



**HAL**  
open science

# Programmation et commande d'un robot à deux bras pour la manipulation et l'assemblage de pièces flexibles par des tâches collaboratives

Meryem Taghbalout

► **To cite this version:**

Meryem Taghbalout. Programmation et commande d'un robot à deux bras pour la manipulation et l'assemblage de pièces flexibles par des tâches collaboratives. Automatique / Robotique. Université de Lorraine, 2021. Français. NNT : 2021LORR0165 . tel-03428454

**HAL Id: tel-03428454**

**<https://hal.univ-lorraine.fr/tel-03428454v1>**

Submitted on 15 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Thèse

Présentée et soutenue publiquement pour l'obtention du titre de

**DOCTEUR DE L'UNIVERSITE DE LORRAINE**

Mention : Automatique

par **Meryem Taghbalout**

Sous la direction de **Gabriel Abba**

---

**Programmation et commande d'un robot  
à deux bras pour la manipulation et  
l'assemblage de pièces flexibles par des  
tâches collaboratives**

---

**Vendredi 03 septembre 2021**

**Membre du jury :**

Directeur de thèse :	M. Gabriel ABBA	Professeur, Université de Lorraine, Metz
Président de jury :	Mme Christine CHEVALLEREAU	Directeur de recherches, CNRS, Nantes
Rapporteurs :	M. Yannick Aoustin	Professeur, Université de Nantes, Nantes
	M. Philippe FRAISSE	Professeur, Université de Montpellier 2, Montpellier
Examineurs :	M. Mourad BENOUSAAD	Maître de conférences, Enit, Tarbes
	M. Jean-François ANTOINE	Maître de conférences, Université de Lorraine, Nancy

---



# Remerciement

Comme le veut la tradition, ces quelques lignes du manuscrit sont les plus agréables à écrire. Elles signifient que la thèse est enfin terminée et elles permettent de remercier toutes les personnes qui ont participé à sa réalisation.

Tout d'abord, je tiens à remercier mon directeur de thèse M. Gabriel ABBA, professeur à l'Université de Lorraine (UL) pour son soutien scientifique et ses conseils. Je remercie mon encadrant de thèse, M. Jean-François ANTOINE, maître de conférences à l'UL, pour son aide et son accompagnement pendant toute la durée de ma formation. Je tiens également à remercier les membres de la Robotix Academy pour leur accueil et leurs conseils avisés.

Mes sincères remerciements à M. Philippe FRAISSE et M. Yannick AOUSTIN pour m'avoir fait l'honneur de rapporter mon travail et aux examinateurs Mme Christine CHEVALLEREAU et M. Mourad BENOUSSAAD pour avoir accepté de consacrer leur précieux temps à l'évaluation du travail effectué durant mon doctorat.

Mes remerciements vont également à M. Vianney PAPOT, ingénieur de recherche au laboratoire LCFC, à Yiguo WANG et Jules TSANGA les deux stagiaires qui j'ai suivi pendant ma thèse, et à l'ensemble du personnel et des doctorants de l'UL et de l'ENSAM. Un grand merci à Atal KUMAR pour avoir partagé mon bureau pendant tout ce temps.

Enfin, je tiens également à exprimer ma gratitude à mes parents Rabha et Mohamed, mon frère Hamza, mes sœurs Samia, Najoua, Ilhame, Aya et mon mari Mohamed qui m'ont offert et m'offrent encore le soutien et la force nécessaires à la réalisation de tous mes projets.



# Table des matières

<b>Acronymes</b>	<b>9</b>
<b>1 État de l'art</b>	<b>15</b>
1.1 Les robots collaboratifs industriels	15
1.2 Robot collaboratif à deux bras	20
1.2.1 La coordination entre deux bras	21
1.2.2 La manipulation et l'assemblage avec deux bras	23
1.3 Dénudage de fil	25
1.4 Conclusion	28
<b>2 Modélisation des robots</b>	<b>29</b>
2.1 Introduction	29
2.2 Notions de base	29
2.2.1 Degrés de liberté d'un corps rigide	29
2.2.2 La redondance	30
2.2.3 Transformations Homogènes	30
2.2.4 Convention de Denavit-Hartenberg modifiée	31
2.3 Modélisation	33
2.3.1 SYMORO+	33
2.3.2 Description du robot YuMi IRB14000	34
2.3.3 Modélisation géométrique	35
2.3.4 Matrice de transformation de base du robot	38
2.3.5 Modèle cinématique	41
2.3.6 Modélisation dynamique	42
2.3.7 Modèle de frottement	44
2.4 Conclusion	46
<b>3 Identification dynamique du robot</b>	<b>47</b>
3.1 Introduction	47
3.2 Définition de la méthode d'identification	49
3.2.1 Identification par MDI, moindres carrés et optimisation	49
3.2.2 Précision des paramètres estimés	54
3.2.3 Trajectoires excitantes	55
3.3 Résultats expérimentaux d'identification	57
3.3.1 Mesure et filtrage des données	59
3.3.2 Résultats	60
3.4 Simulateur de robot	65
3.4.1 Structure du contrôleur de robot	65
3.5 Calcul des forces externes	68
3.6 Conclusion	76

<b>4</b>	<b>Communication externe du robot</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Définition de la méthode de communication . . . . .	79
4.2.1	Premier volet : Création d'une interface client/serveur . . . . .	80
4.2.2	Deuxième volet : Protocoles UDP et TCP . . . . .	80
4.2.3	Troisième volet : Python, une interface intermédiaire . . . . .	81
4.3	Résultats de communication . . . . .	82
4.4	Conclusion . . . . .	85
<b>5</b>	<b>Commande du robot YuMi</b>	<b>87</b>
5.1	État de l'art . . . . .	87
5.2	Définition de la loi de commande . . . . .	91
5.3	Simulation numérique de la commande . . . . .	92
5.3.1	Simulation Matlab/Simulink . . . . .	92
5.4	Résultats expérimentaux de la commande sur YuMi . . . . .	97
5.5	Application au dénudage . . . . .	101
5.5.1	Technique de dénudage adoptée . . . . .	101
5.5.2	Résultats expérimentaux . . . . .	103
5.6	Conclusion . . . . .	109
	<b>Conclusion générale et perspectives</b>	<b>111</b>
	<b>Appendices</b>	<b>115</b>
<b>A</b>	<b>Modèle géométrique direct</b>	<b>116</b>
<b>B</b>	<b>Modèle géométrique inverse</b>	<b>119</b>
<b>C</b>	<b>Matrices de transformation de base des deux bras</b>	<b>122</b>
<b>D</b>	<b>Jacobienne du robot YuMi</b>	<b>123</b>
<b>E</b>	<b>Filtrage</b>	<b>127</b>

# Table des figures

1	Les trois types de collaboration : a) parallèle, b) exosquelette et c) série . . . . .	11
1.1	L'adoption des robots industriels collaboratifs et traditionnels dans le monde [3]	16
1.2	À gauche : 1 <sup>er</sup> prototype de cobot créé par [37] en 1996. À droite : 1 <sup>er</sup> prototype industriel d'un cobot destiné à aider les employés à décharger les portières des voitures [146] . . . . .	17
1.3	Différents modèles (de divers fabricants et architectures) de robots collaboratifs sur le marché actuel . . . . .	18
1.4	Robots à deux bras, de gauche à droite, (a) PR2, (b) YuMi et (c) SDA10D . . .	20
1.5	(a) Manipulation d'un objet par deux bras. (b) Coordination pour une tâche d'assemblage . . . . .	21
1.6	Schéma de principe de solution hybride symétrique [45] . . . . .	23
1.7	Le concept hiérarchique de la manipulation à deux bras [132] . . . . .	24
1.8	Pince à dénuder classique (a), Pince à dénuder automatique (b), Dénudeur work-pro (c), Machine Schleuniger (d), Kit de dénudage DEWALT (e) . . . . .	27
2.1	Le robot YuMi en position de redondance pour éviter les obstacles . . . . .	30
2.2	Transformation des repères . . . . .	31
2.3	Convention de Denavit-Hartenberg modifiée . . . . .	32
2.4	Interface du logiciel SYMORO+ . . . . .	33
2.5	Robot YuMi IRB14000 au laboratoire LCFC . . . . .	34
2.6	Bras droite du YuMi et ses repères DHM. . . . .	36
2.7	Base du bras gauche et base du robot YuMi [RobotStudio] . . . . .	37
2.8	Simulateur du bras gauche du robot sous Matlab . . . . .	40
2.9	Simulateur du robot avec Peter Corke Toolbox sur Matlab . . . . .	40
2.10	Modèle de frottement de Coulomb et frottement visqueux . . . . .	45
3.1	Schéma du principe de la méthode d'identification . . . . .	49
3.2	Courants mesurés lors du suivi de la trajectoire excitante . . . . .	56
3.3	Comparaison des couples mesurés, filtrés et estimés pour les articulations 1, 2 et 3	61
3.4	Comparaison des couples mesurés, filtrés et estimés pour les articulations 4,5 et 6	62
3.5	Comparaison des couples mesurés, filtrés et estimés pour l'articulation 7 . . . . .	63
3.6	Évolution des valeurs propres de la matrice d'inertie sur l'ensemble de l'espace de travail du bras de YuMi . . . . .	64
3.7	Schéma du principe de commande bas-niveau du contrôleur IRC5 . . . . .	66
3.8	Schéma du principe de commande bas-niveau du contrôleur de robot sur Simulink	67
3.9	Écart de la position $q_1$ et de la vitesse $\dot{q}_1$ de l'axe 1 entre la valeur réelle et la valeur de référence . . . . .	67
3.10	Écart de la position $q_2$ et de la vitesse $\dot{q}_2$ de l'axe 2 entre la valeur réelle et la valeur de référence . . . . .	68
3.11	Couples externes mesurés et calculés Robot YuMi LCFC . . . . .	71

3.12	Couples externes mesurés et calculés Robot YuMi ant-Technologie . . . . .	72
3.13	Matériel utilisé pour évaluer les couples externes . . . . .	74
3.14	Force appliquée suivant $z$ sur le robot . . . . .	75
3.15	Force appliquée suivant $z$ sur le ressort . . . . .	75
4.1	Schéma du principe d'une communication client-serveur avec le protocole TCP .	80
4.2	Programmation des sockets en langage RAPID [124] . . . . .	81
4.3	LAN construit entre YuMi et l'ordinateur . . . . .	83
4.4	Surveillance du retard d'échange de données entre le robot et l'ordinateur . . . .	83
4.5	Exemple de démonstration du mouvement du bras gauche du robot YuMi, la séquence des mouvements se fait de gauche à droite, de haut en bas sur la figure	84
4.6	La trajectoire de l'effecteur dans l'espace tridimensionnel . . . . .	84
5.1	Principe de commande hybride position/force . . . . .	88
5.2	Mouvement d'un robot en contact avec l'environnement [20] . . . . .	90
5.3	Principe de commande proposée pour un bras de YuMi . . . . .	91
5.4	Schéma de commande proposée pour un bras de YuMi sur Simulink . . . . .	92
5.5	Positions désirées et réelles du simulateur sans bruit . . . . .	93
5.6	Forces et moments désirés et réels du simulateur avec bruit . . . . .	94
5.7	Positions désirées et réelles du simulateur avec bruit . . . . .	95
5.8	Forces et moments désirés et réels du simulateur avec bruit . . . . .	96
5.9	Couple d'axe 1 avec différents type de filtrage . . . . .	97
5.10	Position désirée et réelle du robot selon $z$ . . . . .	99
5.11	Force désirée et réelle du robot selon $z$ . . . . .	99
5.12	Position désirée et réelle du robot selon $y$ . . . . .	100
5.13	Force désirée et réelle du robot selon $y$ . . . . .	100
5.14	Modèle CAO proposé pour la pince avec des lames identiques en forme de V . .	101
5.15	Pince proposée montée sur le robot . . . . .	102
5.16	Résultats de dénudage avec force de serrage 20N et force contrôlée 30N . . . . .	104
5.17	Résultats de dénudage avec force de serrage 18N et force contrôlée 23N . . . . .	104
5.18	Résultats de dénudage avec force de serrage 15N et force contrôlée 35N . . . . .	105
5.19	Résultats de dénudage avec force de serrage 15N et force contrôlée 25N . . . . .	105
5.20	Résultats de dénudage avec force de serrage 8N et force contrôlée 18N . . . . .	106
5.21	Les étapes de dénudage avec YuMi . . . . .	107
5.22	Position du plan de coupure sur le fil . . . . .	108
E.1	Réponse en amplitude des filtres de Butterworth(différents ordre)[55] . . . . .	128
E.2	Réponse en amplitude des filtres Chebychev pour 0.5 dB d'ondulation [55] . . . .	129

# Liste des tableaux

2.1	Caractéristiques techniques de YuMi . . . . .	35
2.2	<i>Paramètres DHM du bras gauche de YuMi</i> . . . . .	35
2.3	<i>Paramètres DHM du base de robot</i> . . . . .	39
3.1	<i>Regroupement des paramètres du bras du robot YuMi</i> . . . . .	57
3.2	Résultats de l'identification des paramètres de frottement et des offsets . . . . .	63
3.3	Paramètres inertiels identifiés . . . . .	65
3.4	Valeurs numériques des facteurs des correcteurs . . . . .	66
4.1	La configuration appropriée des composants dans le réseau local . . . . .	82
E.1	Polynômes de Butterworth normalisés . . . . .	128
E.2	Polynômes normalisés de filtre Chebychev de type 1 pour $\varepsilon = 1dB$ . . . . .	129

# Acronymes

**MDI** Modèle Dynamique Inverse

**MDD** Modèle Dynamique Direct

**Tcp** Tool center point

**EAJs** Euler Angles Joints

**DH** Denavit-Hartenberg

**DHM** Denavit-Hartenberg Modifiée

**DDL** Degrés De Liberté

**FKE** Filtre de Kalman Etendu

**CAO** Conception Assistée par Ordinateur

**CLOE** Closed-Loop Output Error

**SDP** State-Dependent Parameter

**NRMSE** Normalized Root-Mean-Square Error

**DIDIM** Direct and Inverse Dynamic Identification Model

**IDIM-LS** Inverse Dynamic Identification Model with Least Squares

**SDP** State Dependent Parameter

**CTM** Centre De Masse

**NRMSE** Normalized Root Mean Square Error

**RAPID** Robot Application Programming Interface Dialog

**DSP** Digital signal processor

**TCP** Transmission Control Protocol

**UDP** User Datagram Protocol

**LAN** Local Area Network

**WAN** Wide Area Network

**DSP** Digital Signal Processor

**ROS** Robot Operating System

# Introduction Générale

Il n'y a pas si longtemps, le robot industriel a traversé d'importants développements. Pour définir un robot industriel, on peut dire qu'il s'agit d'une machine composée d'une partie mécanique avec un certain nombre de degrés de liberté pour accomplir certains mouvements. Cette machine est également composée de capteurs, dont une partie est utilisée par le robot pour connaître son propre état et assurer son fonctionnement, et l'autre partie pour connaître son environnement afin d'effectuer correctement une tâche. De plus, une partie avec de l'énergie permet d'alimenter les différents actionneurs et capteurs et une partie traitement du signal pour interpréter les informations provenant des capteurs et générer les différentes commandes. Les robots industriels ont principalement été utilisés pour effectuer des tâches fastidieuses et répétitives à grande vitesse et avec une grande efficacité ou pour fonctionner dans des environnements inaccessibles à l'homme.

En revanche, ces robots industriels traditionnels ont connu une révolution majeure qui les a rendus de plus en plus autonomes et faciles à commander : un robot industriel est aujourd'hui une machine capable de s'adapter et d'agir sur son environnement, même plus complexe, en remplaçant l'être humain, travaillant avec lui ou étendant certaines de ses fonctions. Un demi-siècle après leur apparition, ces robots ne sont plus confinés à des environnements très fermés, et comme le souligne parfaitement [97], " Robots have long left the cage of industrial settings : they work together with humans – collaboratively." c.-à-d. les robots ont depuis longtemps quitté le cadre de l'environnement industriel : ils travaillent en collaboration avec les êtres humains. En effet, les progrès technologiques rapides en matière d'électronique, d'informatique, de communication et de miniaturisation élargissent les applications concevables des robots. Des applications permettent au robot d'interagir avec les êtres humains en partageant l'espace de travail avec eux et d'effectuer des tâches en parallèle en toute sécurité : on parle alors de robots collaboratifs (cobots). En fonction du type de contact entre le robot et l'être humain, on peut distinguer trois catégories de collaboration : parallèle, exosquelette et série (voir Figure 1).

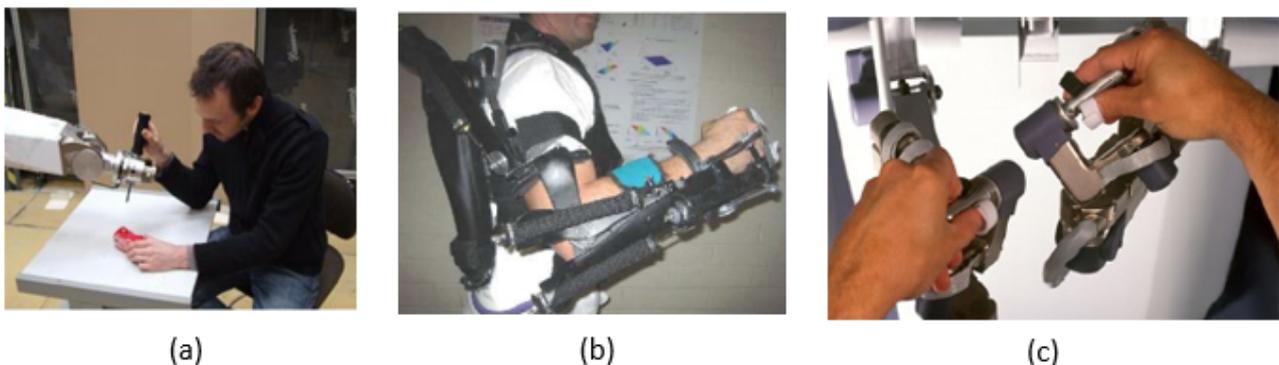


FIGURE 1 – Les trois types de collaboration : a) parallèle, b) exosquelette et c) série

- La première est la collaboration parallèle, dans laquelle le robot et l'opérateur sont tous deux en contact avec l'outil. Ils forment deux chaînes parallèles au niveau des articulations. Les forces appliquées à l'outil sont la somme des deux forces générées par le robot et l'opérateur.
- Dans la seconde catégorie, il s'agit d'un exosquelette ou d'une collaboration orthopédique ; dans ce cas, on peut imaginer que les forces sont réparties entre plusieurs points de contact avec le corps humain et ont donc une architecture propre à l'opérateur. Ils font l'objet de nombreuses recherches et ce type de robot est principalement utilisé en rééducation [30].
- Le troisième type est la collaboration série, caractérisée par une chaîne sérielle formée par le contact de l'opérateur avec le robot et le contact du robot avec l'outil. La position de l'outil résulte de la composition du mouvement du robot avec celui de l'opérateur. Ce type d'assistance se présente nécessairement sous la forme d'un dispositif portable pouvant être utilisé en chirurgie permettant comme exemple dans [106] de manipuler très précisément un outil médical en supprimant les tremblements de médecin dans une opération de la rétine ou encore par des simples utilisations comme l'usage de tournevis électrique.

Nous nous intéressons dans notre projet à la collaboration parallèle applicable aux robots industriels.

Dans le domaine de la robotique industrielle, outre l'introduction de robots collaboratifs, l'intérêt et le besoin de systèmes de type humanoïde, en particulier de robots à deux bras, ont également augmenté de manière significative ces dernières années et sont devenus un aspect important de la recherche scientifique. Cette thèse a été développée dans le cadre du projet Robotix Academy financé par le fonds européen de développement régional INTERREG V-A Grande Région [1]. L'objectif de la thèse est de mener des recherches dans le domaine de la robotique collaborative industrielle visant à utiliser le robot collaboratif à deux bras ABB IRB 14000 "YuMi" pour la manipulation et l'assemblage de pièces flexibles. Ces pièces flexibles sont des câbles et des fils. La manipulation de pièces flexibles a différentes applications, comme par exemple, souder des câbles à un connecteur, dénuder des fils ou des câbles avec une tresse et plusieurs fils gainés. Toutes ces tâches sont effectuées en tenant compte du fait qu'un opérateur humain est proche du robot, soit pour effectuer une tâche simultanément, soit pour superviser et qualifier le processus.

## Motivation et Problématique

Cette thèse s'inscrit dans le contexte de la robotique industrielle collaborative, un aspect de recherche très intéressant pour moi qui est passionnée par la robotique et en particulier la robotique proche de l'homme, mais qui a une contribution productive dans la vie. Ce fut ma première motivation pour me pousser dans le monde de la recherche, un monde où j'ai aussi découvert l'aspect de robots à deux bras pour pouvoir robotiser le dénudage de fil. Cette application devra donc être la première de ce type dans le monde industriel. La mise au point d'un système de dénudage robotisé est non seulement très utile pour l'industrie, mais elle représente également une avancée significative, dans la mesure où il s'agit de la première expérience de robotisation dans ce type d'application. Nous ne nions pas qu'il existe des appareils de dénudage automatique, mais cela n'empêche pas qu'une robotisation d'une telle application soit bénéfique au domaine de la manipulation de petits objets, notamment dans l'industrie des composants électroniques. Pour ce faire, il faut toutefois connaître le robot dans son intégralité. Or, ce n'est pas le cas dans la mesure où nous avons affaire à un robot à architecture fermée, un défi à

relever pour parvenir à lui appliquer des lois de commande externes.

L'objectif principal de cette thèse est de piloter les deux bras anthropomorphes de YuMi afin d'accomplir les tâches de manipulation et donc de gérer les aspects suivants :

- La modélisation géométrique, cinématique et dynamique des bras du robot YuMi .
- L'identification des paramètres dynamiques du robot, y compris la matrice d'inertie, les coefficients de frottement, les offsets, etc
- Le développement d'un simulateur Matlab/Simulink du robot, pour éviter les dommages au robot (collision, contact avec l'environnement) et pour valider les approches proposées
- Le développement d'une communication externe avec le robot via le protocole de communication TCP/IP pour permettre à appliquer les lois de commande en force
- Le développement d'une loi de contrôle externe sur le robot pour contrôler les forces externes appliquées au robot
- La mise à l'essai et la validation expérimentalement de la loi de contrôle par le dénudage des fils.

## Organisation de la thèse

Afin de clarifier le sujet de recherche, de présenter le travail effectué au cours de ce projet et de discuter des perspectives à suivre, nous avons organisé ce mémoire de thèse en cinq chapitres :

- **Chapitre 1** : Dans cette première partie, nous nous sommes consacrés à un état de l'art sur les différents aspects de recherche auxquels nous avons été confrontés, en particulier, les robots industriels collaboratifs, la collaboration entre les deux bras du robot ainsi que la collaboration Homme-Robot. Enfin, nous avons abordé l'application du dénudage et les outils destinés au dénudage que nous souhaitons robotiser et ce grâce à une loi de commande externe développée sur le robot YuMi.
- **Chapitre 2** : Ce chapitre porte sur la modélisation des robots et plus précisément sur la modélisation de notre robot YuMi. Quelques notions de base de la robotique en général et des robots manipulateurs en particulier sont présentées, comme les degrés de liberté, la redondance, etc. Une présentation du robot YuMi IRB14000 adopté dans le projet est effectuée, pour lequel une modélisation géométrique, cinématique et dynamique est définie, calculée et validée au regard des différents travaux de recherche traitant de manière très pertinente la modélisation des robots.
- **Chapitre 3** : Lors de ce chapitre, nous abordons la méthode d'identification des paramètres dynamiques du robot, dans la perspective de pouvoir simuler le comportement réel du robot et, notamment, de pouvoir déterminer les forces externes appliquées au robot afin d'effectuer le contrôle force/position. Dans cette optique, nous décrivons d'abord toutes les étapes envisagées par la méthode d'identification développée. Par la suite, nous présentons les résultats expérimentaux de cette méthode. La détermination des forces externes appliquées au robot est également traitée dans ce chapitre.
- **Chapitre 4** : Pour commander un robot industriel de manière externe, il faut pouvoir communiquer avec le robot en temps réel. De plus, bien que la simulation ait été validée, elle n'est pas suffisante pour assurer le bon fonctionnement de celle-ci sur le robot.

