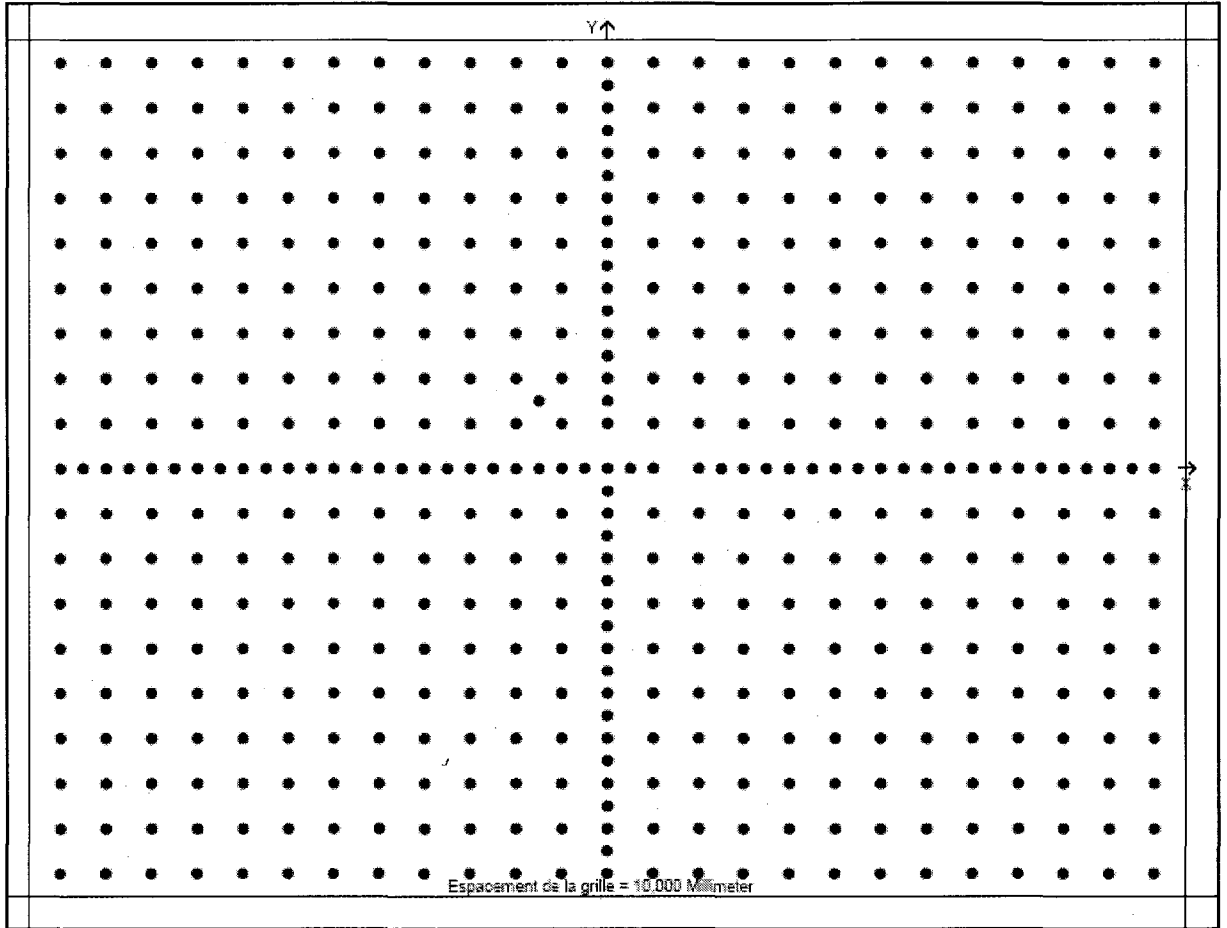
Dans l'optique de poursuivre le travail présenté dans ce mémoire, voici quelques avenues qui pourraient être explorées pour accroître l'intégration de bras robotisé en industrie :

- Ajouter un dispositif permettant l'inspection pour évaluer la qualité du travail fait par le système robotisé.
- Étudier la possibilité d'utiliser une caméra fixe pour le positionnement de la pièce, le positionnement de l'outil, la supervision de l'opération et l'inspection.
- Étudier la possibilité d'utiliser un système de caméra pour suivre et positionner le robot avec plus de précision.
- Comparer les différentes techniques faciles à mettre en application en milieu industriel permettant de calibrer une caméra sur un robot.