Nous présentons donc dans ce chapitre le protocole de communication réalisé sur le robot pour implémenter une nouvelle loi de commande. Les résultats expérimentaux de la communication sont également présentés.

- **Chapitre 5** : Ce chapitre cinq est dédié à la présentation de la loi de commande en position de la force externe développée sur le robot. Les résultats de la simulation dans Matlab/Simulink ainsi que les résultats expérimentaux sur le robot de cette commande sur différents exemples sont exposés, et pour finir les premiers résultats de dénudage de fils électriques avec la commande sont par ailleurs évoqués.

La conclusion générale et les perspectives sont présentées à la fin du manuscrit.

# Chapitre 1

## État de l'art

L'objectif principal de ce chapitre est d'élaborer le contexte scientifique et de présenter de manière synthétique au lecteur un état de l'art sur les domaines connexes à cette thèse. Le premier paragraphe évoque les travaux de recherche sur les robots collaboratifs industriels et leur évolution. Le second paragraphe aborde brièvement le principe des robots multi-bras et l'intérêt des robots collaboratifs, mais cette fois-ci avec des robots à deux bras dans le domaine de la co-manipulation Robot-Robot et Robot-Homme, leur domaine d'application, les défis rencontrés par la robotique collaborative et enfin une revue de la littérature spécifique à ce type des robots. Enfin, le troisième paragraphe décrit l'application du dénudage de fils et les outils existants pour réaliser cette opération dans l'industrie.

### 1.1 Les robots collaboratifs industriels

Pour comprendre le concept et la notion de robot collaboratif industriel, nous devons d'abord évoquer brièvement l'évolution des robots depuis leur apparition jusqu'à aujourd'hui. Un robot industriel est une machine capable d'effectuer des tâches lorsque son programme est mis en place. Son fonctionnement est généralement assuré par trois entités : une partie mécanique, une partie électronique, composée de capteurs dans le système de contrôle, ainsi qu'une partie informatique qui le relie à son utilisateur et à son environnement. Ce type de robots est utilisé pour effectuer de nombreuses tâches dans le secteur industriel : assemblage, emballage, inspections dans des zones sensibles, découpe laser, contrôle de pièces ; et surtout, ils permettent d'effectuer des tâches qui seraient très pénibles et risquées pour l'homme. En règle générale, ces derniers sont spécifiques à une application, excluent du champ de travail de l'homme et disposent de leur propre espace de travail.

Par ailleurs, toujours plus nombreux, les robots collaboratifs ou " cobots " font leur entrée dans la production industrielle. Conformément à [22], ces cobots ont le potentiel de transformer radicalement le marché et de devenir la technologie robotique dominante dans les prochaines décennies, sachant que les robots collaboratifs sont devenus l'un des segments du marché de la robotique qui connaît la croissance la plus rapide [67]. Cette évolution, quant à elle, a mis de plus en plus de pression sur les constructeurs pour qu'ils fabriquent des produits de haute qualité et réduisent leurs coûts dans un marché de plus en plus concurrentiel. En résumé, les fabricants du monde entier subissent une pression croissante dans ce domaine et, pour y remédier, on observe une explosion des idées innovantes. Ces innovations comprennent l'adoption de nouvelles technologies robotiques qui apportent une plus grande flexibilité aux processus de fabrication, dont notamment l'émergence de robots collaboratifs (voir Figure 1.1).

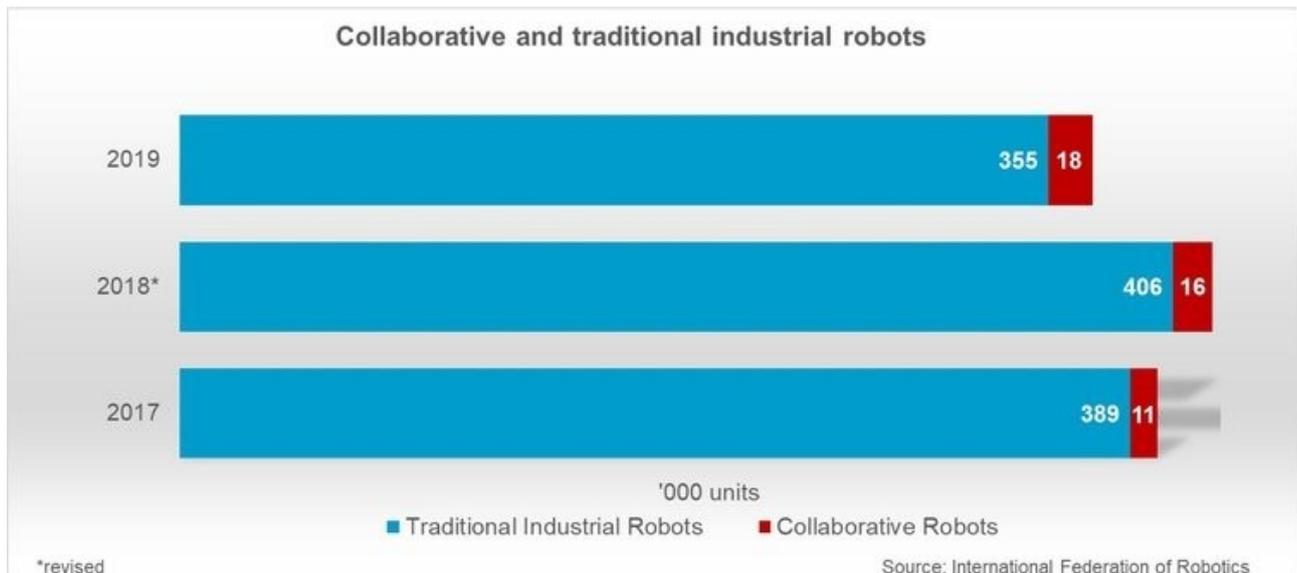


FIGURE 1.1 – L'adoption des robots industriels collaboratifs et traditionnels dans le monde [3]

Sur la dernière décennie, la croissance des robots, y compris les cobots, a été considérable [52] [73]. À noter que, selon le rapport global sur la robotique 2020 "Global Robotics Report 2020", publié par la Fédération Internationale de la Robotique (IFR) [3], 2,7 millions de robots industriels sont utilisés dans les usines du monde entier, soit une augmentation record de 12%. Par ailleurs, en France, on compte un total de 42000 robots dans les industries, soit une augmentation de 10% par rapport à l'année précédente, un chiffre qui continue de croître. Par la même occasion, le taux d'adoption des robots collaboratifs a augmenté de 11% en 2019. Une dynamique commerciale qui contraste avec la tendance générale des robots industriels traditionnels. En effet, plus les fournisseurs proposent des robots collaboratifs et une gamme d'applications plus large, plus le pourcentage de marché augmente et atteint 4,8% en 2019 de la base totale installée de 373000 robots industriels. En [127], un aperçu des tendances actuelles et futures du développement des cobots est présenté, en mettant l'accent sur le rôle des robots collaboratifs dans le concept d'Industrie 4.0 et les évolutions mondiales.

Du côté de la recherche, le concept de robots collaboratifs n'est pas tout à fait récent, mais remonte à la recherche décrite dans [37] où le terme cobot, issu de l'expression robots et collaboration, était utilisé en 1996. Une initiative avait alors été lancée pour spécifier un dispositif passif doté de commandes capables de contraindre et de guider le mouvement d'une personne, en définissant des surfaces virtuelles ; l'objectif était de trouver un moyen de rendre les robots ou les dispositifs semblables à des robots suffisamment sûrs pour cohabiter avec des êtres humains. En 1998, un premier prototype de robot industriel a été développé dans [146] pour aider les opérateurs à décharger les portières de voitures (Figure 1.2). Cette nouvelle forme d'assistance a permis, selon [13], d'améliorer la santé, la sécurité, la productivité et la qualité.

Depuis, le concept est resté relativement inexploré jusqu'en 2008, date à laquelle une nouvelle étape dans le déploiement de l'interaction homme-robot dans l'industrie a été marquée avec la commercialisation du premier robot collaboratif, l'UR5, par la société Universal Robots. Celui-ci a été conçu pour être capable d'interagir physiquement avec un être humain en toute sécurité ; grâce à la mesure du courant moteur au niveau des articulations, il est possible d'évaluer l'efficacité de l'interaction et les forces exercées sur le robot peuvent être définies et réduites.



FIGURE 1.2 – À gauche : 1<sup>er</sup> prototype de cobot créé par [37] en 1996. À droite : 1<sup>er</sup> prototype industriel d'un cobot destiné à aider les employés à décharger les portières des voitures [146]

Ce genre de robots collaboratifs diffère des robots industriels traditionnels, qui agissent séparément de leurs collaborateurs humains. Dans le cadre de cette approche, les robots sont conçus pour interagir directement avec les êtres humains, manipuler des tâches utiles partagées et fonctionner en toute sécurité sans les cages de sécurité traditionnelles ou autres dispositifs de protection similaires [113] et [25]. Ils sont également connus sous le nom de "robots coopératifs". Ces robots sont beaucoup plus légers et donc très mobiles par rapport aux robots industriels encombrants, ce qui facilite leur déplacement dans les industries visées. De plus, leur flexibilité est un autre avantage par rapport aux robots industriels, car ils sont très flexibles et peuvent être affectés à une multitude de tâches.

Toutes ces raisons ont incité les principaux fabricants de robots à développer leurs séries de robots collaboratifs afin de répondre à la croissance enregistrée sur le marché, tels que le YuMi d'ABB, le CR-35iA de Fanuc, la série TX2 de Stäubli ou encore la série UR d'Universal Robots. La figure 1.3 illustre certains des robots collaboratifs présents sur le marché. Bien que leurs architectures diffèrent, tous ces robots ont la capacité de travailler avec des êtres humains sans avoir besoin de barrières de protection supplémentaires. En fait, la plupart d'entre eux sont conçus pour parvenir à un mode de fonctionnement collaboratif en limitant la puissance et la force par conception ou par commande.



FIGURE 1.3 – Différents modèles (de divers fabricants et architectures) de robots collaboratifs sur le marché actuel

Cette collaboration tire parti de la dextérité et des capacités de décision des êtres humains avec la force et l'endurance des robots [48]. La plupart des robots sont légers et ont la capacité de se déplacer d'un endroit à l'autre, pour ainsi dire : ils sont mobiles, du moins dans un sens passif [18]. Ils s'appuient sur des stratégies technologiques spécifiques qui diffèrent des robots industriels classiques. Par exemple, le LBR iiwa de KUKA utilise des capteurs de couple-moteur dans chaque articulation pour mesurer les forces exercées par le robot sur son environnement extérieur. D'autres fabricants utilisent des mousses pour rendre les robots conventionnels "collaboratifs". D'autres utilisent une combinaison de mousse de protection et de capteurs de proximité et sensoriels pour transformer leur gamme de robots conventionnels en "robots collaboratifs". Tout cela fait de la robotique une bonne solution utilisée pour accéder à l'automatisation des usines [103] et une technologie propice à ce que l'on appelle la quatrième révolution industrielle, où la flexibilité est un élément clé pour compléter l'automatisation.

La différence entre les robots collaboratifs industriels et les robots industriels traditionnels réside dans leur interaction directe avec l'Homme. Grâce à cette interaction, il est possible pour les employeurs, du moins en théorie, de tirer parti des forces et de la robustesse des robots, ainsi que des connaissances tacites et des compétences décisionnelles agiles des êtres humains. De cette façon, les entreprises peuvent bénéficier des avantages clés que possèdent à la fois les opérateurs humains et les robots [52]. Les robots peuvent aspirer les tâches répétitives et monotones, mais les travailleurs humains sont toujours mieux à même de gérer les tâches non planifiées et inattendues que leurs correspondants automatisés. En un sens, les humains restent la ressource la plus flexible du système. En tirant parti de cette diversité, la collaboration

Homme-Robot peut être supérieure aux processus uniquement robotiques. Ce principe s'est également avéré vrai dans des contextes de recherche expérimentale [22].

Selon [49], les modèles d'interaction entre les opérateurs humains et les robots peuvent être définis sous forme de scénarios collaboratifs. Dans chaque scénario, au moins un opérateur humain et au moins un robot partagent le même espace de travail pour effectuer un processus de fabrication sur une (plusieurs) pièce(s). Dans la littérature, il existe plusieurs définitions qui distinguent la collaboration de la coopération ou de l'interaction. En effet, la référence [18] indique par exemple que les robots coopératifs travaillent avec un humain pour atteindre un objectif commun, étape par étape, tandis que les robots collaboratifs travaillent main dans la main avec un humain pour accomplir une tâche commune. Conformément à la définition générale du constructeur de cobots, tout robot qui collabore avec des humains peut être qualifié de robot collaboratif, tout en définissant quatre scénarios de collaboration différents [49] :

- **Indépendant** : un opérateur humain et un cobot agissent indépendamment sur différentes pièces selon leurs processus de production respectifs. Cette collaboration repose sur un espace de travail partagé, dépourvu de barrières ou de cages.
- **Simultané** : un opérateur humain et un cobot travaillent sur des processus séparés sur la même pièce en même temps. Il n'y a pas de dépendance temporelle ou de tâche entre eux. Toutefois, le cobot devra tenir compte des exigences de l'opérateur en matière d'espace et de tâches pour respecter l'espace de l'opérateur. Travailler simultanément sur la même pièce minimise le temps de transfert de la pièce entre le cobot et l'humain, ce qui augmente la productivité et l'utilisation de l'espace.
- **Séquentiel** : un opérateur humain et un cobot effectuent des processus de fabrication séquentiels sur la même pièce. En effet, dans ce cas, il existe une dépendance temporelle entre l'opérateur et le processus cobot. En règle générale, le cobot est chargé de gérer les processus fastidieux afin d'améliorer les conditions de travail de l'opérateur.
- **De support** : un opérateur humain et un cobot effectuent le même processus sur la même pièce de manière interactive. La dépendance entre l'homme et le cobot est alors totale, l'un ne peut accomplir la tâche sans l'autre.

En revanche, les robots collaboratifs sont actuellement rarement utilisés en collaboration avec des humains. Cela provient du fait que les industriels les apprécient principalement pour leur coût d'intégration nettement inférieur à celui des manipulateurs traditionnels et pour leur faible encombrement. En effet, les robots collaboratifs intègrent des dispositifs de sécurité pour l'interaction qui réduisent la taille de la zone de sécurité requise. En outre, leur programmation par guidage sur des tâches simples réduit le temps et les coûts de mise en place. Selon [49], la majorité des exemples de robots collaboratifs déployés dans le monde industriel sont désormais liés à des scénarios de collaboration "Indépendants" ou "Simultanés". De leur côté, les travaux de recherche visant des objectifs d'innovation portent plutôt sur le développement de scénarios de collaboration "Séquentiels" ou "De support", qui nécessitent des systèmes et des solutions plus complexes. En effet, à mesure que le niveau d'interdépendance et de collaboration augmente, les robots doivent avoir une meilleure compréhension sémantique des objectifs des tâches et des actions ainsi que des intentions de leurs homologues humains, mais également, les humains doivent pouvoir communiquer avec le robot de manière plus intuitive.

## 1.2 Robot collaboratif à deux bras

La collaboration ou "co-manipulation" consiste, en robotique, à réaliser une tâche donnée simultanément par un robot et un humain (co-manipulation homme-robot) ou par deux robots (co-manipulation robot-robot) ou encore entre deux ou plusieurs bras du même robot. Cette co-manipulation nécessite le partage de l'espace de travail, mais aussi du temps de l'intervention. La co-manipulation homme-robot vise à utiliser les capacités du robot (force, précision, ...) et la capacité de décision de l'homme pour effectuer des tâches, alors que pour la co-manipulation entre deux ou plusieurs bras vise à imiter le comportement humain pour effectuer des tâches difficiles et précises selon la nature de la tâche.

En effet, pour la plupart des gens, un robot de rêve doit ressembler à un être humain. Il a des bras comme les humains, il est sûr, intelligent et émotionnel, il peut utiliser avec souplesse les outils humains comme les humains et il s'intègre facilement dans la société humaine. Comme un début d'un rêve, les robots à deux bras peuvent devenir le prochain centre de recherche sur les robots. Il n'y a pas si longtemps, la plupart des robots industriels que nous voyons étaient des manipulateurs à un bras, et même lorsque les robots de service ont deux bras, la plupart d'entre eux jouent et dansent et n'ont pas de réelle productivité. Dans le contexte actuel, de nombreuses entreprises de recherche et de développement et des fabricants ont successivement lancé des robots à deux bras, tels que YuMi d'ABB, Motoman/SDA de Yaskawa, Baxter, PR2 de Rethink, etc., et ont déployé de nombreux efforts et tentatives d'applications pratiques (voir Figure 1.4).

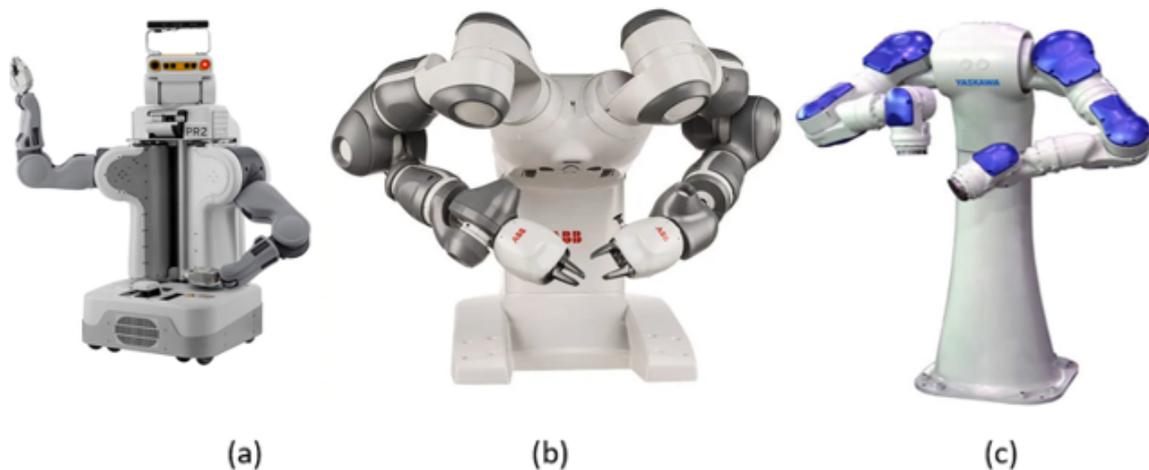


FIGURE 1.4 – Robots à deux bras, de gauche à droite, (a) PR2, (b) YuMi et (c) SDA10D

Le fonctionnement collaboratif de plusieurs robots est une exigence inévitable pour le développement de l'industrie. Le robot à deux bras est un nouveau type de robot développé pour répondre à cette exigence. Par rapport au robot à un bras, un robot à deux bras peut également améliorer considérablement l'adaptabilité du robot aux tâches d'assemblage complexes et améliorer l'efficacité de l'utilisation de l'espace de travail. Néanmoins, ces avantages impliquent l'inconvénient de disposer d'un système plus complexe. En effet, les manipulateurs multiples provoquent des contraintes internes dans l'objet manipulé, et un système de contrôle des forces doit généralement intervenir pour minimiser ces forces. Dans cette partie, nous présentons les travaux de recherche déjà effectués sur les robots multi-bras et plus particulièrement ceux qui

peuvent seulement faire des mouvements et des danses, mais qui seraient destinés à un réel service de production et de coordination entre les deux bras.

### 1.2.1 La coordination entre deux bras

Cela fait environ 40 ans que le besoin d'un robot à deux bras capable de travailler en coordination pour manipuler (Figure (a) 1.5), assembler des pièces (Figure (b) 1.5), et réaliser des compétences humaines souvent effectuées avec deux bras, a été décrit [80]. Zollner dans [153] distingue deux types de tâches coordonnées de manipulation. Le premier type est représenté par la manipulation d'un seul objet par plusieurs bras appelé coordination symétrique (Co-manipulation). Le deuxième regroupe les tâches d'assemblage où les pièces sont positionnées chacune par un bras manipulateur. Les questions restant à poser sont donc :

Comment peut-on définir cette coordination ?

Quels sont les moyens de sa mise en œuvre sur un robot à deux bras ?

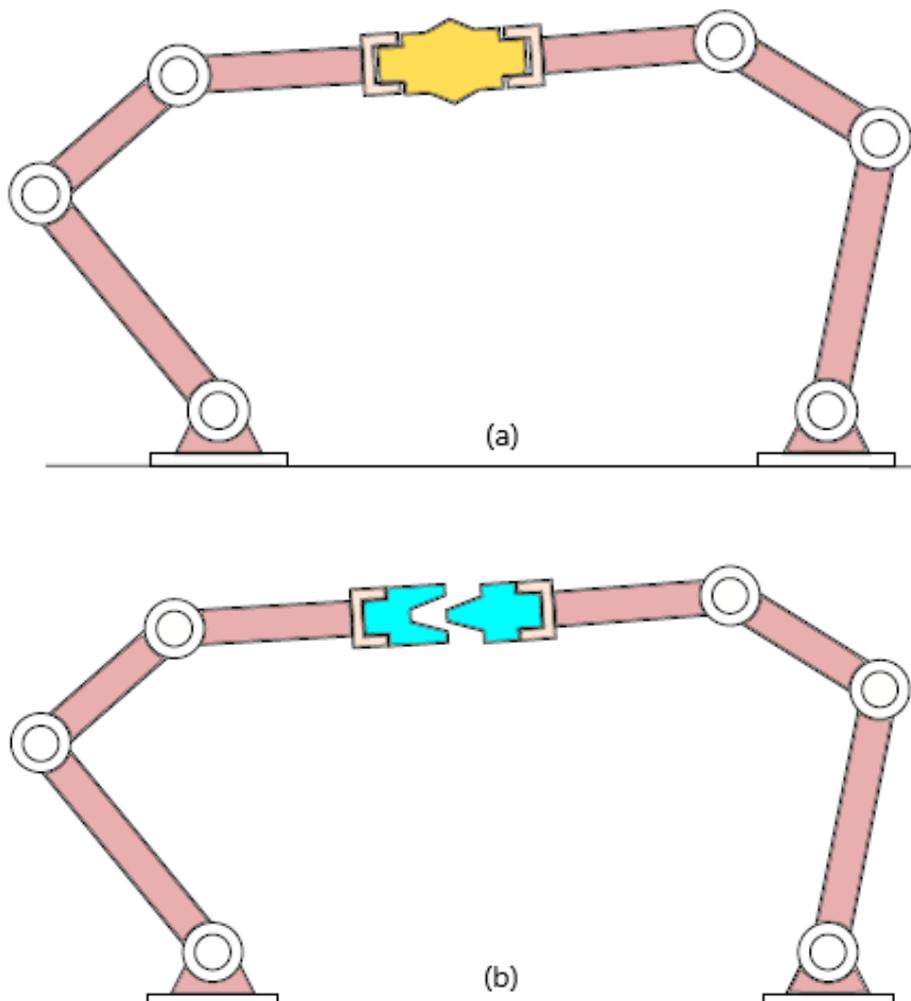


FIGURE 1.5 – (a) Manipulation d'un objet par deux bras. (b) Coordination pour une tâche d'assemblage

D'après [45] l'approche Maître-Esclave était la première solution développée en 1977 utilisant deux robots différents pour décrire les tâches possibles afin de travailler en coordination. La coordination est la prise en compte de contraintes (effort, position relative ...) entre deux organes terminaux des bras.

Le principe de la solution était d'une part, commander le premier robot en position ; il calcule la trajectoire de l'organe terminal à partir de la trajectoire désirée de l'objet en se basant sur des informations proprioceptives de position. Par ailleurs, le deuxième robot était commandé en effort. Son rôle était d'exercer des efforts au niveau de l'organe terminal pour contraindre l'objet dans certaines directions et pour compenser une partie du poids de l'objet.

Quelques années plus tard, apparaît la solution Maître-Esclave généralisée, cette fois-ci chaque bras a des capteurs internes pour faire l'asservissement de position dans l'espace opérationnel et des capteurs externes pour faire l'asservissement en effort. Donc, pour décrire une tâche pour chaque bras (robot) on utilise deux consignes, une consigne de position et une autre consigne de force. La solution Maître-Esclave généralisée permet d'avoir plus de choix sur les directions commandées et permet également de manipuler un objet flexible. Contrairement à la solution Maître-Esclave, le moindre déplacement du bras maître est ressenti par le capteur d'efforts et entraîne un couplage de commande.