Du côté de l'opération d'ébavurage, beaucoup de travail exploratoire doit être fait pour évaluer et cerner les abrasifs permettant les meilleurs résultats (fini de surface par rapport au nombre de passes nécessaires, à l'usure de l'abrasif, etc.) selon les matériaux et géométries qui doivent être travaillés.

ANNEXE I

GRILLE DE CALIBRATION DE LA CAMÉRA



LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- ABB. 1997. *Method and device for calibration of movement axes of an industrial robot*. Snell, John-Erik (inv.). United States Patents 5687293.
<http://www.google.com/patents/about?id=n9QbAAAAEBAJ&dq=robot*+calibration+sphere&as_drrb_ap=q&as_minm_ap=0&as_miny_ap=&as_maxm_ap=0&as_maxy_ap=&as_drrb_is=q&as_minm_is=0&as_miny_is=&as_maxm_is=0&as_maxy_is=&num=50>.
- ABB. 2001. *Device and method for calibration of a multi-axis industrial robot*. Brogardh, Torgny, et John-Erik Snell (inv.). United States Patent 6205839.
<http://www.google.com/patents/about?id=OycFAAAEBAJ&dq=robot*+calibration+sphere&as_drrb_ap=q&as_minm_ap=0&as_miny_ap=&as_maxm_ap=0&as_maxy_ap=&as_drrb_is=q&as_minm_is=0&as_miny_is=&as_maxm_is=0&as_maxy_is=&num=50>.
- ABB. 2005. « Application manual: Force Control for Machining ». En ligne.
<http://www.abbrobots.co.uk/files/3HAC027595-001_revA_en.pdf>.
Consulté le 2009-09-17.
- ABB. 2010. *ABB Robotics: ABB in Canada*. En ligne.
<<http://www.abb.ca/product/seitp327/7c4717912301eb02c1256efc00278a26.aspx>>.
Consulté le 2010-05-10.
- Abderrahim, M., et A. R. Whittaker. 2000. « Kinematic model identification of industrial manipulators ». *Robotics and Computer-Integrated Manufacturing*, vol. 16, p. 1-8.
- Bennett, David J., John M. Hollerbach et Paul D. Henri. 1992. « Kinematic calibration by direct estimation of the Jacobian matrix ». In *Proceedings 1992 IEEE International Conference on Robotics and Automation, May 12-14 1992*. Vol. 1, p. 351-357. IEEE, Piscataway, NJ, USA. <<http://dx.doi.org/10.1109/ROBOT.1992.220314>>.
Consulté le 2008-09-15.
- Bernhardt, Rolf, et S. L. Albright. 1993. *Robot calibration*, 1st. London ; New York: Chapman & Hall, xii, 311 p.
<http://books.google.ca/books?id=A3ksLnOEK4C&printsec=frontcover&dq=robot+calibration&ei=3xrISJ6HMPWyyOTgi53iDg&hl=fr&sig=ACfU3U25pNoLs_rMcLV_PBc7jydYtEfdyxQ#PPT7,M1>. Consulté le 2008-09-02.
- Caenen, J. L., et J. C. Angue. 1990. « Identification of geometric and non geometric parameters of robots ». In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation, May 13-18 1990*. p. 1032-1037. IEEE, Los Alamitos, CA, USA. <<http://dx.doi.org/10.1109/ROBOT.1990.126128>>. Consulté le 2008-09-29.