Cette solution classique reste une approche adoptée jusqu'à aujourd'hui. En 2017, pour l'exploitation et la maintenance d'une station spatiale chinoise les principes de conception du robot ont été proposés sur la base de l'analyse des missions du robot astronaute à deux bras [78]. Le modèle dynamique du robot astronaute est établi sur la base du problème de contrôle de la coordination des forces dans le cas d'un fonctionnement à deux bras. À partir de la décomposition de la pose et de la force, la méthode de contrôle de la force Maître-Esclave est adoptée pour contrôler la force coordonnée des deux bras.

Fin des années quatre-vingt, la solution hybride symétrique proposée par Uchiyama a été mise en place (Figure 1.6), pour cette fois-ci considérer les deux robots comme un seul et unique robot avec un seul système de commande [141]. Cette approche consiste à faire suivre une trajectoire à l'objet en contrôlant les contraintes internes et externes (d'environnement) :

- Position absolue de l'objet : Déplacement dans l'espace.
- Position relative des préhenseurs : Déformation de l'objet.
- Effort externe : Efforts de contact (phase d'assemblage...).
- Effort interne pour exercer des contraintes mécaniques en cas d'objet rigide et pour déformation de l'objet et le maintien de cette déformation durant le déplacement pour un objet flexible.

Depuis, les chercheurs ont abordé plusieurs aspects du système multi-robots, cependant, la manipulation d'objets déformables n'a reçu d'attention qu'au début des années 90 avec [42] et [45]. Le but était de présenter des solutions pour manipuler des objets avec deux bras robotisés pour des applications de transport, l'assemblage et la déformation d'objets flexibles [149].

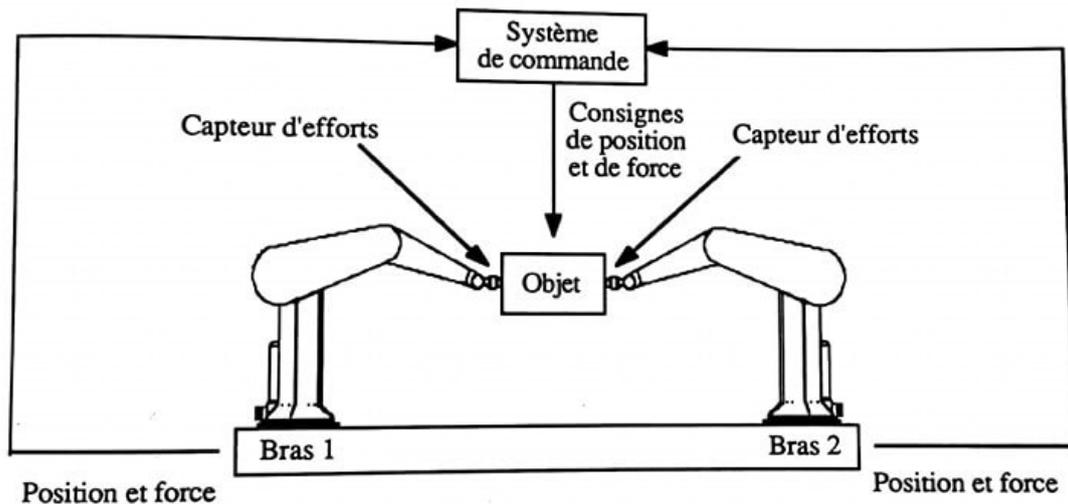


FIGURE 1.6 – Schéma de principe de solution hybride symétrique [45]

La référence [152] a abordé la problématique de planification des trajectoires de deux robots pour manipuler des objets flexibles en se basant sur deux méthodes de coordination différentes entre les deux bras. La première méthode permet de modifier seulement la position relative des deux manipulateurs. La deuxième méthode permet de modifier à la fois la position et l'orientation des bras. Une étude comparative entre une approximation linéaire par morceaux et une approximation par morceaux adaptative a été faite pour générer des trajectoires optimales sur le même intérêt de manipulation d'objet flexible [84].

En 2008, une approche générale sur la planification de trajectoire pour systèmes multi-bras basée sur une extension de l'algorithme PRM (Probabilistic Roadmap Method) pour les chaînes cinématiques fermées, a été présenté [66]. Dans [36] est abordée la planification de trajectoire pour un robot à deux bras dans des environnements contraints d'orientation verticale. Une nouvelle représentation de la manipulation à deux bras est proposée en 2010 en utilisant l'algèbre des quaternions doubles afin d'unifier les approches existantes dans la littérature [11], à savoir la méthode d'objet augmenté et lien virtuel développée par Khatib [95], le système de contrôle symétrique développé par Uchiyama [141], l'espace coopératif des tâches développé par Chiacchio [33] ou encore la commande synchronisée développée Sun [131]. Cette nouvelle méthode a été élargie pour prendre en compte toutes les chaînes cinématiques couplées, y incluant une simulation d'un manipulateur mobile. Il en résulte une représentation compacte et sans singularité pour les systèmes à deux bras, dénommée l'espace dual des tâches de coopération.

### 1.2.2 La manipulation et l'assemblage avec deux bras

Les robots industriels à deux bras semblent être une tendance prometteuse de la robotisation industrielle grâce à un éventail de propriétés [147] :

- une capacité à manipuler de manière plus intuitive est offerte aux opérateurs de robots à deux bras
- une cinématique qui permet d'ajuster la rigidité lorsque les deux bras sont en contact avec un objet commun

- une redondance importante qui améliore la flexibilité et la polyvalence de la manipulation des tâches.
- une manipulation synchronisée qui réduit le temps d'exécution des tâches.

À la différence des robots à un seul bras, les robots à deux bras ont toujours fait preuve de flexibilité ; une flexibilité qui se retrouve dans le fait que ces types de robots peuvent être utilisés pour différentes tâches de différentes manières en fonction de l'opération souhaitée. Par exemple, dans [150], une tâche d'assemblage de boîte de vitesses est démontrée avec un robot à deux bras en utilisant une méthode préprogrammée. Dans [138], la caractéristique anthropomorphique des robots à deux bras est exprimée pour mettre en œuvre des tâches d'assemblage en utilisant le modèle d'apprentissage à partir d'observations (LFO) ; une méthode qui propose de reconnaître ces tâches comme une séquence de primitives de mouvement à partir de données bruitées obtenues par un système de suivi d'objets. La vision a également été adoptée par [64] et [51] pour programmer des tâches d'assemblage de petites pièces réalisées par des robots à deux bras. Un robot à deux bras ou plusieurs robots à un seul bras peuvent également être organisés pour effectuer des tâches plus complexes et plus difficiles susceptibles de surpasser la capacité d'un seul robot. La problématique réside dans la méthode de définition de ses tâches.

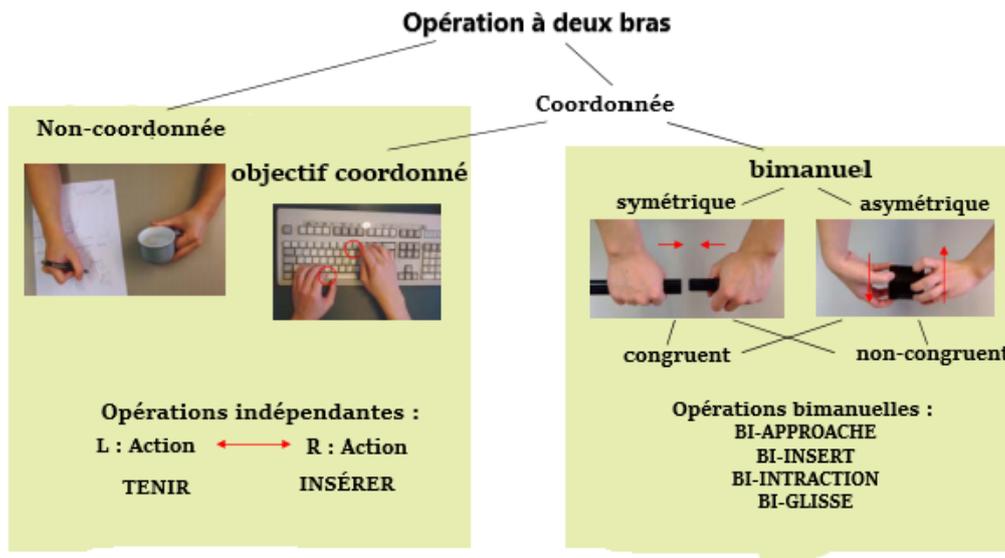


FIGURE 1.7 – Le concept hiérarchique de la manipulation à deux bras [132]

Pour répondre à cette question, [132] fournit une définition hiérarchique de la manipulation à deux bras ( voir Figure 1.7) dont la manipulation à deux bras se classe en manipulation non coordonnée et en manipulation coordonnée. On appelle manipulation non coordonnée la situation dans laquelle un mouvement indépendant est attribué à chaque bras, comme dans le cas d'un être humain tenant une tasse de café avec son bras gauche tout en faisant un dessin avec son bras droit. Quant à la manipulation coordonnée, elle se subdivise en deux catégories : la manipulation coordonnée par but et la manipulation bi-manuelle. La manipulation coordonnée fait référence à la situation où deux bras partagent la même tâche, mais les deux bras sont affectés à des mouvements indépendants, comme par exemple un humain qui tape sur un clavier avec ses mains gauche et droite. Le concept de manipulation bi-manuelle désigne la situation dans laquelle deux bras agissent simultanément sur un même objet de forme symétrique ou asymétrique, ou de forme congruente ou non congruente. En particulier, la manipulation coordonnée avec objectif est jugée la plus difficile et la plus complexe, car elle peut nécessiter

une longue pratique pour un humain, alors que la manipulation bi-manuelle est relativement intuitive pour l'humain.

Selon [147], la manipulation à deux bras est focalisée sur sa particularité d'anthropomorphisme<sup>1</sup>. Une grande partie de la planification de la manipulation des robots à deux bras peut faire référence à des mouvements effectués par des humains. Grâce à des taxonomies<sup>2</sup> et des représentations claires des compétences d'assemblage robotique, une tâche d'assemblage peut être décomposée en une série d'actions. Huckaby propose dans [79] une taxonomie d'assemblage conçue pour représenter la décomposition de tâches d'assemblage complexes de haut niveau en compétences simples et en primitives de compétences que le robot doit utiliser dans une séquence spécifiée.

### 1.3 Dénudage de fil

Le dénudage des câbles électriques est un processus très répandu dans le monde et son importance s'étend à de nombreux domaines. On l'utilise par exemple dans le monde industriel, à domicile pour le raccordement des installations électriques, et dans le sertissage pour la préparation des câbles. Le concept de dénudage est similaire à celui du dégainage dans certains secteurs dans lesquels les câbles sont dénudés pour récupérer les conducteurs en cuivre. Le dégainage peut également être considéré comme le dénudage des câbles dont la section transversale est supérieure à 16 mm<sup>2</sup> [4]. Cette technique est mise en place lors de la deuxième révolution industrielle réalisée à partir de 1870. En effet, le recours à l'électricité dans la production industrielle fera en sorte de précipiter les hommes à inventer des machines électriques [122]. Ainsi, désormais, pour réaliser les connexions entre les machines, les opérateurs auraient besoin de dénuder les câbles électriques.

Dénuder un câble électrique revient à retirer la couche protectrice appelée " isolant " de la couche conductrice sur une certaine longueur de câble. En outre, cela consiste à mettre à découvert l'âme du câble. Au fil des époques et avec un marché en pleine croissance, la demande de techniques de dénudage innovantes n'a cessé d'augmenter [5]. De nombreuses entreprises se sont spécialisées dans ce procédé en proposant des techniques toujours plus innovantes. Un grand nombre de brevets le justifie. De nos jours, les outils et les machines spéciaux créés grâce aux différentes révolutions industrielles sont très nombreux.

#### En quoi ressemble un câble ou un fil électrique ?

Un câble ou fil électrique est composé d'une gaine de protection appelée isolant et d'une partie centrale ou âme conductrice qui est généralement en cuivre grâce à son excellente conductivité, mais aussi en aluminium, en argent et même en or dans certaines applications industrielles. Un câble électrique est constitué d'un ensemble de 2 à 5 fils électriques, rassemblés dans une simple ou double gaine de protection [6]. On distingue principalement deux types de fils électriques :

- Fil flexible ou multibrin : flexible et constitué d'une tresse de plusieurs fils conducteurs. Il sert à réaliser la majorité des interconnexions entre les équipements électriques.
- Fil rigide ou massif : constitué d'un seul fil de conducteurs et généralement utilisé pour le transport de fortes intensités de courant électrique.

---

1. *Attribution de caractéristiques humaines à un non-humain*

2. *Étude des principes fondamentaux des classifications scientifiques*

## Quelles sont les propriétés du procédé de dénudage ?

Les techniques et outils de dénudage n'ont cessé de se créer et d'évoluer au fil des années, au point qu'il en existe aujourd'hui un grand nombre. Pour dénuder un fil électrique, il faut tenir compte d'un certain nombre de paramètres pour que le dénudage se fasse correctement. Conformément à la norme *NF EN 2812* [2] le dénudage est correctement réalisé lorsque :

- L'isolant ou le blindage ne présente aucune trace de détérioration qui pourrait être la cause le cas échéant de courts-circuits dans les installations électriques
- L'âme conductrice ne présente pas de fissures ou d'altérations qui pourraient entraîner la rupture du fil et provoquer un choc électrique
- Les performances d'origine du fil ou du câble définies dans les spécifications techniques ne sont pas réduites après l'opération de dénudage.

En outre, cette norme identifie également les défauts qui peuvent apparaître après l'opération de dénudage [130] [2]. Ces anomalies sont notamment :

- Une absence de torsion et de mise au rebut de l'âme
- Marques sur l'isolant (entaille, brûlure, fil visible, etc.)
- Résidu de l'isolant
- Fils ou torons coupés
- Contamination de surface (isolant brûlé, résidus de colle, poussière, etc.)

## Concepts et techniques de dénudage

Pour dénuder un câble électrique, on distingue plusieurs techniques. Quatre principales méthodes de dénudage sont présentées dans la norme *NF EN 2812*. Elles sont basées sur des outils mécaniques réalisés selon des technologies différentes et variées. À cet effet, il existe :

- Dénudation manuelle : cette pratique s'effectue à l'aide d'un scalpel, d'une lame de rasoir ou de tout autre objet tranchant (pinces coupantes, ciseaux, etc.). Il est le plus souvent utilisé lorsque les utilisateurs ne disposent pas des outils adéquats pour réaliser le dénudage. Cela entraîne dans la grande majorité des cas de nombreux défauts de dénudage et pour des applications telles que l'aéronautique, le recours au dénudage manuel des câbles est interdit lorsqu'un autre procédé de dénudage peut être utilisé.
- Dénudation mécanique : cette méthode de dénudage est la plus courante et la plus utilisée par les professionnels pour dénuder des fils individuels. Elle comprend de nombreux outils de différentes puissances adaptés à des sections de câbles données. La particularité de ces outils est leur simplicité, ce qui les rend idéaux pour une utilisation dans une large gamme d'applications.
- Dénudation thermique : il englobe tous les appareils alimentés par une tension et composés de deux parties : un générateur de tension pour contrôler la température et une partie manuelle constituée de deux fils avec des mâchoires à leur extrémité. Les deux fils sont alimentés par un courant et chauffent jusqu'à une température comprise entre 0 et 99 degrés. Le dénudage se fait alors en appuyant les mâchoires sur le câble. En effet, ce

dispositif a l'avantage de ne pas nécessiter une pression très forte sur le câble électrique. En revanche, sa conception reste complexe.

- Dénudation au laser : le laser est un procédé utilisé par les professionnels pour sa précision, sa répétabilité et sa robustesse [7]. Il est également utilisé pour des applications pour lesquels la présence du moindre défaut sur l'âme conductrice n'est pas acceptable et lorsque la géométrie extérieure du câble à dénuder est incompatible avec un procédé mécanique.

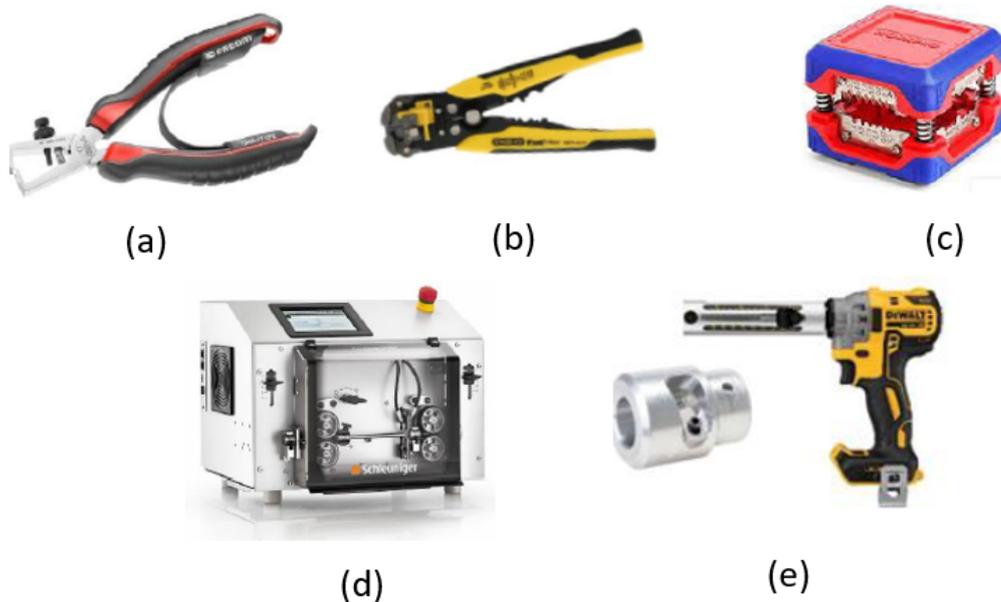


FIGURE 1.8 – Pince à dénuder classique (a), Pince à dénuder automatique (b), Dénudeur workpro (c), Machine Schleuniger (d), Kit de dénudage DEWALT (e)

Le dénudage mécanique est la technique la plus répandue aujourd'hui dans le monde industriel. Dans le contexte du projet, c'est la plus facile à adapter. De plus, en termes de coût, c'est celle qui présente le moins d'exigences financières. Nous avons donc choisi dans notre projet les technologies de dénudage mécanique pour leur faisabilité et leur facilité de mise en œuvre sur le robot YuMi et également pour nous permettre de valider la loi de commande externe que nous cherchons à appliquer sur le robot. On retrouve dans le cadre du dénudage mécanique :

Pince à dénuder (avec mâchoires en V) : en forme de X, ces outils sont dotés d'une partie active (partie en contact avec les matériaux) et d'une partie passive recouverte d'une enveloppe isolante pour la prise. La partie active est constituée de deux têtes en forme de V qui sont utiles pour serrer un câble. Elle est également constituée d'une vis contre écrou qui sert à régler le diamètre souhaité. Pour dénuder un câble avec cet outil, il faut régler la vis pour le diamètre souhaité. Ensuite, il faut serrer le câble et appliquer une force en tournant autour du câble. Enfin, tirer sur le câble de manière à ce que la gaine isolante se détache de l'âme conductrice sur une longueur définie.

Pince à dénuder automatique : il existe plusieurs variantes de pinces à dénuder automatiques qui fonctionnent toujours selon le même concept. Pour les sections comprises entre  $0,4\text{mm}^2$  et  $6\text{mm}^2$ , un seul réglage suffit. La partie active de la pince est constituée de crochets qui serrent le câble et le maintiennent en place, et de lames mobiles qui serrent, appliquent une force et tirent sur la gaine isolante qui sera détachée de la couche conductrice. Ces pinces permettent de dénuder un câble électrique en un seul mouvement.

Pinces à dénuder à encoches : elles sont constituées d'une partie active composée de deux mâchoires. Chacune des mâchoires comporte une ou plusieurs encoches permettant de dénuder des câbles de diamètres spécifiques. Pour dénuder un câble avec ces pinces, il faut choisir l'encoche correspondante et pincer le câble. Ensuite, il faut exercer une pression sur la partie passive de l'outil et tirer le câble vers l'extérieur pour séparer la gaine isolante de l'âme du câble.

Par ailleurs, on trouve également des dénudeurs universels de différentes formes, chacune basée sur une technologie particulière. Par exemple, une forme cubique composée d'une partie supérieure et d'une partie inférieure, toutes deux équipées de lames de différents diamètres (voir Figure 1.8). Ces deux parties sont reliées entre elles par un jeu de ressorts. Il existe encore des formes cylindriques (voir Figure 1.8) et des formes plus complexes à dénuder avec ce dernier outil, il faut régler le calibre du câble correspondant, puis il faut tendre le câble, tourner autour du câble, et se déplacer sur la longueur du câble. Enfin, la gaine isolante doit être retirée manuellement. Les machines à dénuder sont également une solution conçue pour dénuder les câbles et les fils. Elles se composent généralement d'une sorte de tapis roulant, d'un dispositif de guidage pour orienter les fils et d'un outil de coupe pour dénuder l'isolant. Certains appareils automatiques équipés d'un moteur qui met en mouvement le système d'embrayage électronique relié au roulement approprié sont également une solution proposée sur le marché, appelée kit de dénudage. Ce roulement est doté d'une lame interne qui va permettre de dénuder les câbles électriques en toute simplicité.

## 1.4 Conclusion

Au cours de ce chapitre, nous avons évoqué trois aspects différents sur lesquels porte notre sujet de thèse. Dans la première partie, nous avons abordé l'évolution des robots industriels collaboratifs au cours du temps ainsi que leurs intérêts dans l'industrie d'aujourd'hui dite industrie 4.0. De même, nous avons présenté certaines statistiques témoignant de l'intérêt d'un tel sujet de recherche. Les robots à deux bras sont également un thème très intéressant dont la problématique a été abordée il y a maintenant des années avec [80]. Nous avons donc exposé les travaux déjà existants traitant ce concept et plus particulièrement le sujet de la coordination entre les deux bras et les différentes approches cherchant à décrire l'assemblage et la manipulation à deux bras. En troisième et dernière partie, nous avons brièvement évoqué l'application du dénudage, celle que nous cherchons à robotiser à travers ce travail de thèse. En effet, nous avons défini l'application et invoqué les techniques et les outils existants actuellement en industrie pour réaliser de telles applications.

Ayant couvert la totalité ou une bonne partie de l'état de l'art qui porte au sujet de notre thèse, nous allons pouvoir passer à la présentation et au traitement des étapes retenues pour le mener à bien. Tout d'abord, la première étape est la modélisation des robots et plus particulièrement la modélisation de notre robot YuMi traitée dans le chapitre suivant.

# Chapitre 2

## Modélisation des robots

### 2.1 Introduction

Ce chapitre expose la première démarche adoptée pour parvenir à l'objectif de notre thèse, il porte sur la modélisation des robots et plus précisément sur la modélisation de notre robot YuMi.

En guise d'introduction, quelques notions de base de la robotique en général et des robots manipulateurs en particulier sont présentées, telles que les degrés de liberté, la redondance, les transformations de base et la méthode de Denavit-hartenberg modifiée. Ensuite, nous présentons le robot YuMi IRB14000, pour lequel une modélisation géométrique, cinématique et dynamique sera définie, calculée et validée grâce aux différents travaux de recherche portant très pertinemment sur la modélisation des robots.

Pour la modélisation dynamique de notre robot, nous avons intégré un modèle de frottement que nous présentons également.

### 2.2 Notions de base

Nous abordons succinctement dans cette partie quelques notions fondamentales propres à la modélisation des robots manipulateurs.

#### 2.2.1 Degrés de liberté d'un corps rigide

La notion de degré de liberté (DDL) d'un corps rigide est définie par le nombre de mouvements indépendants qu'il peut effectuer. Dans l'espace, tout corps rigide non contraint possède six degrés de liberté : trois mouvements de translation le long des axes  $x$ ,  $y$  et  $z$  et trois mouvements de rotation autour des axes  $x$ ,  $y$  et  $z$  respectivement. Il convient également de noter que :

- Plus le nombre de DDL est élevé, plus la position de l'outil est flexible.
- Chaque degré de liberté nécessite une articulation. Les degrés de liberté dans le bras définissent la configuration.
- Trois degrés de liberté dans le poignet donnent plus de flexibilité à l'effecteur terminal.
- Un minimum de six degrés de liberté est nécessaire pour positionner l'outil en tout point de l'espace de travail du robot.

### 2.2.2 La redondance

La redondance a pour objectif de rendre le robot plus habile et plus rentable. Les redondances peuvent être classées en deux catégories, à savoir :

**Redondance cinématique :** On parle de redondance cinématique lorsqu'un manipulateur de robot a plus de DDL que nécessaire pour une tâche spécifique. Ceci permet de spécifier plus librement la configuration pour éviter les obstacles (Figure 2.1), ou pour minimiser le mouvement nécessaire des articulations. Un robot avec six DDL peut être redondant cinématiquement pour certaines tâches. Le forage, par exemple, où la rotation autour de l'axe de forage n'est pas importante.

**Redondance intrinsèque :** Tout robot ayant au moins sept DDL est intrinsèquement redondant. Le suivi d'une trajectoire dans l'espace nécessite six DDL, ce qui signifie qu'un robot avec sept DDL peut atteindre chaque pose sur la trajectoire avec plusieurs configurations différentes. Le robot YuMi a sept DDL, ce qui le rend à la fois cinématiquement et intrinsèquement redondant.



FIGURE 2.1 – Le robot YuMi en position de redondance pour éviter les obstacles

### 2.2.3 Transformations Homogènes

En robotique, afin de calculer la position et l'orientation des articulations du robot les unes par rapport aux autres ou de décrire la position et l'orientation des objets, on attribue un ou plusieurs repères à chaque articulation du robot et à chaque objet de la cellule de commande.

La transformation de ces repères est donc une notion incontournable dans la modélisation et la programmation d'un robot. Cette transformation permet également de préciser la trajectoire et la vitesse de l'effecteur terminal du robot pour une tâche déterminée et de contrôler les forces exercées lors de son interaction avec l'environnement.

Par ailleurs, le passage en translation et/ou rotation, d'un repère  $R_i$  à un repère  $R_j$  (Figure 2.2) est très répandu et se définit par une matrice  ${}^i T_j$  appelée la matrice de transformation homogène.

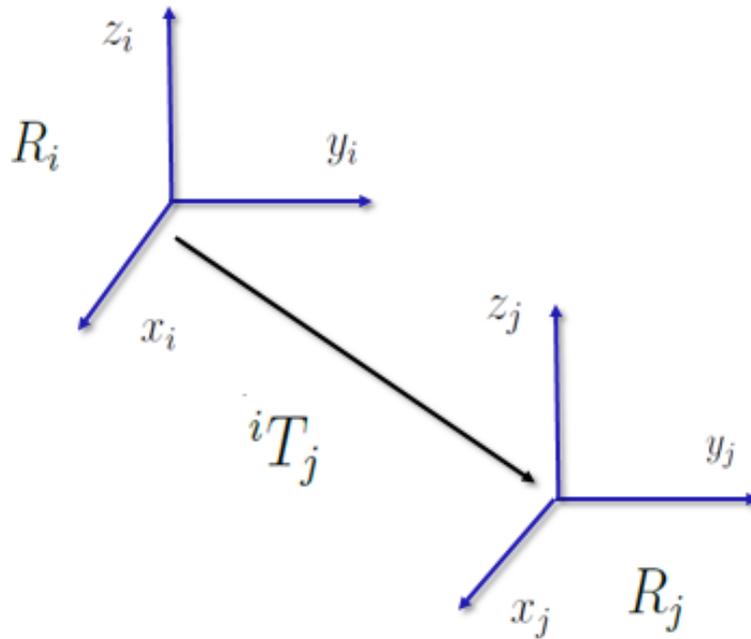


FIGURE 2.2 – Transformation des repères

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0_3 & 1 \end{bmatrix}$$

- ${}^i A_j$  : matrice  $[3 \times 3]$  représentant l'orientation de repère  $R_j$  dans le repère  $R_i$
- ${}^i P_j$  : vecteur  $[3 \times 1]$  représentant la position du centre du repère  $R_j$  dans le repère  $R_i$ .

#### 2.2.4 Convention de Denavit-Hartenberg modifiée

Toute chaîne cinématique, et plus particulièrement un robot, est constituée d'une série de liaisons cinématiques reliées par des articulations rotoïdes ou prismatiques correspondant au degré de liberté en rotation ou en translation. Il existe de nombreuses approches pour représenter mathématiquement une chaîne cinématique et la principale différence entre ces dernières repose sur la fixation de repères de coordonnées.

Dans ce contexte, la méthode Denavit-Hartenberg modifiée a été instaurée pour la première fois dans [94]; une méthode fondée sur la célèbre théorie D-H (Denavit-Hartenberg) [47]; très puissante pour les robots séries, mais ambiguë dans le cas des robots à structure tridimensionnelle ou à boucle fermée.

En effet, cette méthode fréquemment utilisée repose sur des matrices de transformation homogènes plus performantes pour le positionnement des repères de coordonnées et permettant de réduire les opérations mathématiques/arithmétiques pour décrire un éventuel déplacement cinématique.

Dans la méthode DHM, on utilise quatre paramètres appelés "paramètres DHM". Ceux-ci sont utilisés pour définir la géométrie de la liaison et le déplacement relatif de l'articulation.

Nous allons à présent nous concentrer sur toutes les propriétés caractéristiques des paramètres de la DHM afin de modéliser le lien cinématique considéré dans Figure 2.3.

Pour cela quelques notations sont considérées :

- $C_i$  : le corps correspond à l'articulation  $i$
- $R_i(O_i, X_i, Y_i, Z_i)$  : le repère attaché au corps  $i$

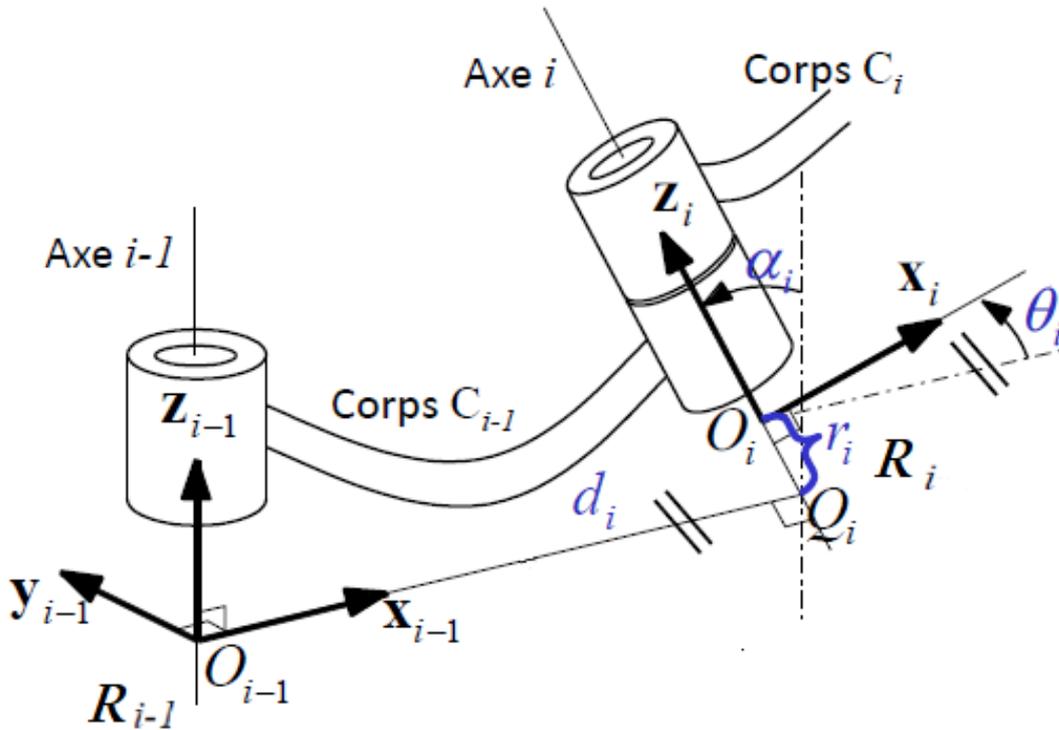


FIGURE 2.3 – Convention de Denavit-Hartenberg modifiée

Chaque paramètre DHM est défini comme suit :

- $\alpha_i$  : angle entre  $Z_{i-1}$  et  $Z_i$  correspondant à une rotation autour de  $X_{i-1}$ .
- $d_i$  : distance entre  $Z_{i-1}$  et  $Z_i$  le long de  $X_{i-1}$  (distance entre  $O_{i-1}$  et  $Q_i$ ).
- $r_i$  : distance signée entre  $X_{i-1}$  et  $X_i$  le long de  $Z_i$  (distance entre  $Q_i$  et  $O_i$ ).
- $\theta_i$  : angle entre  $X_{i-1}$  et  $X_i$  correspondant à une rotation autour de  $Z_i$

Par conséquent, la modélisation géométrique d'un robot, selon la méthode DHM revient à déterminer la matrice de transformation homogène  $T$  décrivant la position de l'organe terminal dans le repère fixé à la base du robot à partir des matrices homogènes décrivant la position de chaque articulation du robot par rapport à l'articulation précédente. On obtient ainsi la matrice de transformation homogène (Eq. 2.1) qui donne la position du repère  $i$  (corps  $C_i$ ) par rapport au repère  $i-1$  (corps  $C_{i-1}$ ) :

$${}^{i-1}T_i = \text{Trans}(x, d_i) \text{Rot}(x, \alpha_i) \text{Trans}(z, r_i) \text{Rot}(z, \alpha_i) \quad (2.1)$$

Avec :

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & d_i \\ C\alpha_i S\theta_i & C\alpha_i C\theta_i & -S\alpha_i & -r_i S\alpha_i \\ S\alpha_i S\theta_i & S\alpha_i C\theta_i & C\alpha_i & r_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notons que  $C\theta_i$ ,  $C\alpha_i$ ,  $S\theta_i$  et  $S\alpha_i$  sont les abréviations respectives de  $\cos(\theta_i)$ ,  $\cos(\alpha_i)$ ,  $\sin(\theta_i)$  et  $\sin(\alpha_i)$ .

## 2.3 Modélisation

### 2.3.1 SYMORO+

SYMORO (SYmbolic MOdeling of Robots) est un logiciel développé par l'équipe de robotique du laboratoire IRCyN. Il permet de générer automatiquement des modèles symboliques des robots [91]. De plus, il met à la disposition de l'utilisateur des outils rapides, sûrs et efficaces qui lui permettent de générer les différents modèles nécessaires au contrôle, à la simulation et à la conception des robots (modèles géométriques, modèles cinématiques, modèles dynamiques, modèles d'identification...). Symoro+ est exclusivement compatible avec le système d'exploitation Windows et seulement avec le logiciel de calcul Mathematica 9.0.

La convention Denavit-Hartenberg modifiée est utilisée dans ce logiciel. Une fois les paramètres nécessaires entrés dans l'interface interactive (Figure 2.4), le logiciel est en mesure de fournir les modèles symboliques qui peuvent être directement utilisés par différents environnements ( Matlab, C++ ...).

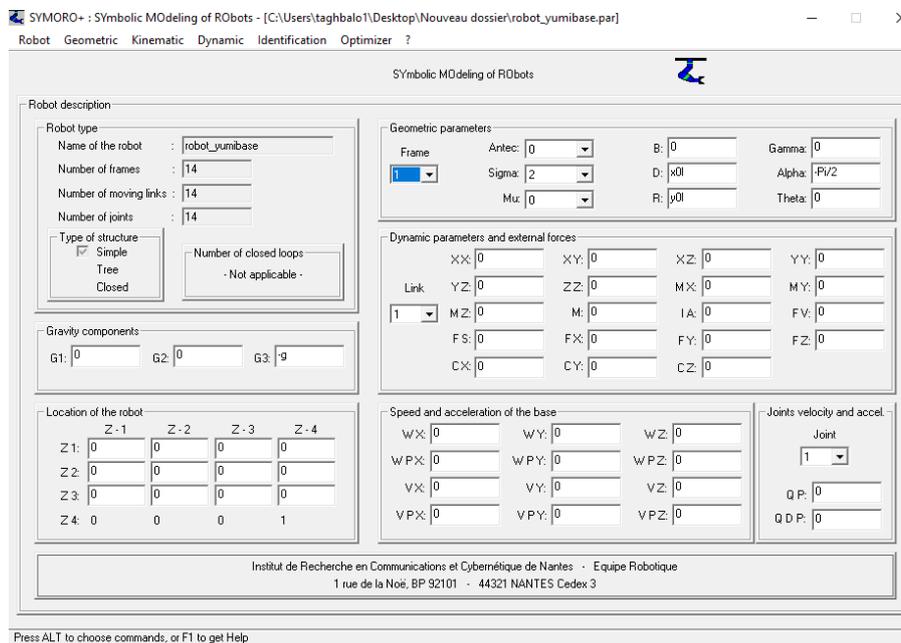


FIGURE 2.4 – Interface du logiciel SYMORO+

### 2.3.2 Description du robot YuMi IRB14000

YuMi est le robot à deux bras, conçu pour que les humains travaillent aux côtés des robots. La conception compacte à deux bras de YuMi imite délibérément le corps humain, lui donnant une sensation d'intimité et de convivialité, et sa taille est parfaite pour s'intégrer dans un poste de travail humain.

Le YuMi est conçu pour tirer pleinement parti du potentiel de la collaboration homme-machine dans l'assemblage de petites pièces. En tant que robot à deux bras construit pour l'assemblage de petites pièces, YuMi offre une nouvelle solution grâce à sa conception et à sa précision intrinsèque. Les deux bras en alliage léger de YuMi, tous deux dotés de sept axes de liberté, reproduit les mouvements des membres humains, ce qui améliore considérablement l'utilisation de l'espace tout en répondant aux besoins de production flexibles et agiles de l'industrie (électronique) grand public. Ses caractéristiques techniques sont représentées dans Table 2.1



FIGURE 2.5 – Robot YuMi IRB14000 au laboratoire LCFC

La solution YuMi intègre une conception à double bras, des manipulateurs flexibles, des systèmes d'alimentation universels et une commande de mouvement par caméra pour les systèmes de positionnement de travail, qui peuvent également être utilisés dans d'autres applications d'assemblage de petites pièces.

TABLE 2.1 – Caractéristiques techniques de YuMi

<b>Charge utile (kg)</b>	<b>0.5 par bras</b>
<b>Portée maximale (mm)</b>	<b>559</b>
<b>Axes contrôlés</b>	<b>14 (7 par bras)</b>
<b>Précision (mm)</b>	<b>0.02</b>
<b>Poids (Kg)</b>	<b>35</b>

Le YuMi est un robot léger, dans son ensemble, il ne pèse que 35 kg, ce qui réduit le risque de dangereuses chutes. Il peut seulement soulever des charges très légères. En effet, les éléments que le YuMi peut manipuler ne peuvent pas dépasser 500 grammes pour chaque bras.

### 2.3.3 Modélisation géométrique

#### Modèle géométrique direct

La modélisation d'un robot en série nécessite tout d'abord d'établir son modèle mathématique. Les méthodes de modélisation sont différentes, nous avons choisi le concept de Denavit-Hartenberg Modifiés (DHM) largement utilisé en robotique (présenté dans la section précédente (2.2.3)).

On considère les différents paramètres de Denavit-Hartenberg modifiés, les limites articulaires et les vitesses maximales définis dans (Table 2.2) ci-dessous pour modéliser un seul bras du robot (Figure 3.6).

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. On doit trouver alors la position et l'orientation du repère  $R_{T_{cp}}$  par rapport au repère  $R_0$ .

TABLE 2.2 – Paramètres DHM du bras gauche de YuMi

Liaison	$\alpha_j(\text{rad})$	$d_j(\text{mm})$	$r_j(\text{mm})$	$\theta_j$	$[\theta_{jmin} \ \theta_{jmax}]$	$vitesse_{max}$
1	0	0	166	$-\pi + q_1$	$[-168,5 \ 168,5]$	180
2	$\pi/2$	-30	0	$-\pi + q_2$	$[-143,5 \ 43,5]$	180
3	$\pi/2$	30	251,5	$q_3$	$[-123,5 \ 80]$	180
4	$-\pi/2$	40,5	0	$-\pi/2 + q_4$	$[-290 \ 290]$	400
5	$-\pi/2$	-40,5	265	$\pi + q_5$	$[-88 \ 138]$	400
6	$-\pi/2$	27	0	$-\pi + q_6$	$[-229 \ 229]$	400
7	$-\pi/2$	-27	36	$\pi + q_7$	$[-168,5 \ 168,5]$	180

Pour cela, on calcule la matrice homogène  ${}^0T_7$  à partir des paramètres de Denavit-Hartenberg. Dans ce qui suit,  $C_i$  désigne  $\cos(\theta_i)$  et  $S_i$  désigne  $\sin(\theta_i)$ .

$${}^0T_7 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7 \quad (2.2)$$

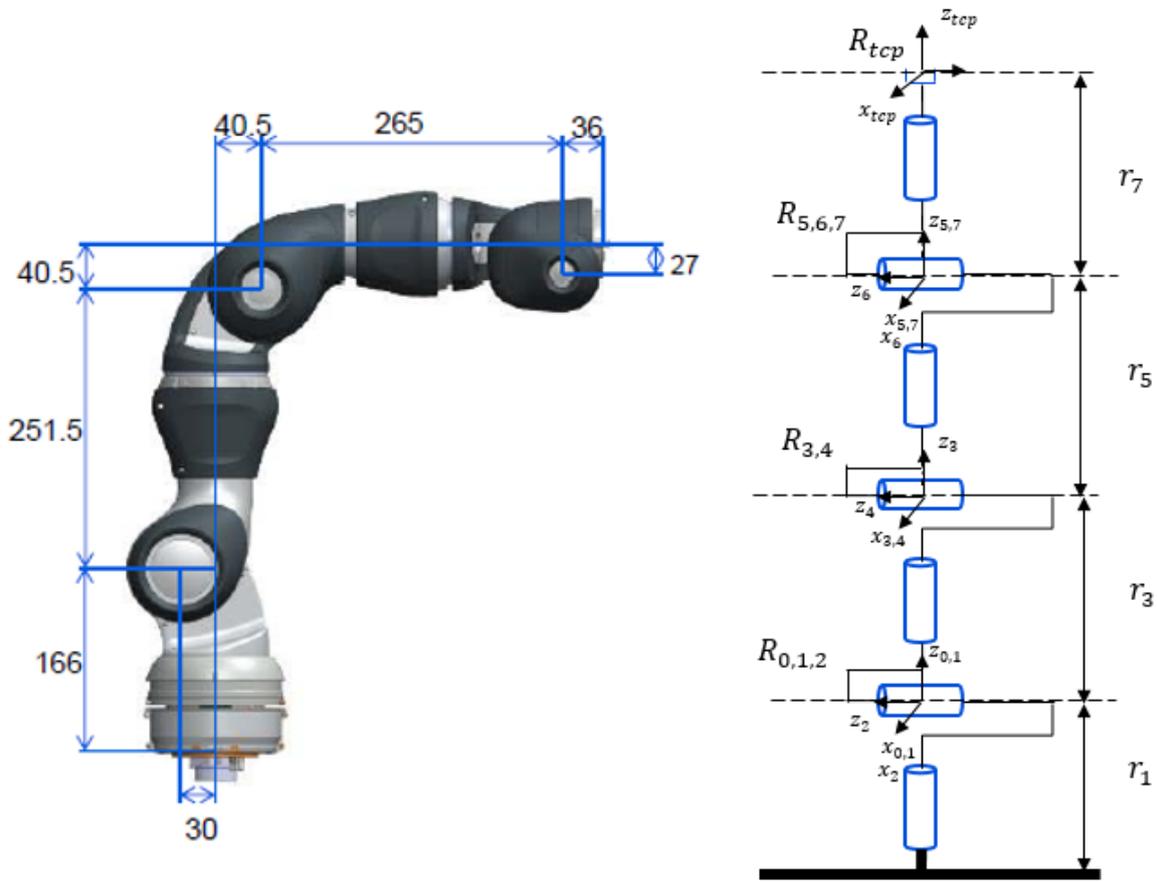


FIGURE 2.6 – Bras droit du YuMi et ses repères DHM.

Les matrices de transformation correspondantes à chaque articulation sont les suivantes :

$${}^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & -d_1 \\ -S_1 & C_1 & -0 & 0 \\ 0 & 0 & 1 & -r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & d_2 \\ 0 & 0 & -1 & -r_2 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} C_3 & -S_3 & 0 & d_3 \\ 0 & 0 & -1 & -r_3 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3T_4 = \begin{bmatrix} C_4 & -S_4 & 0 & d_4 \\ 0 & 0 & 1 & r_4 \\ -S_4 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4T_5 = \begin{bmatrix} C_5 & -S_5 & 0 & d_5 \\ 0 & 0 & 1 & r_5 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5T_6 = \begin{bmatrix} C_6 & -S_6 & 0 & d_6 \\ 0 & 0 & 1 & r_6 \\ -S_6 & -C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

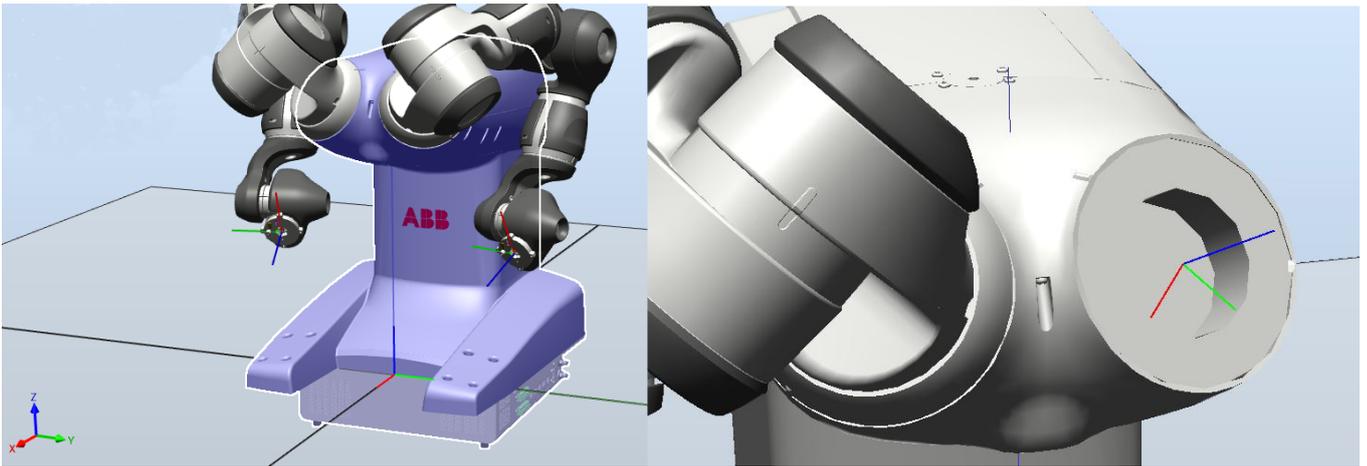


FIGURE 2.7 – Base du bras gauche et base du robot YuMi [RobotStudio]

$${}^6T_7 = \begin{bmatrix} C_7 & -S_7 & 0 & d_7 \\ 0 & 0 & 1 & r_7 \\ -S_7 & -C_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les composants de la matrice  ${}^0T_7$  du modèle géométrique du bras gauche ainsi que celui du bras droit sont élaborés dans l'appendice A.

## Modèle géométrique inverse

Une modélisation géométrique inverse consiste à déterminer les coordonnées articulaires amenant l'organe terminal dans une position souhaitée, spécifiée par ses coordonnées opérationnelles; c'est-à-dire connaissant l'expression (désirée)  $T$  de la matrice de transformation homogène  ${}^0T_n$ .

En effet, la solution d'un problème de géométrie inverse consiste à résoudre un système d'équations non linéaires dont la complexité impose de traiter :

### L'existence des solutions

Un robot manipulateur est considéré comme résoluble lorsqu'il est possible de calculer toutes les configurations possibles permettant d'atteindre une position donnée. En revanche, on peut se retrouver en cas d'absence de solutions. Par exemple, lorsque la position désirée se trouve en dehors de l'espace de travail du robot. Cet espace est limité par le nombre de DDL, les trajectoires des articulations et la longueur des segments du robot.