- Chiaverini, Stefano, Bruno Siciliano et Luigi Villani. 1994. « Force/position regulation of compliant robot manipulators ». *IEEE Transactions on Automatic Control*, vol. 39, n° 3, p. 647-652.
- Conrad, Kevin L., Panayiotis S. Shiakolas et T. C. Yih. 2000. « Robotic calibration issues: Accuracy, Repeatability and Calibration ». In *Proceeding of the 8th Mediterranean Conference on Control & Automation (MED 2000)* (Rio, Patras, GREECE, 17-19 July 2000). Rio, Patras, Greece. Consulté le 2008-08-12.
- Corke, P.I. 1996. « A Robotics Toolbox for MATLAB ». *IEEE Robotics and Automation Magazine*, vol. 3, n° 1 (mars), p. 24-32.
- Craig, John J. 2005. *Introduction to robotics : mechanics and control*, 3rd. Upper Saddle River, N.J.: Pearson/Prentice Hall, viii, 400 p. Consulté le 2007-10-16.
- Denavit, J., et R. S. Hartenberg. 1955. « A kinematic notation for lower-pair mechanisms based on matrices ». *Trans ASME J. Appl. Mech*, vol. 23, p. 215-221.
- Duelen, G., et K. Schroer. 1991. « Robot calibration. Method and results ». *Robotics and Computer-Integrated Manufacturing*, vol. 8, n° 4, p. 223.
- Gatla, Chandra Sekhar, Ron Lumia, John Wood et Greg Starr. 2007. « An automated method to calibrate industrial robots using a virtual closed kinematic chain ». *IEEE Transactions on Robotics*, vol. 23, n° 6, p. 1105-1116.
- Hanqi, Zhuang, S. H. Motaghedi et Z. S. Roth. 1999. « Robot calibration with planar constraints ». In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. Vol. 1, p. 805-810 vol.1. Consulté le 2009-09-20.
- Hayati, S., et M. Mirmirani. 1985. « Improving the absolute positioning accuracy of robot manipulators ». *Journal of robotic systems* vol. vol. 2, n° no4, p. pp. 397-413
- Hollerbach, John M., Lydia Giygovaz, Martin Buehler et Yangming Xy. 1993. « Screw axis measurement for kinematic calibration of the Sarcos dexterous arm ». In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Jul 26-30 1993*. p. 1617-1621. IEEE, Piscataway, NJ, USA.
<<http://dx.doi.org/10.1109/IROS.1993.583854>>. Consulté le 2008-09-30.
- Ikits, M., et J. M. Hollerbach. 1997. « Kinematic calibration using a plane constraint ». In *Proceedings of International Conference on Robotics and Automation, 20-25 April 1997*. Vol. vol.4, p. 3191-6. IEEE.
<<http://dx.doi.org/10.1109/ROBOT.1997.606774>>.