### Le nombre de solutions

Le nombre de solutions dépend de l'architecture du robot manipulateur. Par exemple, dans le cas d'une redondance implicite (par rapport à la tâche à effectuer) qui résulte du fait que le robot ne peut pas déplacer son organe terminal dans certaines directions ou dans le cas d'une configuration singulière, le robot perd un ou plusieurs DDL ce qui rend la solution plus complexe à trouver. Tout cela est dû aux valeurs numériques particulières prises par les paramètres

pour décrire la position désirée.

Il n'existe donc pas de formule générale pour résoudre ce problème. Tout dépend de la pose et des valeurs des paramètres géométriques du robot. Dans le cas d'un robot à 6 DDL avec des articulations rotoïdes, le nombre maximum de solutions est de 16.

On parle de solution finie dans le cas où toutes les solutions peuvent être calculées sans ambiguïté. La forme explicite est alors un ensemble de vecteurs  $[q_1, q_2, \dots, q_n]$  satisfaisant l'équation :

$$X = f(q_i) \quad (2.3)$$

### Méthode de résolution

- Résolution locale numérique du PGI (Algorithme de Newton-Raphson [75])
- Méthode de détermination des solutions sous forme explicite (résolution du modèle géométrique inverse).

Dans le cadre de notre thèse, ce modèle est utilisé pour la simulation de la commande du robot et nous avons choisi de résoudre ce problème explicitement en posant l'hypothèse ( $q_3 = 0$ ). Cette hypothèse a un impact certain sur les performances de ce modèle dans son application réelle, toutefois celle-ci ne sera pas retenue dans les expérimentations sur le robot réel.

Le calcul du modèle géométrique inverse de notre robot est détaillé dans l'appendice B.

#### 2.3.4 Matrice de transformation de base du robot

La matrice de transformation de la base est la transformation de la base du bras vers la base du robot. Cette transformation permet de décrire les positions des organes terminaux des deux bras dans le même repère. La base du robot est, *a priori* définie au pied du robot, tandis que la base du bras est définie à la première articulation. Pour les autres robots ABB, ces deux repères sont identiques, mais pour YuMi, la base est située au pied du robot entre les bras, alors que les bras sont montés à une certaine distance de la table avec une légère rotation vers l'extérieur et vers le haut comme le montrent les figures ci-après (Figure 2.7).

Le calcul de la base de transformation se base sur la formule suivante :

$${}^{base}T_{T_{cp}} = {}^{base}T_{bras} \cdot {}^{bras}T_{T_{cp}} \quad (2.4)$$

où :

- ${}^{base}T_{bras}$  : est la transformation de la base du bras, droit ou gauche, à la base du robot (voir appendice C).
- ${}^{bras}T_{T_{cp}}$  : est la transformation de la position d'effecteur à la base du bras.

TABLE 2.3 – Paramètres DHM du base de robot

$d_j(mm)$	$r_j(mm)$	$\alpha_j(rad)$	$\theta_j(rad)$
$x_0$	$y_0$	$-\pi/2$	0
0	0	$\pi/2$	0
0	$z_0$	0	0
0	0	0	$A_x$
0	0	$-\pi/2$	$A_y+\pi/2$
0	0	$\pi/2$	$A_z+\pi/2$
0	0	$-\pi/2$	$-\pi/2$

Afin de modéliser correctement le robot, il a fallu trouver les paramètres DHM correspondant à la configuration de la matrice de transformation (Table 2.3). Nous avons donc calculé la position du centre de repère du bras et la matrice de rotation à partir des mesures effectuées sur le robot lui-même et sur son simulateur sur Robotstudio. À partir de la matrice de rotation, nous avons pu définir les angles d'Euler correspondants en se basant sur les travaux de [123].

Pour toutes les articulations du tableau (Table 2.3), nous avons considéré que leurs repères correspondants sont des repères immobiles (non prismatiques ou rotatifs). Sur Symoro+, nous exprimons cela en fixant le paramètre  $\sigma$  à 2.

Les trois premières articulations correspondent à la position du centre de la base du robot  $P=[x_0, y_0, z_0]$ , respectivement, selon les directions  $x$ ,  $y$  et  $z$ . Quant aux quatre autres articulations, elles correspondent à la matrice de rotation, selon les paramètres DH des EAJ suivant la convention Z-Y-X, de la base du bras à la base du robot [123]. Avec  $A_x$ ,  $A_y$ ,  $A_z$  les angles de rotation en  $rad$ , selon  $x$ ,  $y$  et  $z$  respectivement, sans oublier de préciser que les deux bras du robot sont identiques à l'exception de la position de la base qui change. Les index  $l$  et  $r$  sont donc respectivement les index indiquant le bras gauche et le bras droit du robot YuMi.

avec :

$$\begin{aligned}
 - A_{xl} &= -0,85 & A_{yl} &= 0,56 & A_{zl} &= -0,99 & x_{0l} &= -0,004 & y_{0l} &= 0,0543 & z_{0l} &= 0,3875 \\
 - A_{xr} &= 0,85 & A_{yr} &= 0,56 & A_{zr} &= 0,99 & x_{0r} &= -0,004 & y_{0r} &= -0,0543 & z_{0r} &= 0,3875
 \end{aligned}$$

NB : les angles sont en  $rad$  et les positions sont en  $m$ .

Les figures (Figure 2.8) et (Figure 2.9) illustrent respectivement, le simulateur réalisé sur Matlab sans prise en compte de la matrice de transformation de base du bras à la base du robot ; ainsi que le simulateur, réalisé cette fois avec la matrice de base et les deux bras sur la Matlab Robotic Toolbox.

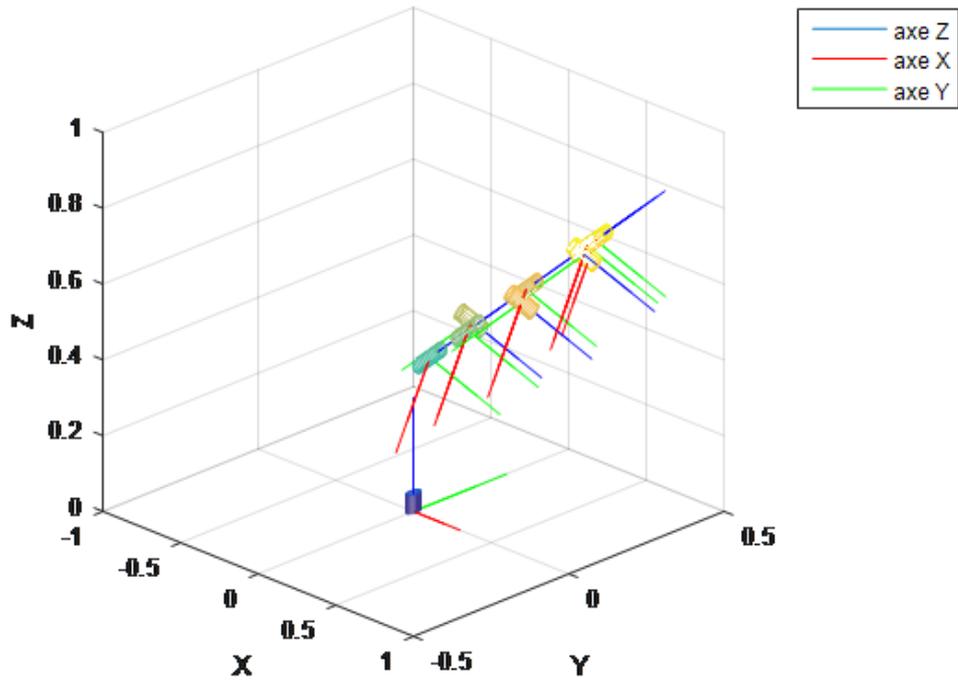


FIGURE 2.8 – Simulateur du bras gauche du robot sous Matlab

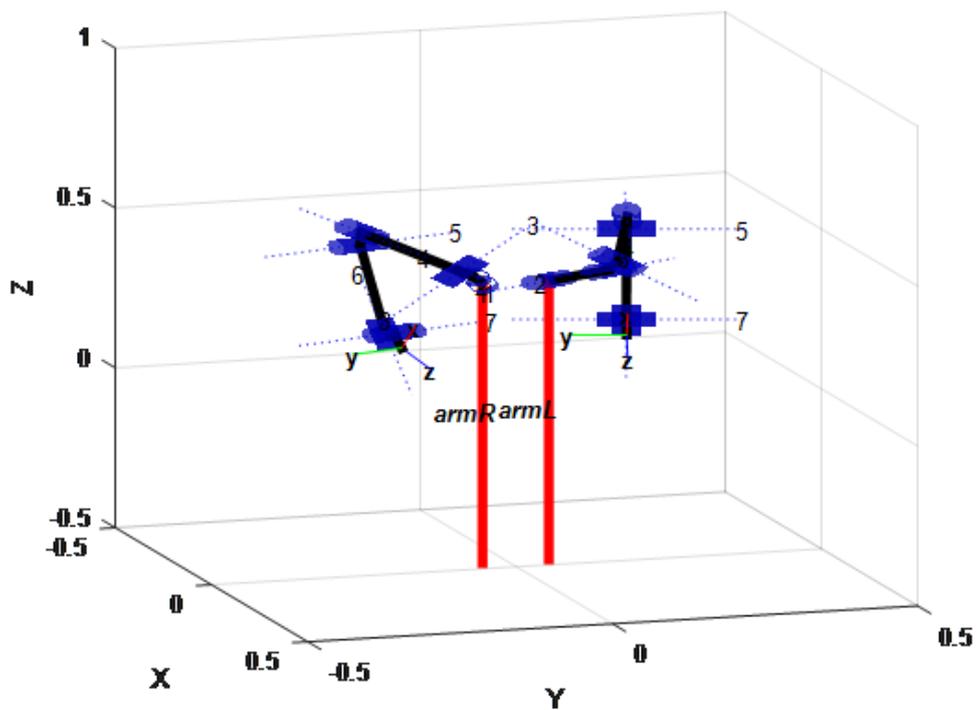


FIGURE 2.9 – Simulateur du robot avec Peter Corke Toolbox sur Matlab

### 2.3.5 Modèle cinématique

Le modèle cinématique est, au sens propre, un modèle de vitesse. Il décrit les vitesses de l'organe terminal en fonction des dérivées, par rapport au temps, des variables actives (positions et vitesses des articulations). Plus encore, la cinématique est une représentation linéaire du système d'équations non linéaires correspondant au modèle géométrique, ce qui permet facilement d'obtenir des transformations inverses.

#### Modèle cinématique direct

Ce modèle peut être exprimé comme suit :

$$\dot{X} = \begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)_{[6 \times n]} \dot{q} \quad (2.5)$$

Avec :

- $\dot{X}$  : le vecteur de dimension  $[n \times 1]$ , exprimant le vecteur de vitesse linéaire, selon les trois axes  $x$ ,  $y$  et  $z$  respectivement,  $v = [v_x, v_y, v_z]$  et le vecteur de vitesse angulaire de l'organe terminal  $\omega = [\omega_x, \omega_y, \omega_z]$  du robot, selon  $\alpha$ ,  $\beta$  et  $\gamma$ ; les axes des rotations calculés et exprimés dans le repère de base de robot  $T_0$ .
- $J$  : la matrice jacobienne  $[6 \times n]$ , c'est-à-dire  $[6 \times 6]$  pour un robot non redondant, égale à  $\frac{\partial X}{\partial q}$  en fonction de la configuration articulaire, qui donne les variations élémentaires  $dX$  des coordonnées opérationnelles en fonction des variations articulaires  $dq$ .
- $\dot{q}$  : le vecteur de dimension  $[n \times 1]$  regroupant les  $n$  vitesses articulaires du robot

$$\dot{q} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]$$

Il convient de noter que, la jacobienne joue un rôle très important dans la modélisation de tout robot, parce que non seulement elle établit la relation entre les forces exercées par l'organe terminal sur l'environnement et les forces et les couples des moteurs, mais elle facilite également le calcul des singularités et celui de la dimension de l'espace opérationnel.

#### Modèle cinématique inverse

Le modèle cinématique inverse a pour objectif de calculer, à partir d'une configuration  $q$ , les vitesses des articulations  $\dot{q}$ , qui accordent au repère de l'outil une vitesse opérationnelle imposée  $\dot{X}$ . Théoriquement, deux méthodes nous permettent d'obtenir le modèle cinématique inverse, à savoir la première, par dérivation du modèle géométrique inverse (longue et complexe), et la seconde par inversion de la matrice jacobienne  $J$  (couramment utilisée en robotique). Donc, pour obtenir ce modèle inverse, le problème revient à inverser la matrice jacobienne  $J$ , pour autant que cela soit possible.

En partant de l'équation (Eq. 2.5), ce modèle se présente comme ci-dessous :

$$\dot{q} = J^{-1} \dot{X} \quad (2.6)$$

Quand, le système est redondant comme c'est le cas pour notre robot YuMi, chaque bras a 7 DDL, il convient d'utiliser la fonction pseudo-inverse  $J^+$  (la matrice inverse généralisée (Eq. 2.7) plutôt que la fonction inverse.

$$J^+ = (J^T J)^{-1} J^T \quad (2.7)$$

Pour le calcul de la jacobienne de notre robot, voir l'appendice D.

### 2.3.6 Modélisation dynamique

La relation entre le mouvement du robot manipulateur défini par les positions, les vitesses et les accélérations, ainsi que les forces appliquées sur ce dernier est décrite par une modélisation dynamique. À la différence de la cinématique, où les forces ne sont pas prises en compte, la dynamique joue un rôle important dans la simulation du mouvement du robot en présence de couples d'actionnement des moteurs mais utile également pour l'optimisation de la structure du robot et la conception des différents algorithmes de commande.

#### Modèle dynamique inverse :

Dans la littérature, pour obtenir le modèle dynamique du robot, notamment un robot manipulateur, on peut utiliser la méthode de Lagrange, la méthode de Newton-Euler et la méthode de Kane [92] [139]. Soit un robot manipulateur avec  $n$  degrés de liberté ; son modèle dynamique peut être écrit dans l'espace articulaire sous la forme générale :

$$M(q)\ddot{q} = \tau - H(\dot{q}, q) + J^T(q)F_{ex} \quad (2.8a)$$

$$I_a\ddot{\theta} = \tau_m - N^{-T}\tau - F_{frot} \quad (2.8b)$$

Avec :

- $M(q)(n \times n)$  : la matrice d'inertie de robot, une matrice symétrique, définie positive
- $H(q, \dot{q})(n \times 1)$  : la somme des efforts centrifuges, de Coriolis et de gravité
- $F_{ex}(n \times 1)$  : le vecteur des efforts extérieurs appliqués par l'environnement sur le robot, exprimés dans le repère  $T_{cp}$
- $J^T(q)(n \times n)$  : la matrice jacobienne transposée du repère  $T_{cp}$  par rapport au repère de base
- $\tau(n \times 1)$  : le vecteur des couples générés par les réducteurs
- $q, \dot{q}$  et  $\ddot{q}$  : respectivement les vecteurs des positions, des vitesses et des accélération articulaires du robot
- $I_a(n \times n)$  : la matrice d'inertie des axes du moteur
- $\tau_m(n \times 1)$  : le vecteur des couples-moteurs
- $F_{frot}(\dot{q})(n \times 1)$  : le vecteur du frottement sur les axes du moteur
- $N^{-1}$  : l'inverse de la matrice  $N$  du rapport de transmission des réducteurs
- $\theta, \dot{\theta}$  et  $\ddot{\theta}$  : respectivement le vecteur des positions, des vitesses et des accélérations angulaires, des axes du moteur

Les rapports de transmission  $N$  de notre robot YuMi IRB14000 forment une matrice diagonale, conformément à la formule suivante :

$$N = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (2.9)$$

### Modèle dynamique direct :

Le modèle dynamique direct (MDD) est un modèle qui exprime les accélérations articulaires en fonction des efforts, des positions et des vitesses. Ce modèle permet de simuler le comportement temporel du robot. En effet, il s'agit d'une équation différentielle non linéaire implicite obtenue à partir du modèle dynamique :

$$\ddot{q} = M(q)^{-1}(\tau - H(\dot{q}, q) + J^T(q)F_e) \quad (2.10)$$

Dans cette thèse, nous utilisons les deux modèles pour valider la simulation de contrôle de position et des forces extérieures dans l'espace opérationnel du robot.

### Linéarisation du modèle dynamique par rapport aux paramètres de base :

Le modèle dynamique (inverse) des robots manipulateurs s'écrit sous forme linéaire par rapport à des paramètres dynamiques identifiables, appelés paramètres inertiels de base. Dans ces paramètres, en plus des paramètres d'inertie, on trouve les paramètres liés au modèle de frottement choisi (voir la section suivante pour plus de détails). Dans notre cas, il s'agit de trois paramètres supplémentaires  $Fv_i$ ,  $Fc_i$  et  $Off_i$ . La linéarité est obtenue en supprimant les paramètres inertiels standards  $X_{standard}$  (Eq. 2.11) qui n'interviennent pas dans le modèle dynamique et en regroupant certains d'entre eux sous forme de relations linéaires.

Ces opérations d'élimination et de regroupement sont basées depuis les années 80 [17] sur un calcul théorique de l'énergie mécanique des corps de robots ou sur un calcul numérique basé sur la factorisation QR ou SVD [60] [104].

$$X_{standard} = [XX_i, XY_i, XZ_i, YZ_i, ZZ_i, MX_i, MY_i, MZ_i, M_i, Ia_i, Fv_i, Fc_i, Off_i]^T \quad (2.11)$$

avec :

- $XX_i, XY_i, XZ_i, YZ_i$  et  $ZZ_i$  : Paramètres du tenseur d'inertie du corps  $i$
- $MX_i, MY_i$  et  $MZ_i$  : Moments d'inertie de corps  $i$
- $Fv_i$  et  $Fc_i$  : Coefficients de frottement de Coulomb et visqueux
- $M_i$  : Masse de corps  $i$
- $Off_i$  : Offset de l'articulation  $i$
- $Ia_i$  : Inertie du moteur  $i$

En effet, ces paramètres de base sont les seuls paramètres identifiables avec le modèle linéaire et représentent un ensemble minimum de paramètres nécessaires pour calculer la dynamique inverse du robot.

Le modèle dynamique inverse (Eq. 2.8a ) se simplifie donc comme suit :

$$\tau = Y(\ddot{q}, \dot{q}, q)\chi \quad (2.12)$$

où :

- $Y$  matrice ( $n \times r$ ) des fonctions connues.
- $\chi$  vecteur ( $r \times 1$ ) des paramètres de base.

### 2.3.7 Modèle de frottement

Le frottement est présent dans tous les systèmes mécaniques. Cette propriété peut être souhaitable, notamment en cas de freinage, mais elle constitue également un réel problème pour les robots et les systèmes pneumatiques et hydrauliques. Il peut atteindre entre 10 et 30 % de la force nominale de fonctionnement d'un robot. En conséquence, il faut en tenir compte dans notre modélisation dynamique.

De nombreux travaux de recherche et des essais expérimentaux ont été réalisés sur la modélisation des frottements au niveau des axes et des moteurs du robot. Les caractéristiques de ces frottements sont décrites par de multiples modèles mathématiques tels le modèle LuGre présenté dans [31] un modèle qui permet de mesurer la plupart des frottements observés expérimentalement dont l'effet Stribeck, les caractéristiques de frottement des ressorts. Également le modèle étudié dans [23] proposé pour les systèmes mécaniques utilisant une charge sur le moteur, tout comme le modèle Dahl [40] qui simule le frottement de glissement, y compris le frottement de Coulomb et de stiction et le frottement de roulement à forte percussion, ou encore le modèle de Bliman et Sorine [70]

Dans le cadre de notre thèse, afin d'établir le modèle dynamique du robot, nous avons retenu un modèle classique (voir Figure 2.10) où le frottement de Coulomb est combiné avec le frottement visqueux. Ce modèle comprend donc deux coefficients, Coulomb et frottement visqueux, et est suffisamment puissant pour les vitesses utilisées sur le robot et pratiquées par différentes études dynamiques. Au modèle s'ajoute un terme de décalage  $Off_j$  qui prend en compte le décalage de l'amplificateur de puissance qui alimente le moteur articulé et l'asymétrie du frottement de Coulomb. On peut donc l'exprimer comme suit :

$$\tau_{f_i} = F_{c_i} \text{sign}(\dot{q}_i) + F_{v_j}(\dot{q}_i) + Off_i \quad (2.13)$$

avec la fonction  $\text{sign}()$  de la vitesse qui est définie comme suit :

$$\text{sign}(\dot{q}) = \begin{cases} 1 & \text{pour } \dot{q} > 0 \\ 0 & \text{pour } \dot{q} = 0 \\ -1 & \text{pour } \dot{q} < 0 \end{cases}$$

et

$\tau_{f_i}$  : le vecteur des efforts de frottement.

$F_{v_j}$  : les matrices des coefficients de frottement visqueux de l'articulation  $i$ .

$F_{c_i}$  : les matrices de frottement de Coulomb de l'articulation  $i$ .

$Off_j$  : l'offset de l'amplificateur de puissance de l'articulation  $i$ .

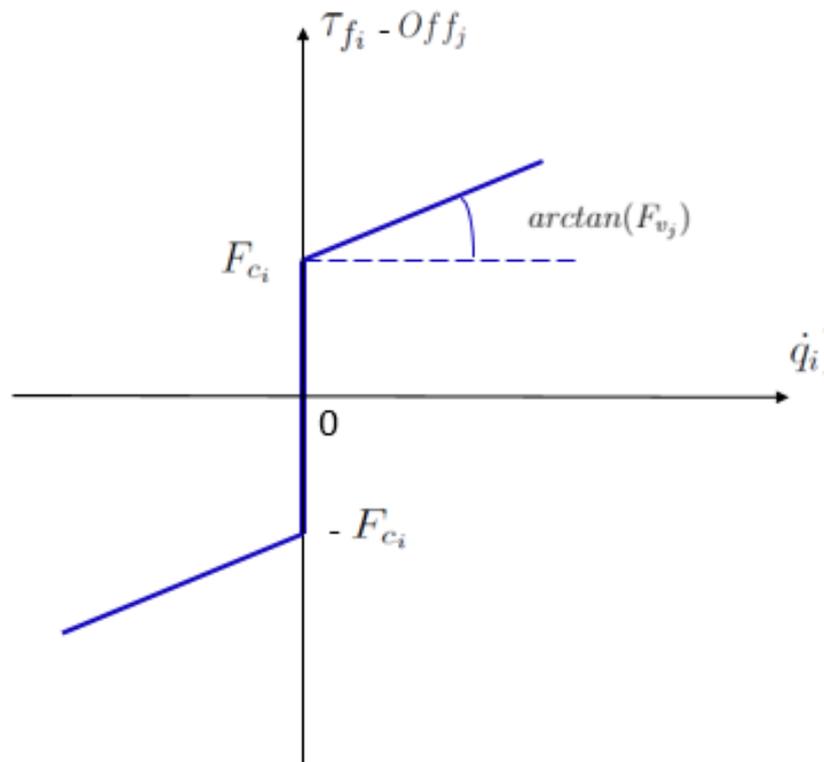


FIGURE 2.10 – Modèle de frottement de Coulomb et frottement visqueux

## 2.4 Conclusion

Dans ce chapitre, nous avons développé tous les éléments de base pour les prochaines étapes. La première étape a consisté à aborder quelques notions de base de la modélisation des robots, et plus particulièrement des robots manipulateurs (tels que la redondance, les transformations homogènes...). Pour la modélisation de notre robot, nous avons choisi la méthode Denavit-Hartenberg modifiée. Celle-ci a été présentée en détail tout juste avant la présentation de notre robot, le robot YuMi IRB14000, dont toutes les simulations et les travaux expérimentaux de cette thèse ont été réalisés.

Afin d'identifier ou de contrôler le robot, il était manifestement impératif de le modéliser. Nous avons donc présenté et calculé la modélisation géométrique directe et inverse du robot ; des hypothèses ont été prises en compte dans ce dernier.

Cette modélisation a été validée par trois méthodes différentes (par la modélisation théorique, par la Toolbox Robotic de Matlab ainsi que par le modèle fourni par le logiciel Symoro. En outre, ce dernier est un logiciel extrêmement pertinent en robotique qui a également été brièvement décrit dans ce chapitre.

En ce qui concerne la modélisation cinématique, nous l'avons calculée en déterminant la jacobienne et la pseudo-inverse de la jacobienne du robot. La dynamique étant une discipline essentielle pour modéliser le comportement des robots, nous avons réussi à établir symboliquement le modèle dynamique en prenant en compte le modèle de frottement, tout comme le modèle linéaire par rapport aux paramètres de base.

Dans le prochain chapitre, nous devons désormais passer à l'identification de ces paramètres dynamiques de base.

# Chapitre 3

## Identification dynamique du robot

Dans ce chapitre, nous présentons la méthode d'identification des paramètres dynamiques du robot, dans la nécessité de simuler le comportement réel du robot et surtout de pouvoir déterminer les forces externes appliquées sur le robot pour lequel on veut effectuer la commande force/position. Pour ce faire, on commence par décrire toutes les étapes envisagées par la méthode d'identification choisie. Par la suite, nous présentons les résultats expérimentaux issus de cette méthode. La détermination des forces externes appliquées au robot est également décrite dans ce chapitre.

### 3.1 Introduction

Un robot est caractérisé par une structure et par des paramètres dynamiques. Ces valeurs de différents paramètres ne sont pas toujours communiqués par le constructeur [136] , [148]. En outre, les paramètres obtenus à partir du modèle de conception assistée par ordinateur (CAO) ne sont pas précis du fait du risque de variation que les robots du même modèle peuvent avoir en phase de fabrication ; ce risque est très important, et entraîne une incertitude quant à ses éléments. Cette incertitude peut être classée en deux catégories [74] : une incertitude structurelle liée à l'identification de ses paramètres ; ainsi qu'une incertitude non structurelle fortement liée au modèle du robot, comme le modèle de frottement et de bruit, l'amortissement des articulations etc. Par ailleurs, pour commander un robot industriel avec précision, il est primordial de pouvoir estimer ses efforts, ce qui suppose une identification précise des paramètres dynamiques du robot afin d'établir un modèle dynamique fiable et de réaliser une commande de haute précision.

De nombreuses recherches se sont penchées sur l'identification des paramètres dynamiques des robots manipulateurs. Les techniques de cette identification des paramètres peuvent être classées en méthodes hors-ligne et en ligne [88]. Les méthodes hors-ligne sont basées sur la collecte de toutes les données d'entrée-sortie, le pré-traitement des données et la mise en œuvre de l'algorithme d'identification. Les techniques d'identification en ligne mettent à jour les paramètres estimés en temps réel en utilisant les données mesurées en ligne pendant le fonctionnement du robot. Dans les deux cas, les signaux d'entrée-sortie du robot sont enregistrés pendant que le robot suit les trajectoires d'excitantes, généralement faite par mise en œuvre d'un contrôleur robuste en boucle fermée. Ensuite, un système linéaire surdéterminé déterminé est créé et résolu à l'aide de méthodes d'optimisation numériques telles que la méthode des moindres carrés [60] , la méthode des moindres carrés pondérés [85] ou la méthode du maximum de vraisemblance [133], très utilisées en industrie, néanmoins, il ne devrait pas être envisagé dans les cas où les deux mesures, la position et le couple sont bruités.

Depuis les années 80, le Filtre de Kalman Étendu (FKE) est utilisé pour les systèmes à peu de paramètres [71]. L'algorithme estime à la fois l'état du système et les paramètres en utilisant le modèle dynamique direct. En revanche, il est sensible aux conditions initiales et sa vitesse de convergence est très lente [62] [116]. Dans [57], l'identification de modèles paramétriques linéaires à temps continu a été abordée.

En robotique, la méthode basée sur le modèle dynamique inverse linéaire par rapport aux paramètres dynamiques et les moindres carrés IDIM-LS (Inverse Dynamic Identification Model with Least Squares) est généralement adoptée. En effet, cette approche a été retenue avec beaucoup de succès pour l'identification expérimentale de robots dans [61] [81] et représente une étape intermédiaire pour d'autres approches en ligne ou hors-ligne [59][63]. La méthode de l'erreur de sortie (CLOE) compare le comportement du système réel avec le modèle mathématique. Ce modèle est modifié de manière itérative pour optimiser les paramètres, afin de rapprocher le comportement du modèle de celui du robot réel. Le critère souvent utilisé est la norme quadratique de l'erreur de sortie. Cette méthode peut être mise en œuvre en boucle ouverte ou en boucle fermée [98] [32].

Dans [59], Gautier et al. ont proposé une méthode d'erreur de sortie en boucle fermée basée sur une simulation en boucle fermée du robot, en minimisant une erreur quadratique entre les couples d'articulation réels et simulés en utilisant le modèle dynamique direct et inverse du robot appelé DIDIM (Direct and Inverse Dynamic Identification Model). Cette nouvelle méthode a été appliquée avec succès aux robots manipulateurs rigides. Une comparaison entre DIDIM et CLOE a été effectuée par [82] et une extension de cette méthode a été validée pour les robots à articulations flexibles en [87].

Dans [154], une nouvelle procédure d'identification des paramètres dynamiques des robots à articulations élastiques a été proposée. Cette approche est basée sur la formule lagrangienne du modèle dynamique du manipulateur, tenant compte de l'énergie élastique du robot, en plus des contributions d'énergie cinétique et de gravité. La référence [74] propose aussi une autre méthode d'identification des paramètres dynamiques des robots industriels à articulations élastiques dont la rigidité de l'articulation dans le modèle dynamique est examiné et la précision de l'identification est améliorée.

Janot et al. proposent dans [83] une approche basée sur la méthode des paramètres dépendant de l'état SDP (State-Dependent Parameter) pour l'estimation des modèles non linéaires. La présente méthode SDP est une procédure d'identification en deux étapes capable d'identifier la présence et la forme graphique des non-linéarités dans les modèles de systèmes dynamiques. Les performances de cette méthode ont été également comparées à celles de IDIM-LS.

Grâce à cette revue de la littérature et la référence [88], nous avons constaté que la méthode utilisant le modèle inverse et les moindres carrés reste un très bon choix pour identifier les paramètres d'un manipulateur de robot rigide. Par conséquent, nous avons choisi d'adopter cette méthode pour identifier les paramètres de notre robot et ensuite construire un simulateur rigide, à partir duquel nous allons élaborer notre loi de commande. En effet, le but de ce simulateur est de nous permettre de tester notre loi de commande avant de l'appliquer au robot pour éviter de l'endommager.

## 3.2 Définition de la méthode d'identification

### 3.2.1 Identification par MDI, moindres carrés et optimisation

Au vu de la littérature, la méthode utilisant le MDI et les moindres carrés se révèle être une méthode robuste pour identifier les paramètres dynamiques d'un robot à articulations rigides. Elle a été adoptée et validée par différentes recherches. Ces dernières ont montré qu'en respectant toutes ses étapes, l'identification ne devrait plus être un véritable verrou. En revanche, dans notre cas, nous avons obtenu des résultats non conformes à la physique; nous avons donc proposé d'ajouter une étape d'optimisation dans laquelle nous prenons en compte l'aspect physique du robot tel que les paramètres de frottement obligatoirement positifs ou encore la positivité de la matrice d'inertie [135] [136]. La description des différentes étapes de la méthode est présentée dans le schéma suivant (Figure 3.1).

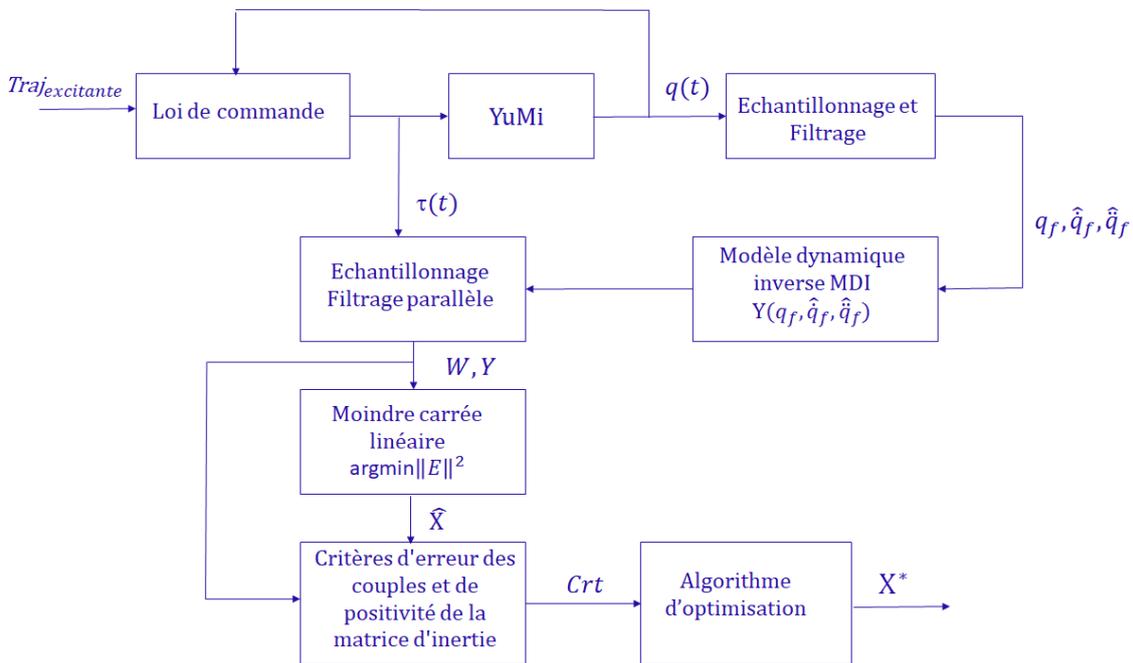


FIGURE 3.1 – Schéma du principe de la méthode d'identification

#### Première étape : Filtrage de positions et calcul de vitesses et accélérations

A partir de la simple mesure des positions des articulations, les valeurs de vitesse et d'accélération sont estimées. En effet, on commence par échantillonner les mesures, ensuite on applique une technique de filtrage numérique non linéaire pour éliminer les valeurs aberrantes; ce filtrage est effectué par un filtre médian. On établit ensuite le calcul à partir d'un filtre dérivateur, produit d'un filtre dérivateur de type différence centrée (Eq. 3.1), d'un filtre médian puis d'un filtre passe-bas de type moyen 5 aller-retour. La dérivée centrée des positions est utilisée pour les vitesses et la dérivée centrée des vitesses pour les accélérations.

$$\hat{q}_e = \frac{q_{e+1} - q_{e-1}}{2T_e} \quad \hat{\dot{q}}_e = \frac{\hat{q}_{e+1} - \hat{q}_{e-1}}{2T_e} \quad (3.1)$$

Avec :

- $q_e$  : la position échantillonnée à l'instant  $t_e$
- $\hat{q}_e$  : la vitesse estimée à l'instant  $t_e$
- $\hat{\ddot{q}}_e$  : l'accélération estimée à l'instant  $t_e$
- $T_e$  : la période d'échantillonnage

### Seconde étape : Calcul de la matrice d'identification par MDI.

À cette étape, on calcule à partir des positions, des vitesses et des accélérations mesurées et calculées la matrice d'identification  $w(\hat{q}, \hat{\dot{q}}, q)$  par l'équation linéaire du MDI (Eq. 2.12). Par conséquent, on obtient un couple modèle, distinct du couple réel du robot à cause, des erreurs de bruit et de mesure dénommées  $\epsilon$ ; un bruit blanc additif  $\epsilon \in (0, \sigma^2)$ ). Ce dernier s'écrit alors sous la forme générale suivante :

$$\tau = w(\hat{q}, \hat{\dot{q}}, q)\chi + \epsilon \quad (3.2)$$

### Troisième étape : Mesure des couples réels du robot.

Pour le plus grand nombre de robots, le contrôleur ne fournit que les consignes des boucles de courant des amplificateurs de puissance qui actionnent les moteurs. Toutefois, ces boucles sont connues par une forte dynamique par rapport à la dynamique mécanique du robot, ce qui implique un gain d'actionnement entre les deux. Ces gains sont donnés par le constructeur, et peuvent également être identifiés [58].

$$\tau_m = g_r I_r = N_r G k_t I_r \quad (3.3)$$

avec :

- $\tau_m$  : l'effort moteur de l'articulation
- $I_r$  : la consigne de courant de l'articulation
- $g_r$  : le gain d'actionnement de l'articulation
- $N_r$  : le coefficient de réduction de la transmission mécanique
- $G$  : le gain statique de la boucle de courant
- $k_t$  : la constante de couple électromagnétique

### Quatrième étape : Filtrage parallèle

A ce stade, on échantillonne le système (Eq. 3.2) à une fréquence de mesure, ensuite on regroupe ces échantillons axe par axe, de manière à construire un système surdéterminé (Eq. 3.4). En conséquence, la matrice  $Y$  et le vecteur  $W$  sont construits à partir de données biaisées du fait que les acquisitions sont expérimentales et que les estimations de vitesse et d'accélération sont bruitées. Ainsi, l'estimation  $\hat{\chi}$  peut présenter un biais.

On applique donc un filtrage parallèle pour les deux variables par un filtre passe-bas de fréquence

$\omega_p$  de type Butterworth aller-retour d'ordre 5. Cela ne remet pas en cause la solution des moindres carrés car la perturbation apportée par le filtre est identique dans tous les termes du système linéaire.

$$Y = W\chi + \rho \quad (3.4)$$

avec :

$Y$  : le vecteur ( $s \times 1$ ) de mesure de couples

$W$  : la matrice d'observation ( $s \times d$ )

$\rho$  : le vecteur ( $s \times 1$ ) des résidus

$s$  : le nombre de mesures

$d$  : le nombre de paramètres à identifier

$$Y = \begin{bmatrix} \begin{bmatrix} \tau_1(t_1) \\ \tau_1(t_2) \\ \vdots \\ \tau_1(t_e) \end{bmatrix} \\ \begin{bmatrix} \tau_2(t_1) \\ \tau_2(t_2) \\ \vdots \\ \tau_2(t_e) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \tau_n(t_1) \\ \tau_n(t_2) \\ \vdots \\ \tau_n(t_e) \end{bmatrix} \end{bmatrix} \quad W = \begin{bmatrix} \begin{bmatrix} w_1(\hat{q}(t_1), \dot{\hat{q}}(t_1), q(t_1)) \\ w_1(\hat{q}(t_2), \dot{\hat{q}}(t_2), q(t_2)) \\ \vdots \\ w_1(\hat{q}(t_e), \dot{\hat{q}}(t_e), q(t_e)) \end{bmatrix} \\ \begin{bmatrix} w_2(\hat{q}(t_1), \dot{\hat{q}}(t_1), q(t_1)) \\ w_2(\hat{q}(t_2), \dot{\hat{q}}(t_2), q(t_2)) \\ \vdots \\ w_2(\hat{q}(t_e), \dot{\hat{q}}(t_e), q(t_e)) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} w_n(\hat{q}(t_1), \dot{\hat{q}}(t_1), q(t_1)) \\ w_n(\hat{q}(t_2), \dot{\hat{q}}(t_2), q(t_2)) \\ \vdots \\ w_n(\hat{q}(t_e), \dot{\hat{q}}(t_e), q(t_e)) \end{bmatrix} \end{bmatrix}$$

*NB : La matrice d'observation du robot peut ne pas être de plein rang : on peut en effet considérer que le modèle utilisé pour l'identification n'est pas suffisamment simplifié, à savoir qu'il existe des paramètres dynamiques qui n'ont pas d'influence directe sur le système. Pour surmonter ce problème, des méthodes algébriques ou numériques peuvent être utilisées pour avoir un jeu de paramètres identifiables. Ou bien, la trajectoire d'excitation utilisée pour l'identification n'est pas assez riche en fréquences/amplitudes pour exciter tous les paramètres du système. Dans ce cas, il est indispensable de recalculer la trajectoire d'excitation.*

### Cinquième étape : Moindre carrée

La solution unique du système surdéterminé (Eq. 3.4) est les moindres carrés ordinaires en minimisant  $\rho$  :

$$\hat{\chi} = \operatorname{argmin}_{\chi} \|\rho\|^2 = W^+ Y \quad (3.5)$$

avec :

$$W^+ = (W^T W)^{-1} W^T \text{ pseudo inverse de } W$$

$\rho$  représente le vecteur des résidus liés au bruit de mesure et aux erreurs de modélisation. On suppose en principe que ce bruit est un bruit blanc de moyenne nulle et d'écart-type  $\sigma_\rho$ . Sa matrice de variance-covariance  $C_\rho$  s'écrit sous la forme suivante :

$$C_\rho = E(\rho\rho^T) = \sigma_\rho^2 I_{nr} \quad (3.6)$$

avec :

- $E$  : l'espérance mathématique
- $I_{nr}$  : la matrice d'identité de dimension  $(s \times s)$ .

On obtient l'écart-type à partir de la relation suivante :

$$\sigma_\rho^2 = \frac{\|Y - W\hat{\chi}\|^2}{s - d} \quad (3.7)$$

avec cette fois ci :

- $s$  : est le nombre de lignes de la matrice d'observation  $W$
- $d$  : est le nombre de colonnes de la matrice d'observation  $W$

La matrice de variance-covariance de l'erreur d'estimation s'écrit :

$$C_\chi = E[(\chi - \hat{\chi})(\chi - \hat{\chi})^T] = W^+ C_\rho (W^+)^T = \sigma_\rho^2 (W^T W)^{-1} \quad (3.8)$$

L'écart-type du  $i^{me}$  paramètre s'écrit comme suit :

$$\sigma_{\chi_i}^2 = C_{\chi\chi}(i, i) \quad (3.9)$$

L'écart-type relatif permet de vérifier la pertinence de l'estimation des paramètres. En principe, lorsque les écarts types relatifs de tous les paramètres identifiés sont inférieurs ou égaux à 1%, il convient d'estimer que tous les paramètres ont été correctement identifiés.

L'écart-type relatif  $\sigma_{\chi_{ir}}(\%)$  est calculé comme suit :

$$\sigma_{\chi_{ir}}(\%) = 100 \cdot \frac{\sigma_{\chi_i}}{\hat{\chi}_i} \quad \text{pour} \quad |\hat{\chi}_i| \neq 0 \quad (3.10)$$

Certains paramètres ne contribuent pas ou ont une contribution faible dans la dynamique du robot. Ils sont très faibles et leurs écart-types relatifs sont très importants. Dans ce cas, une élimination de ces derniers est considérée.

### Dernière étape : Optimisation

Les premiers résultats sur le robot ont révélé que le vecteur identifié par la méthode des moindres carrés a des valeurs non réalistes, autrement dit non cohérentes avec le sens physique du paramètre. Une étape d'optimisation numérique a donc été ajoutée. Le principe de cette optimisation est basé sur l'estimation des paramètres en minimisant un ensemble de critères.

$$\hat{\chi} = \operatorname{argmin}_{\chi} \|(Y - W\chi)\|^2 \quad (3.11)$$

$$\text{tel que } \begin{cases} M(q, \hat{\chi}) > 0 & \forall q \in \mathbb{R}^n \\ Fs_i > 0, \quad Fv_i > 0 & i = 1, \dots, n \quad \forall q \in \mathbb{R}^n \\ Ia_i > 0 & i = 1, \dots, n \quad \forall q \in \mathbb{R}^n \end{cases}$$

La problématique retenue (Eq. 3.11) consiste à minimiser le carré de la norme de l'erreur entre les couples articulaires mesurés et calculés, sous la contrainte de maintenir en permanence la matrice d'inertie du robot  $M(q)$  positive, les valeurs d'inertie des moteurs et celles des paramètres de frottement visqueux et secs devant être également positives. Pour vérifier la première contrainte, plusieurs méthodes existent.

Nous avons opté pour la démarche suivante : Dans un premier temps, nous vérifions la positivité des valeurs propres de la matrice d'inertie le long de la trajectoire d'identification, une condition suffisante pour avoir une matrice d'inertie définie positive. Dans un second temps, en utilisant les paramètres optimaux estimés lors de la première étape, nous vérifions sur tout l'espace de travail la validité du critère en modifiant la trajectoire du robot, et donc en ayant des valeurs propres positives, quelle que soit la configuration du robot. Conformément à [151], les contraintes sur les paramètres physiques des articulations du robot, principalement la masse et les tenseurs d'inertie, qui aboutissent à une matrice d'inertie définie positive, sont les suivantes :

$$\begin{cases} m_i > 0, & i = 1, \dots, n \\ {}^i I_{ctm_i} > 0, & {}^i I_{ctm_i} = \begin{bmatrix} xx_{ctm_i} & xy_{ctm_i} & xz_{ctm_i} \\ xy_{ctm_i} & yy_{ctm_i} & yz_{ctm_i} \\ xz_{ctm_i} & yz_{ctm_i} & zz_{ctm_i} \end{bmatrix} \end{cases}$$

où :

$m_i$  : la masse du corps  $i$

${}^i I_{ctm_i}$  : la matrice du tenseur d'inertie autour du centre de masse  $ctm_i$  du corps  $i$

La relation entre  ${}^i I_i$  le tenseur d'inertie autour de  $O_i$  et  ${}^i I_{ctm_i}$  est donnée par l'équation suivante :

$${}^i I_i = {}^i I_{ctm_i} + m_i ({}^i r_i^T r_i I_{33} - {}^i r_i {}^i r_i^T) = \begin{bmatrix} XX_i & XY_i & XZ_i \\ XY_i & YY_i & ZZ_i \\ XZ_i & YZ_i & ZZ_i \end{bmatrix}$$

avec :

${}^i I_i$  : le tenseur d'inertie autour de  $O_i$

${}^i r_i = [x_i \ y_i \ z_i]^T$  : le vecteur entre le point  $O_i$  et le centre de masse  $ctm_i$  de l'articulation  $i$

$I_{33}$  : la matrice identité ( $3 \times 3$ )

L'implémentation de l'optimisation numérique a été faite avec la fonction "fminsearch" du logiciel MATLAB.

### 3.2.2 Précision des paramètres estimés

La précision des paramètres identifiés est un critère primordial pour optimiser le modèle développé du robot. Néanmoins, tous les paramètres ne sont pas estimés avec la même précision [35]. Nous proposons dans cette section une méthode pour examiner ces différents paramètres.

La valeur de  $P_\chi$  représente la précision des paramètres identifiés par rapport aux couples. Nous utilisons une méthode classique basée sur des propriétés statistiques. Cette méthode a été également utilisée dans [24] pour estimer la précision des paramètres mécaniques pour une platine de micro-positionnement utilisant un modèle d'hystérésis de vitesse.

$$P_\chi = 100 \frac{(\text{diag } \sigma^2)^{0,5}}{\hat{\chi}} \quad (3.12)$$

Avec :

$$\sigma^2 = \frac{CRT}{s - d} \quad (3.13)$$

- $s$  : le nombre d'échantillons de mesure
- $d$  : le nombre de paramètres estimés
- $\sigma^2$  : la variance du bruit
- $CRT$  : un critère retenu pour le calcul de la précision [24]

Pour calculer le critère  $CTR$ , on considère que le modèle de robot basé sur les paramètres estimés a le même comportement que le robot réel.

$$\tau(k) = \hat{\tau}(k, \hat{\chi}) + \epsilon(k) \quad (3.14)$$

L'erreur  $e$  entre le vecteur des paramètres estimés  $\hat{\chi}$  et la valeur optimale  $\chi^*$  ainsi que le critère de convergence sont calculés par :

$$e = \hat{\chi} - \chi^* \quad (3.15)$$

$$C = \sum_{i=1}^n (\tau(k) - \hat{\tau}(k, \chi^*))^2 = \sum_{k=1}^n \left[ \hat{\tau}(\hat{\chi}) + \epsilon - \hat{\tau}(\hat{\chi}) - \frac{\partial \hat{\tau}^T}{\partial \chi}(\hat{\chi}) e \right]^2 \quad (3.16)$$

Avec le vecteur de sensibilité  $s = \frac{\partial \hat{\tau}}{\partial \chi}(\hat{\chi})$  l'équation (Eq. 3.16) devient :

$$C = \sum_{k=1}^n [\epsilon - s^T e]^2 = \sum_{k=1}^n [\epsilon^2 - 2\epsilon s^T e + e^T s s^T e] \quad (3.17)$$

On pose :

$$B = \sum_{k=1}^n s s^T$$

$$\beta^T = \sum_{k=1}^n \epsilon s$$

$$e = B^{-1}\beta$$

On remplace chaque variable dans l'équation 3.17

$$C = \sum_{k=1}^n \epsilon^2 - 2\beta^T e + e^T B e \quad (3.18)$$

La valeur optimale de  $\chi^*$  est obtenue pour  $\frac{\partial C}{\partial \chi} = 0$ . La variance de l'estimaion et sa précision dépendent de la puissance du bruit et des fonctions de sensibilité du modèle. A partir de travail réalisé dans [24], le critère utilisé pour le calcul de la précision des paramètres identifiés est le suivant :

$$CRT = \sum_{k=1}^n [\epsilon^2 - \beta e] \quad (3.19)$$

### 3.2.3 Trajectoires excitantes

Une trajectoire riche en fréquences et en amplitudes permet une bonne excitation des paramètres et donc une bonne identification. C'est ainsi que dans [19] des trajectoires dynamiques ont été conçues pour atteindre des vitesses et des accélérations maximales physiquement réalisables afin d'identifier le couple de frottement associé à un bras hydraulique. Les trajectoires à faible vitesse utilisant le maximum de débattement angulaire excitent les paramètres de gravité et de frottement sec. Pour l'excitation des paramètres d'inertie, des trajectoires sinusoïdales avec différentes amplitudes et fréquences en position sont appliquées dans [81], ainsi que l'excitation de la flexibilité se fait par des positions sinusoïdales à petites amplitudes. D'après [118], des trajectoires avec des B-splines optimisées sur des points choisis aléatoirement sont réalisées sur un robot 7 DDL de type Light Weight avec une optimisation non-linéaire pour des critères différents.