- Khalil, W., P. Lemoine, M. Gautier et al. 1996. « Autonomous calibration of Robots using planar points ». In *World Automation Congress, WAC'96 - Robotic and Manufacturing Systems*, ISBN : 1-889335-00-2 (1996-05-28). hal-00362605, version 1. Vol. 3, p. pp. 383-388. <<http://hal.archives-ouvertes.fr/hal-00362605/en/>>.
- Khatib, Oussama, et Joel Burdick. 1986. « MOTION AND FORCE CONTROL OF ROBOT MANIPULATORS ». In *Proceedings - 1986 IEEE International Conference on Robotics and Automation*. p. 1381-1386. IEEE, New York, NY, USA. Consulté le 02 juillet 2008.
- Liao, Liang, Xi FengFeng et Kefy Liu. 2008. « Modeling and control of automated polishing/deburring process using a dual-purpose compliant toolhead ». *International Journal of Machine Tools and Manufacture*.
- Liu, Guangjun, Karl Iagnemma, Steven Dubowsky et Guillaume Morel. 1998. « Base force/torque sensor approach to robot manipulator inertial parameter estimation ». In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Part 4 (of 4), May 16-20 1998*. Vol. 4, p. 3316-3321. IEEE, Piscataway, NJ, USA. Consulté le 2008-07-02.
- McKerrow, Phillip John. 1991. *Introduction to robotics*. Coll. « Electronic systems engineering series ». Reading, Mass.: Addison-Wesley, xvi, 811 p. Consulté le 2008-10-16.
- Mirman, C. R., et K. C. Gupta. 1992. « Compensation of robot joint variables using special Jacobian matrices ». *Journal of Robotic Systems*, vol. 9, n° 1, p. 113-137.
- Mooring, Benjamin , Morris Driels et Zvi Roth. 1991. *Fundamentals of Manipulator Calibration*. John Wiley & Sons, Inc., 329 p.
- Motta, J. M. S. T., G. C. De Carvalho et R. S. McMaster. 2001. « Robot calibration using a 3D vision-based measurement system with a single camera ». *Robotics and Computer-Integrated Manufacturing*, vol. 17, n° 6, p. 487-497.
- Ramming, John, Stephen Rauch, Ricardo Araujo et J. Norberto Pires. 2002. « Force/torque sensing applied to industrial robotic deburring ». *Sensor Review*, vol. 22, n° 3, p. 232-241.
- Raucent, B., J. C. Samin et R. Gorez. 1991. « Influence of inertial parameters on the positional accuracy of a robot ». In *Fifth International Conference on Advanced Robotics - '91 ICAR, Jun 19-22 1991*. p. 946. IEEE, Piscataway, NJ, USA. <<http://dx.doi.org/10.1109/ICAR.1991.240550>>. Consulté le 2008-09-08.

- Renders, Jean-Michel, Eric Rossignol, Marc Becquet et Raymond Hanus. 1991. « Kinematic calibration and geometrical parameter identification for robots ». *IEEE Transactions on Robotics and Automation*, vol. 7, n° 6, p. 721-732.
- Siciliano, Bruno, et Luigi Villani. 1999. *Robot force control / Bruno Siciliano and Luigi Villani*. Coll. « Kluwer international series in engineering and computer science SECS 540 ». Kluwer Academic. Consulté le 2008-06-16.
- Sun, Lei, Jingtai Liu, Weiwei Sun, Shuihua Wu et Xingbo Huang. 2004. « Geometry-based robot calibration method ». In *Proceedings- 2004 IEEE International Conference on Robotics and Automation, Apr 26-May 1 2004*. Vol. 2004, p. 1907-1912. Institute of Electrical and Electronics Engineers Inc., Piscataway, United States. Consulté le 2008-09-15.
- Vira, Naren, et Tyler Estler. 1990. « Use of a contact-type measurement device to detect robots' hand positions ». *ISA Transactions*, vol. 29, n° 4, p. 21-40.
- Wenbiao, Wang, Li Aiguo, Ma Zi, Xu Huipu et Tian Yuan. 2009. « An automated method to robot calibration using line-structure-light vision sensor ». In *2009 IEEE International Conference on Automation and Logistics (ICAL), 5-7 Aug. 2009*. p. 1896-9. IEEE. <<http://dx.doi.org/10.1109/ICAL.2009.5262655>>.
- Xianlun, Wang, Wang Yong et Xue Yunna. 2007. « Adaptive control of robotic deburring process based on impedance control ». In *2006 IEEE International Conference on Industrial Informatics, INDIN'06, Aug 16-18 2006*. p. 921-925. Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States. <<http://dx.doi.org/10.1109/INDIN.2006.275700>>. Consulté le 2008-08-01.
- Zhong, Xiao-Lin, et John M. Lewis. 1995. « New method for autonomous robot calibration ». In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation. Part 1 (of 3), May 21,1995 - May 27,1995*. Vol. 2, p. 1790-1795. IEEE. Consulté le 2009-03-17.
- Zhong, Xiao-Lin, John M. Lewis et Francis L.N.-Nagy. 1996. « Autonomous robot calibration using a trigger probe ». *Robotics and Autonomous Systems*, vol. 18, n° 4, p. 395-410.