On distingue une première méthode dans laquelle les paramètres sont identifiés par groupe en sollicitant une partie des articulations et en bloquant les autres, appelée identification séquentielle. Ce type d'identification introduit une accumulation d'erreur entre les groupes de paramètres déterminés et ceux préalablement identifiés [93]. Pour estimer les coefficients de frottement des articulations de manière séparée, cette méthode d'identification a été utilisée dans [39]. Plus tard, cette même méthode a été étendue pour permettre l'identification des paramètres inertiels [69].

La seconde méthode est basée sur une optimisation non-linéaires des différents critères. En effet, une trajectoire dite excitante, et une trajectoire ayant pour conséquence un bon conditionnement de la matrice d'observation  $W$ .

L'un des principaux critères à minimiser pour avoir une trajectoire suffisamment existante est :

$$\text{Cond}(W^T W) = \frac{\sigma_{max}}{\sigma_{min}} \quad (3.20)$$

Avec :

- $\sigma_{min}$  : les valeurs singulières minimale de la matrice  $W$
- $\sigma_{max}$  : les valeurs singulières maximale de la matrice  $W$  [77]

Il y a aussi le critère (Eq. 3.21) permettant de garantir une faible incertitude de l'estimation des paramètres dynamiques [117], un autre avec un bon rapport signal/bruit dans  $Y$  de plus que le conditionnement de  $W$  (Eq. 3.22) ou encore un critère permettant de calibrer les écart-types relatifs aux paramètres (Eq. 3.23) [133].

$$-\log(\det(C)) \quad (3.21)$$

$$\text{Cond}(W) + \lambda \frac{1}{\sigma_{\min}(W)} \quad (3.22)$$

$$\text{Cond}(W \text{diag}(Z)) \quad (3.23)$$

où :

- $C$  : la matrice de covariance
- $Z$  : la matrice diagonale représente un vecteur des connaissances sur les paramètres dynamiques détaillé dans [133]

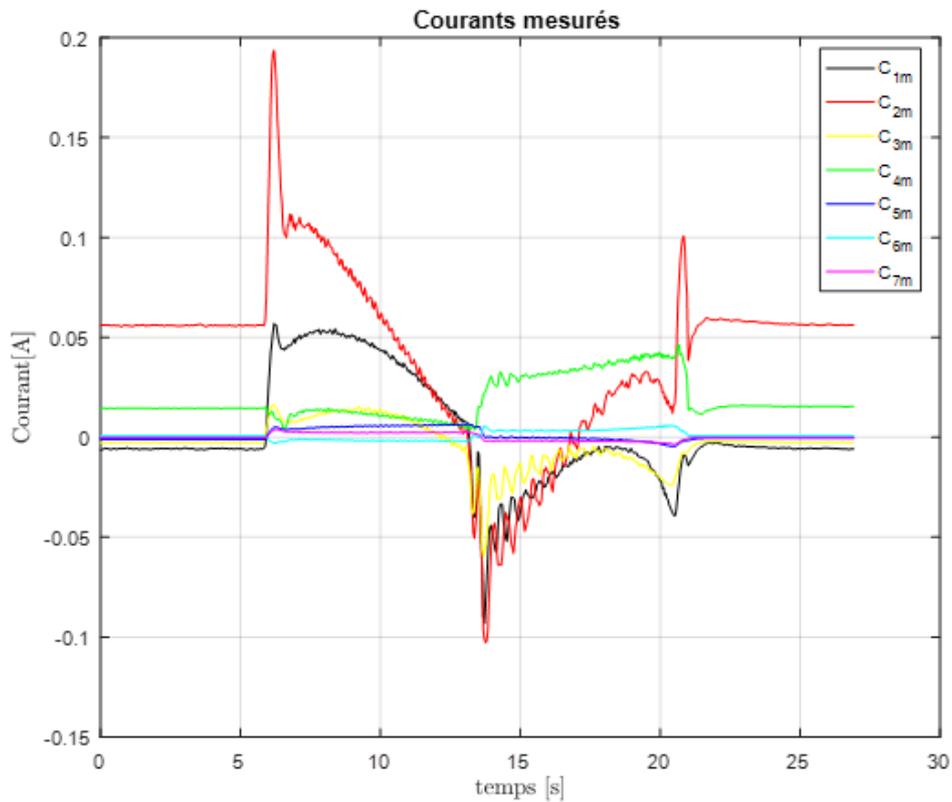


FIGURE 3.2 – Courants mesurés lors du suivi de la trajectoire excitante

### 3.3 Résultats expérimentaux d'identification

Comme évoqué dans le chapitre précédent (Chap. 2), le modèle dynamique du robot est linéaire par rapport au paramètre de bases du robot. Pour ce qui est de notre robot, les sept articulations ont chacune 14 paramètres standard, dont onze paramètres dynamiques, deux paramètres de frottement sec et visqueux ainsi qu'un paramètre d'offset. Cela donne un total de 98 valeurs de paramètres standard à identifier. Sur ce total, 21 sont des paramètres qui ne peuvent pas être regroupés : il s'agit des paramètres de frottement et des offsets.

TABLE 3.1 – *Regroupement des paramètres du bras du robot YuMi*

j	XX	XY	XZ	YY	YZ	ZZ	MZ	MY	MZ	M	IA
1	0	0	0	0	0	ZZ <sub>1R</sub>	MX <sub>1R</sub>	MY <sub>1R</sub>	0	0	0
2	XX <sub>2R</sub>	XY <sub>2R</sub>	XZ <sub>2R</sub>	0	YZ <sub>2R</sub>	ZZ <sub>2R</sub>	MX <sub>2R</sub>	MY <sub>2R</sub>	0	0	IA <sub>2</sub>
3	XX <sub>3R</sub>	XY <sub>3R</sub>	XZ <sub>3R</sub>	0	YZ <sub>3R</sub>	ZZ <sub>3R</sub>	MX <sub>3R</sub>	MY <sub>3R</sub>	0	0	IA <sub>3</sub>
4	XX <sub>4R</sub>	XY <sub>4R</sub>	XZ <sub>4R</sub>	0	YZ <sub>4R</sub>	ZZ <sub>4R</sub>	MX <sub>4R</sub>	MY <sub>4R</sub>	0	0	IA <sub>4</sub>
5	XX <sub>5R</sub>	XY <sub>5R</sub>	XZ <sub>5R</sub>	0	YZ <sub>5R</sub>	ZZ <sub>5R</sub>	MX <sub>5R</sub>	MY <sub>5R</sub>	0	0	IA <sub>5</sub>
6	XX <sub>6R</sub>	XY <sub>6R</sub>	XZ <sub>6R</sub>	0	YZ <sub>6R</sub>	ZZ <sub>6R</sub>	MX <sub>6R</sub>	MY <sub>6R</sub>	0	0	IA <sub>6</sub>
7	XX <sub>7R</sub>	XY <sub>7</sub>	XZ <sub>7</sub>	0	YZ <sub>7</sub>	ZZ <sub>7</sub>	MX <sub>7</sub>	MY <sub>7</sub>	0	0	IA <sub>7</sub>

En utilisant le logiciel Symoro+, nous avons passé de 77 paramètres inertiels à 51 paramètres dynamiques que nous devons identifier (Table 3.1). En tout, 72 paramètres sont désormais identifiables au lieu de 98. En effet, la lettre "R" est ajoutée à la fin des termes groupés. Cette réduction du nombre de paramètres est requise, car il est impossible d'identifier certains paramètres.

Les différentes fonctions de regroupement ci-dessous représentent l'ensemble des paramètres inertiels susceptibles d'être regroupés en fonction des paramètres inertiels de base propres à chaque axe :

**Axe 1 :**

$$ZZ_{1R} = IA_1 + d_3^2(M_3 + M_4 + M_5 + M_6 + M_7) + 2.MZ_2r_2 + (M_2 + M_3 + M_4 + M_5 + M_6 + M_7)(d_2^2 + r_2^2) + YY_2 + ZZ_1$$

**Axe 2 :**

$$\left\{ \begin{array}{l} XX_{2R} = d_4^2(M_4 + M_5 + M_6 + M_7) - d_3^2(M_3 + M_4 + M_5 + M_6 + M_7) + 2MZ_3r_3 + (M_3 + M_4 + M_5 + M_6 + M_7)r_3^2 + XX_2 - YY_2 + YY_3 \\ XY_{2R} = d_3MZ_3 + d_3(M_3 + M_4 + M_5 + M_6 + M_7)r_3 + XY_2 \\ ZZ_{2R} = IA_2 + d_4^2(M_4 + M_5 + M_6 + M_7) + 2MZ_3r_3 + (M_3 + M_4 + M_5 + M_6 + M_7)(d_3^2 + r_3^2) + YY_3 + ZZ_2 \\ MX_{2R} = d_3(M_3 + M_4 + M_5 + M_6 + M_7) + MX_2 \\ MY_{2R} = MY_2 - MZ_3 - (M_3 + M_4 + M_5 + M_6 + M_7)r_3 \end{array} \right.$$

**Axe 3 :**

$$\left\{ \begin{array}{l} XX_{3R} = d_5^2(M_5 + M_6 + M_7) - d_4^2(M_4 + M_5 + M_6 + M_7) + 2MZ_4r_4 + (M_4 + M_5 + M_6 + M_7)r_4^2 + XX_3 - YY_3 + YY_4 \\ XY_{3R} = -(d_4MZ_4) - d_4(M_4 + M_5 + M_6 + M_7)r_4 + XY_3 \\ ZZ_{3R} = d_5^2.(M_5 + M_6 + M_7) + 2.MZ_4.r_4 + (M_4 + M_5 + M_6 + M_7).(d_4^2 + r_4^2) + YY_4 + ZZ_3 \\ MX_{3R} = d_4(M_4 + M_5 + M_6 + M_7) + MX_3 \\ MY_{3R} = MY_3 + MZ_4 + (M_4 + M_5 + M_6 + M_7)r_4 \end{array} \right.$$

**Axe 4 :**

$$\left\{ \begin{array}{l} XX_{4R} = d_6^2(M_6 + M_7) - d_5^2(M_5 + M_6 + M_7) + 2MZ_5r_5 + (M_5 + M_6 + M_7)r_5^2 + XX_4 - YY_4 + YY_5 \\ XY_{4R} = d_5MZ_5 + d_5(M_5 + M_6 + M_7)r_5 + XY_4 \\ ZZ_{4R} = d_6^2(M_6 + M_7) + 2MZ_5r_5 + (M_5 + M_6 + M_7)(d_5^2 + r_5^2) + YY_5 + ZZ_4 \\ MX_{4R} = d_5(M_5 + M_6 + M_7) + MX_4 \\ MY_{4R} = MY_4 - MZ_5 - (M_5 + M_6 + M_7)r_5 \end{array} \right.$$

**Axe 5 :**

$$\left\{ \begin{array}{l} XX_{5R} = d_7^2 M_7 - d_6^2 (M_6 + M_7) + 2MZ_6 r_6 + (M_6 + M_7)r_6^2 + XX_5 - YY_5 + YY_6 \\ XY_{5R} = -(d_6 M Z_6) - d_6 (M_6 + M_7)r_6 + XY_5 \\ ZZ_{5R} = d_7^2 \cdot M_7 + 2MZ_6 r_6 + (M_6 + M_7)(d_6^2 + r_6^2) + YY_6 + ZZ_5 \\ MX_{5R} = d_6 (M_6 + M_7) + MX_5 \\ MY_{5R} = MY_5 + MZ_6 + (M_6 + M_7)r_6 \end{array} \right.$$

**Axe 6 :**

$$\left\{ \begin{array}{l} XX_{6R} = -(d_7^2 M_7) + 2MZ_7 r_7 + M_7 r_7^2 + XX_6 - YY_6 + YY_7 \\ XY_{6R} = d_7 M Z_7 + d_7 M_7 r_7 + XY_6 \\ ZZ_{6R} = 2MZ_7 r_7 + M_7 (d_7^2 + r_7^2) + YY_7 + ZZ_6 \\ MX_{6R} = d_7 M_7 + MX_6 \\ MY_{6R} = MY_6 - MZ_7 - M_7 r_7 \end{array} \right.$$

**Axe 7 :**

$$XX_{7R} = XX_7 - YY_7$$

### 3.3.1 Mesure et filtrage des données

Les positions articulaires  $q$  et les couples de référence sont acquis à une fréquence de 250 Hz alors que le robot poursuit une trajectoire excitante. Les mesures ont été réalisées à l'aide d'un simulateur de robot RobotStudio<sup>1</sup> et du logiciel de mesure Test Signal Viewer<sup>2</sup>. Cette trajectoire a été définie afin d'assurer un bon conditionnement de la matrice d'observation  $W$ . Pour calculer la matrice d'observation  $W$ , les vitesses et accélérations articulaires sont estimées à l'aide d'un filtre dérivé à bande passante limitée. Ce filtre est appliqué après avoir supprimé les échantillons aberrants via un filtre médian d'ordre 5.

La figure (Figure 3.2) illustre les courants moteurs mesurés tout au long du suivi de la trajectoire choisie pour le robot.

1. *RobotStudio* : est un programme qui permet à l'utilisateur d'effectuer une programmation hors-ligne des systèmes de robots d'ABB. Il utilise le contrôleur virtuel d'ABB, qui est une implémentation logicielle ayant les mêmes fonctionnalités que le contrôleur du robot réel. Il est donc possible de simuler tous les mouvements du robot avant de télécharger un programme vers le processus réel [8].

2. *Test Signal Viewer* : est une application indépendante du fabricant ABB qui permet de lire les valeurs des signaux de mesure. L'application communique avec le contrôleur au moyen du protocole TCP/IP. Ce logiciel dispose de douze canaux de mesure auxquels il est possible d'attribuer n'importe quel signal de test. Les signaux de mesure pouvant être : les angles du mouvement des axes sélectionnés, signaux du système de commande de transmission et des freinages, etc. [109]

On obtient le système surdéterminé (Eq. 3.4) après un filtrage parallèle des couples  $Y$  mesurés expérimentalement et de chacune des colonnes de la matrice  $W$ . Nous avons utilisé pour cela un filtre passe-bas de Butterworth d'ordre 5 aller-retour.

En effet, les paramètres estimés  $\hat{\chi}$  sont la solution linéaire des moindres carrés de ce système surdéterminé. Ils ont été estimés à partir des paramètres de base définis par [91], et calculés automatiquement à l'aide du logiciel SYMORO +.

### 3.3.2 Résultats

Nous avons appliqué la méthode d'identification proposée à un seul bras du robot YuMi. Grâce à cette méthode, nous sommes parvenus à identifier 63 paramètres avec une précision remarquable. Ensuite, nous avons vérifié les contraintes d'optimisation pour l'ensemble de l'espace de travail du robot en intégrant toutes les limites de position des sept articulations. La figure (Figure 3.6) montre l'évolution des valeurs de la matrice d'inertie de YuMi. Sur la figure, on constate que toutes les valeurs propres restent positives quelle que soit la configuration du robot, preuve que l'optimisation a été une étape réussie de notre démarche d'identification.

Les paramètres de frottement et de décalage estimés pour les axes 1, 2, 3 et 4 ont été respectivement identifiés avec une grande précision, allant de 0,3% à 4%. Néanmoins, pour les axes 5, 6 et 7 (4, 5 et 6 selon la nomenclature du fabricant du robot), le degré d'imprécision est plus grand, mais reste acceptable et varie entre 3% et 23% (Table 3.2).

Cette même observation peut être faite pour les couples. En effet, les valeurs de l'erreur quadratique moyenne normalisée (NRMSE) (Eq. 3.24) entre les couples mesurés et estimés sont respectivement de 2,53%, 1,29%, 2,41% et 2,19% pour les axes 1, 2, 3 et 4, et de 8,53%, 7,84% et 7,64% pour les axes 5, 6 et 7 (selon l'ordre de numérotation des axes adopté). Les valeurs numériques des offsets identifiés ainsi que les valeurs des paramètres de frottement visqueux et de frottement sec sont présentées dans le tableau (Table 3.2) ; leurs précisions sont également indiquées.

$$NRMSE = 100 \times \sqrt{\sum_i \frac{(\tau_{fi} - \tau_{ei})^2}{\tau_{fi}^2}} \quad (3.24)$$

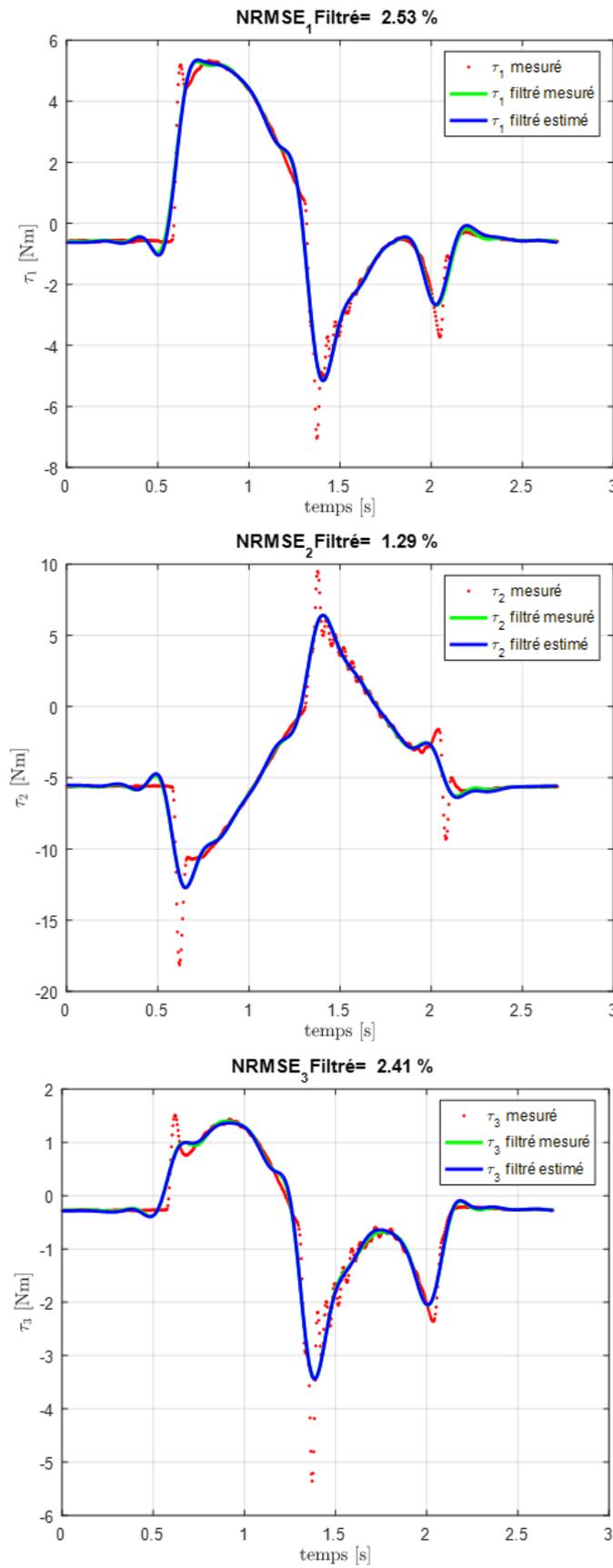


FIGURE 3.3 – Comparaison des couples mesurés, filtrés et estimés pour les articulations 1, 2 et 3

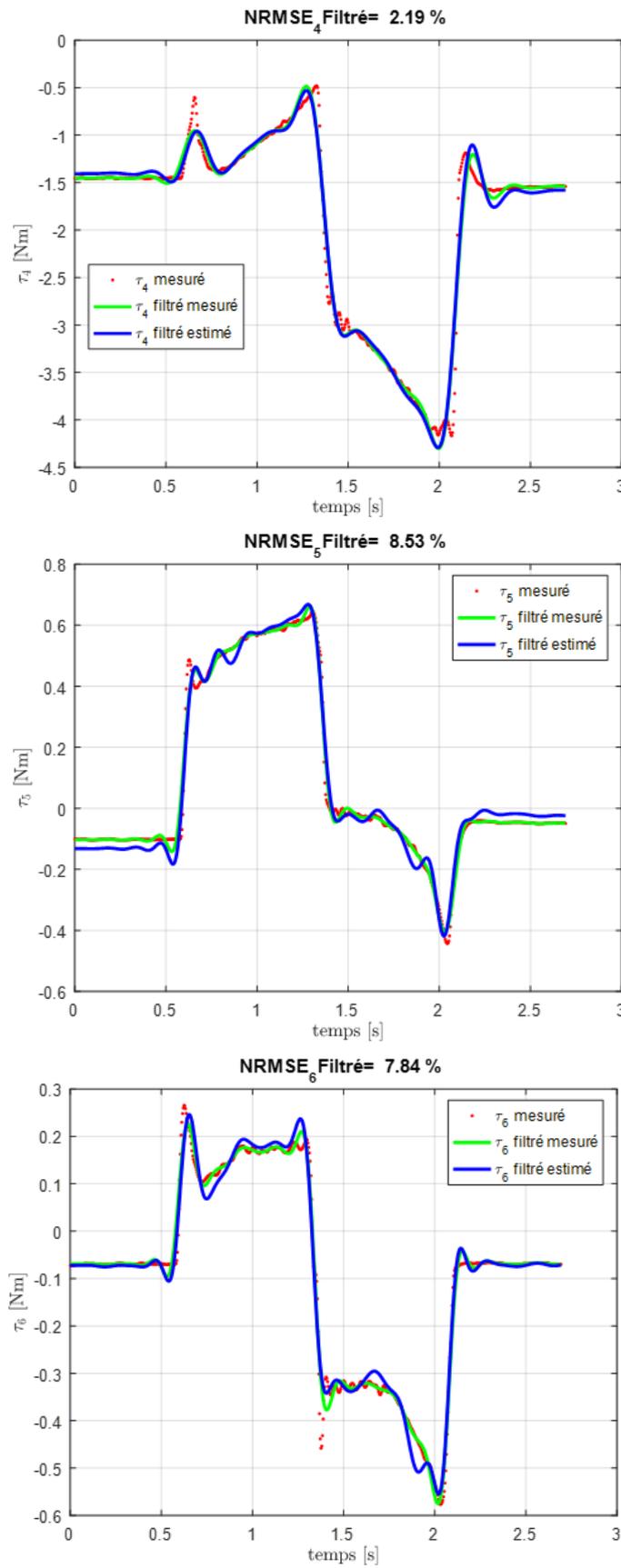


FIGURE 3.4 – Comparaison des couples mesurés, filtrés et estimés pour les articulations 4,5 et 6

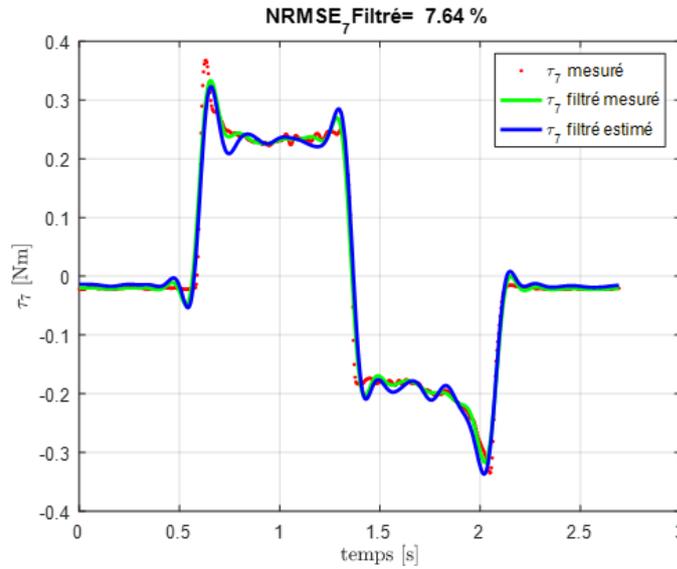


FIGURE 3.5 – Comparaison des couples mesurés, filtrés et estimés pour l'articulation 7

Les Figures (Figure 3.3), (Figure 3.4) et (Figure 3.5) sont des représentations de la comparaison des couples mesurés, filtrés et estimés pour chaque articulation du robot.

TABLE 3.2 – Résultats de l'identification des paramètres de frottement et des offsets

Paramètre	$\hat{\chi}_{optim}$	$\%P_{\hat{\chi}}$
$Off_1$	3.12	6.05
$Off_2$	1.24	5.82
$Off_3$	1.163	2.71
$Off_4$	0.45	7.26
$Off_5$	0.08	11.88
$Off_6$	0.17	3.40
$Off_7$	-0.01	15.93
$Fc_1$	2.18	<b>0.38</b>
$Fv_1$	0.367	<b>0.99</b>
$Fc_2$	1.826	<b>0.52</b>
$Fv_2$	0.758	<b>0.61</b>
$Fc_3$	0.718	<b>1.05</b>
$Fv_3$	0.183	<b>1.97</b>
$Fc_4$	0.836	<b>0.89</b>
$Fv_4$	0.18	<b>1.90</b>
$Fc_5$	0.228	<b>2.87</b>
$Fv_5$	0.044	6.65
$Fc_6$	0.183	3.58
$Fv_6$	0.04	7.52
$Fc_7$	0.132	4.93
$Fv_7$	0.045	6.43

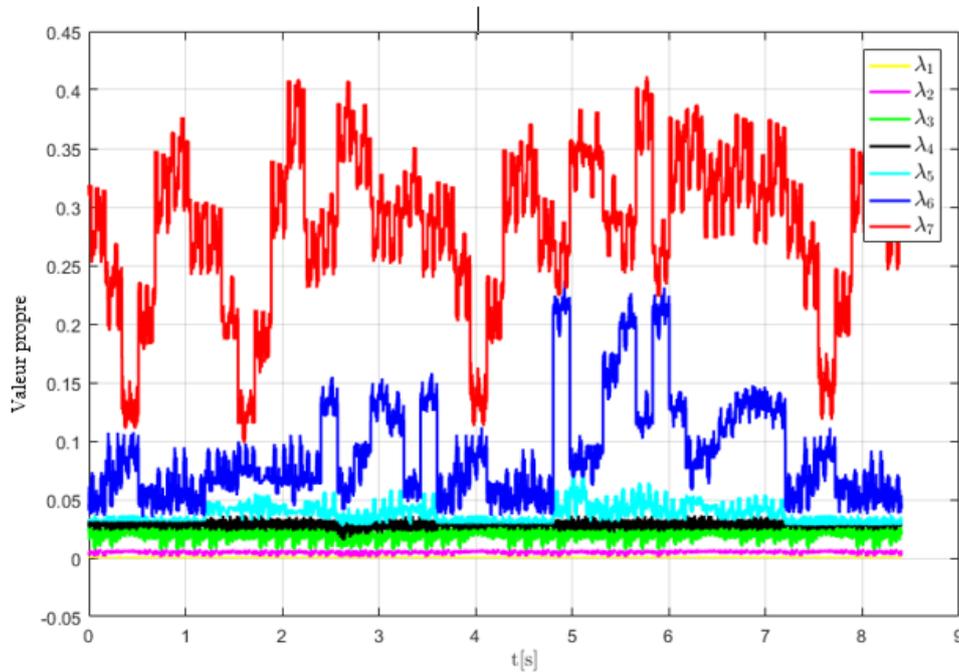


FIGURE 3.6 – Évolution des valeurs propres de la matrice d’inertie sur l’ensemble de l’espace de travail du bras de YuMi

Dans le tableau (Table 3.3) figurent les valeurs numériques estimées des paramètres inertiels dynamiques. De manière générale, les précisions des estimations sont remarquables ; elles varient entre 0,5 % et 15 %. De fortes imprécisions sont observées sur quelques paramètres inertiels ayant de faibles valeurs numériques, preuve que ces paramètres ne contribuent pas réellement à la dynamique du robot, d’où leur suppression du modèle identifié. D’autres par contre, à savoir  $XX_{R4}$  et  $YZ_{R4}$ , sont légèrement moins précis, toutefois ils peuvent être facilement améliorés avec davantage d’itérations d’optimisation.

Dans cette première partie, nous avons réalisé l’identification dynamique du robot YuMi, en déterminant ses paramètres dynamiques ainsi que les paramètres de frottement et les offsets des couples. En réalité, cette identification nous permet de réaliser le véritable simulateur du robot. Ce simulateur va nous permettre de tester et de valider nos lois de commande avant leur application concrète sur le robot ou même son simulateur sur robotstudio pour lequel l’application de lois de commande externes n’est pas réalisable et dont le comportement dynamique représente une légère différence par rapport au robot YuMi.

Dans ce but, nous avons construit notre propre simulateur qui sera utilisé par la suite non seulement pour ce projet, mais aussi pour d’autres projets sur le robot nécessitant des validations en amont des tests expérimentaux. Le développement du simulateur complet du robot est présenté dans la partie suivante de ce chapitre.

TABLE 3.3 – Paramètres inertiels identifiés

Paramètres	$\hat{\chi}$	$\%P_{\hat{\chi}}$	Paramètres	$\hat{\chi}$	$\%P_{\hat{\chi}}$
$ZZ_{R1}$	-0.019	3.61	$XZ_{R5}$	0.0014	<b>1.45</b>
$MX_{R1}$	0.466	<b>1.09</b>	$YZ_{R5}$	0.0019	4.12
$MY_{R1}$	-2.157	<b>0.51</b>	$ZZ_{R5}$	-0.0007	4.25
$XX_{R2}$	-0.023	<b>2.69</b>	$MX_{R5}$	0.0027	3.39
$XY_{R2}$	0.016	<b>1.07</b>	$MY_{R5}$	0.0063	3.14
$XZ_{R2}$	0.007	2.20	$XX_{R6}$	-0.0005	4.62
$YZ_{R2}$	-0.005	6.09	$XY_{R6}$	7.99e – 05	6.26
$ZZ_{R2}$	-0.011	4.96	$YZ_{R6}$	-0.0005	<b>2.10</b>
$MX_{R2}$	-0.394	<b>2.22</b>	$ZZ_{R6}$	-0.00026	9.45
$MY_{R2}$	-0.772	<b>0.62</b>	$MX_{R6}$	-0.0021	<b>2.49</b>
$XX_{R3}$	0.009	<b>2.95</b>	$MY_{R6}$	0.00184	3.70
$XY_{R3}$	-0.003	4.95	$XX_7$	-0.00045	<b>2.59</b>
$XZ_{R3}$	0.036	<b>2.77</b>	$XY_7$	-3.9e – 05	8.68
$YZ_{R3}$	-0.022	3.49	$XZ_7$	3.61e – 05	6.17
$ZZ_{R3}$	0.0009	11.01	$YZ_7$	0.00052	<b>2.59</b>
$MX_{R3}$	0.1415	<b>2.77</b>	$ZZ_7$	-0.00036	<b>2.34</b>
$MY_{R3}$	-0.093	<b>2.93</b>	$MY_7$	0.00219	<b>1.90</b>
$XX_{R4}$	0.0007	14.80	$IA_4$	0.00388	<b>1.94</b>
$YZ_{R4}$	0.0002	15.19	$IA_5$	0.00119	3.92
$ZZ_{R4}$	-0.003	5.91	$IA_6$	0.00059	8.40
$MY_{R4}$	0.0103	4.43	$IA_7$	0.00045	3.16

## 3.4 Simulateur de robot

### 3.4.1 Structure du contrôleur de robot

Le contrôleur du YuMi est le contrôleur IRC5 de ABB. Ce dernier est caractérisé par une structure de contrôle où chaque articulation est commandée individuellement. Cette structure de commande en cascade (Figure 3.8) est utilisée pour chaque articulation dont la boucle de position extérieure a un gain proportionnel, tandis que la boucle de vitesse interne possède à la fois une rétroaction proportionnelle et intégrale. Les paramètres du contrôleur sont les trois gains  $K_p$ ,  $K_v$  et  $K_i$ . Une boucle de contrôle du courant est exécutée dans le contrôleur du robot. La boucle de contrôle du courant garantit que le moteur actionne le couple souhaité.

Selon [128] une interface de recherche est disponible pour les robots qui permet de modifier les gains de contrôle ( $K_p$ ,  $K_v$ ,  $K_i$ ), en revanche la structure du contrôleur n'est pas modifiable. Nous avons essayé de trouver ces valeurs de manière empirique en gardant les allures d'erreur entre les vitesses réelles et les vitesses de référence du robot et les positions. Les boucles de contrôle de bas niveau fonctionnent avec une fréquence d'échantillonnage de 2 kHz, tandis que l'interface de recherche pour définir les références et lire les mesures fonctionne à 250 Hz.



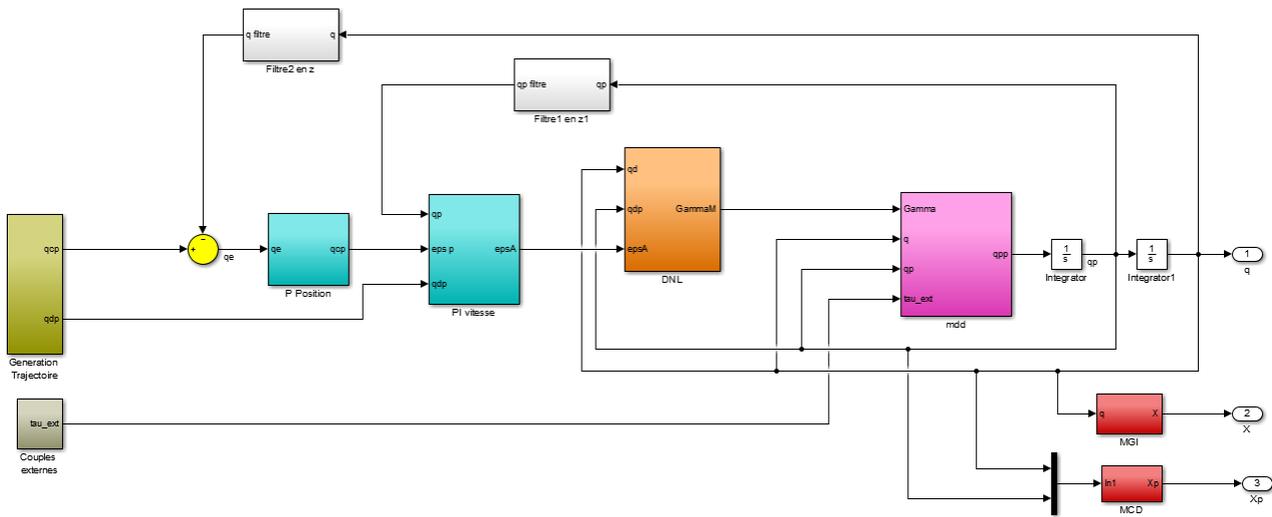


FIGURE 3.8 – Schéma du principe de commande bas-niveau du contrôleur de robot sur Simulink

La figure 3.8 est une illustration du bloc schématique utilisé dans la simulation. Les fonctions des différents blocs sont :

- Générateur de trajectoires (bloc vert) : le générateur des trajectoires des positions et des vitesses articulaires souhaitées,  $(q_d, \dot{q}_d)$
- Couples externes (bloc gris) : le générateur des couples externes appliqués au robot
- P (bloc bleu clair) : le contrôleur de position de type proportionnel
- PI (bloc bleu clair) : le contrôleur de vitesse de type proportionnel-intégral
- DNL (bloc orange) : la dynamique non linéaire du robot
- MDD (bloc rose) : le modèle dynamique du robot
- MCD et MGI (blocs en rouge) : le modèle cinématique direct et le modèle géométrique inverse du robot.

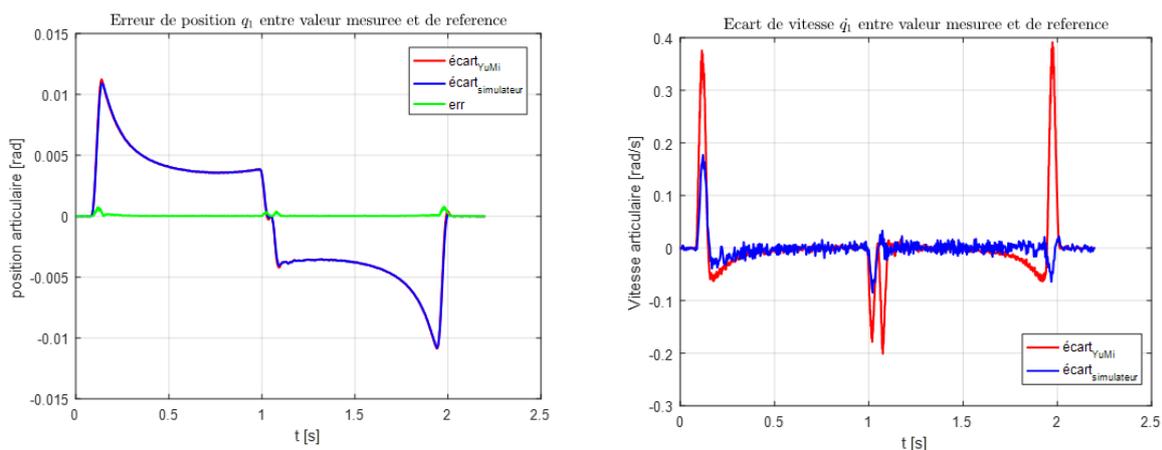


FIGURE 3.9 – Écart de la position  $q_1$  et de la vitesse  $\dot{q}_1$  de l'axe 1 entre la valeur réelle et la valeur de référence

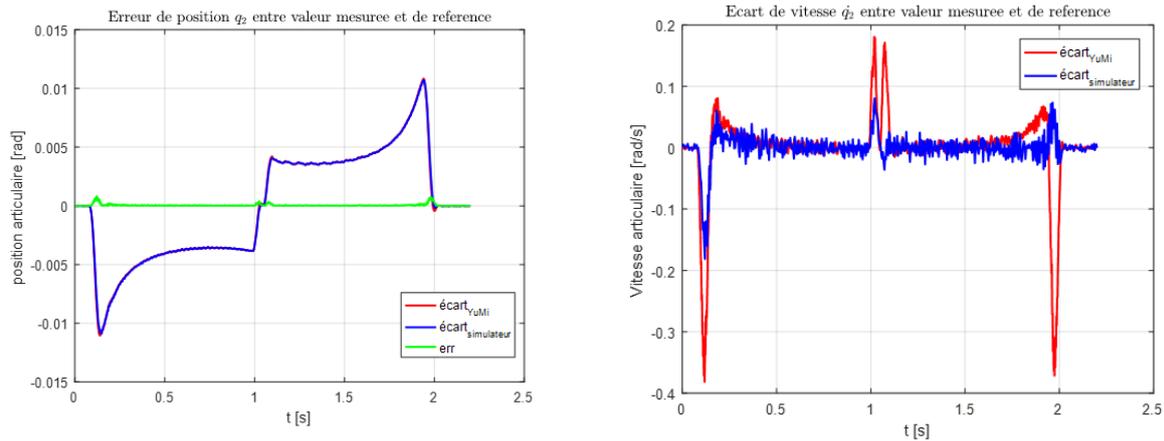


FIGURE 3.10 – Écart de la position  $q_2$  et de la vitesse  $\dot{q}_2$  de l’axe 2 entre la valeur réelle et la valeur de référence

### 3.5 Calcul des forces externes

Les opérations visées par la robotisation étaient en principe réalisées par un être humain. De plus, cette personne était la plupart du temps en contact direct avec la tâche qu’elle effectuait. Par conséquent, les forces de contact générées lors de l’exécution de la tâche étaient gérées par cette personne, de manière inconsciente ou non. Certaines de ces opérations font l’objet d’une robotisation [44] afin de soulager l’être humain des tâches fastidieuses, ou parce que ces tâches nécessitent une précision ou une dextérité, difficiles à réaliser par un être humain. Afin de réaliser correctement ces tâches, le robot doit donc chercher, tout comme l’être humain, à s’adapter en permanence à ces interactions.

La prise en compte des interactions nécessite des mesures des positions et des forces relatives entre le robot et son environnement, par ailleurs l’exploitation de ces données permet de modifier et d’adapter le mouvement de référence en fonction des forces de contact.

La capacité du robot à réagir aux forces externes appliquées à son outil est appelée *compliance*<sup>3</sup>. Il existe deux types de *compliance* :

- **Compliance passive** : résultant exclusivement de la flexibilité des structures et de leur capacité à amortir les erreurs de déplacement entre le robot et la tâche (interposer entre l’organe terminal et l’outil un dispositif présentant une certaine élasticité - ressort par exemple).
- **Compliance active** : imposant une prise en compte des forces de contact contrôlées par des actionneurs (développer des commandes qui permettent de maîtriser de façon continue les forces de contacts). Cette approche nécessite la connaissance des efforts de contacts.

La connaissance des forces de contact est donc une exigence [43] pour des opérations telles que l’assemblage [100], la préhension d’objets fragiles ou fortement déformables [44] ou encore le dénudage des fils.

La meilleure façon de déterminer les forces externes sur l’effecteur terminal d’un robot est d’utiliser un capteur de force de l’effecteur lui-même. Il est également possible d’utiliser des

3. *compliance* : la version française exacte de ce terme, *complaisance*, est rarement utilisée.

capteurs de force au niveau de chaque articulation du robot, comme c'est le cas pour le bras DLR lightweight [14]. Dans [56], une mise en œuvre d'une stratégie de contrôle basée sur trois capteurs différents montés sur le robot Staubli RX60 à six DDL, un capteur de force et des accéléromètres dont le but est d'obtenir un estimateur de force de contact approprié. Cette dernière approche a été utilisée dans [114] en combinaison avec une estimation simultanée d'état et des entrées. Bien qu'il soit facile d'obtenir ces mesures à partir de capteurs de force et de couple spécialisés, il est généralement difficile sur le plan technique ou coûteux de monter de tels capteurs.

Les observateurs sont une autre solution. En effet, on peut utiliser des observateurs de perturbation, à savoir utiliser le modèle dynamique du robot et considérer les distorsions de celui-ci comme des perturbations engendrées par des forces externes [38] et [50]. Ou encore, on peut utiliser des observateurs de force directe, tels qu'un observateur de force  $H$  dans [110] et le filtre de Kalman étendu est proposé dans [89]. Un modèle simplifié pour les observateurs de force a été développé dans [72] et étendu à l'estimation de la force de contact en robotique dans [15]. Selon [129], Il est également possible d'estimer les forces externes si chaque articulation est contrôlée individuellement ; cette pratique est très répandue pour les robots industriels. En désactivant l'action intégrale dans les contrôleurs d'articulation, ceux-ci serviront de ressorts virtuels et la déviation de l'angle de chaque articulation par rapport à sa référence correspondra à un couple articulaire. Par contre, ses erreurs de position peuvent devenir importantes si l'action intégrale est complètement supprimée, ce qui entraîne de mauvaises performances dans le contrôle de la position et un biais dans l'estimation de la force.

En l'absence de capteurs et d'observateurs, il est possible d'utiliser le couple-moteur dans chaque articulation. Cette approche nécessite que les couples mesurés soient corrigés des sources de distorsion (gravité et frottement) avant de pouvoir utiliser les signaux de couple. Il faut également tenir compte du fait que le robot possède ou non des réducteurs, car un ratio important amplifie le bruit et les erreurs de modèle et rend difficile l'obtention d'une estimation précise. Dans notre cas, grâce à l'identification, nous disposons d'un modèle dynamique qui nous permet de calculer facilement les couples externes :

$$\tau_{mot} = M\ddot{q} + H\dot{q} + \tau_{frott} + \tau_{ext} \quad (3.25)$$

Pour un manipulateur avec  $n$  DDL,  $M$  est la matrice d'inertie définie positive. Les efforts de Coriolis, centrifuge et gravité sont représentés par  $H$ .  $\tau_{frott}$  correspond aux couples de frottement dans les articulations, et  $\tau_{mot}$  sont les couples-moteurs qui alimentent le robot. Les couples d'interaction  $\tau_{ext}$  dans les articulations du robot résultent des forces et des couples de contact. Pour réaliser ce calcul la matrice  $M$  et le vecteur  $H$  sont nécessaires. La matrice  $M$  peut être obtenue facilement à partir du logiciel SYMORO+ en utilisant la fonction "inertia matrix", le calcul de  $H$  est également effectué en utilisant la fonction " calculation of Coriolis and gravity forces ". Pour le couple de frottement, nous l'avons calculé à partir du modèle supposé lors le d'identification, nous avons donc :

$$\tau_{ext} = \tau_{mot} - \tau_{inertie} - \tau_{cor-cen-grav} - \tau_{frott} \quad (3.26)$$

Les forces externes  $F_{ext}$  produites au TCP sont liées aux couples de réaction  $\tau_{ext}$  par la jacobienne du robot  $J(q)$ , soit :

$$\tau_{ext} = J^T(q) \cdot F_{ext} \quad (3.27)$$

donc

$$F_{ext} = (J^T)^{-1} \cdot \tau_{ext} \quad (3.28)$$

- $F_{ext}$  : les forces agissants sur l'effecteur terminal
- $\tau_{ext}$  : les couples externes correspondants des sept articulations

conformément à [125]. Dans le cas général, la dimension du couple est  $n_{ext} = 6$  et ses éléments en coordonnées cartésiennes dans le repère de base sont définis comme suit :

$$F_{ext} = [ f_x \quad f_y \quad f_z \quad \tau_x \quad \tau_y \quad \tau_z ]^T \quad (3.29)$$

Pour le robot YuMi, des mesures de couple externe sont également possibles, sur la base des erreurs de contrôle des boucles de contrôle des articulations de bas niveau. Nous avons donc une autre possibilité de valider notre identification tout en calculant les forces externes à contrôler dans le chapitre 5.

Pour y parvenir, nous avons effectué une comparaison entre le calcul des forces externes calculées à partir des mesures des couples externes et celui issu du modèle dynamique identifié auparavant. Nous avons utilisé ce temps pour la mesure de notre robot à partir du protocole de communication externe réalisée avec le robot et présenté dans le prochain chapitre (Chap. 4). Outre notre robot, nous avons fait des essais sur un autre YuMi de la plateforme Ant-Technology du laboratoire de l'Enim (Ecole Nationale d'Ingénieurs de Metz)(Figure 3.5), pour être en mesure d'évaluer en même temps notre modèle identifié et l'influence de la communication externe sur les mesures.

Sur les figures (Figure 3.11) et (Figure 3.12) sont représentés les couples externes mesurés et calculés sur chaque articulation du YuMi de LCFC et d'Ant-Technologie respectivement. Selon ces deux figures, nous pouvons observer qu'en statique, les couples sont différents tandis qu'en dynamique les mesures sont du même ordre de grandeur et ceci en enlevant les mesures aberrantes. On peut également vérifier sur les figures (Figure 3.5) et (Figure 3.5) que les valeurs de force calculées sont en adéquation avec la force appliquée sur le robot à savoir -5N. On peut également remarquer que la force de YuMi (Ant-Technology) est moins perturbée comparée à celle de YuMi (LCFC) du fait de la perturbation provoquée par la communication externe mise en place sur le robot.

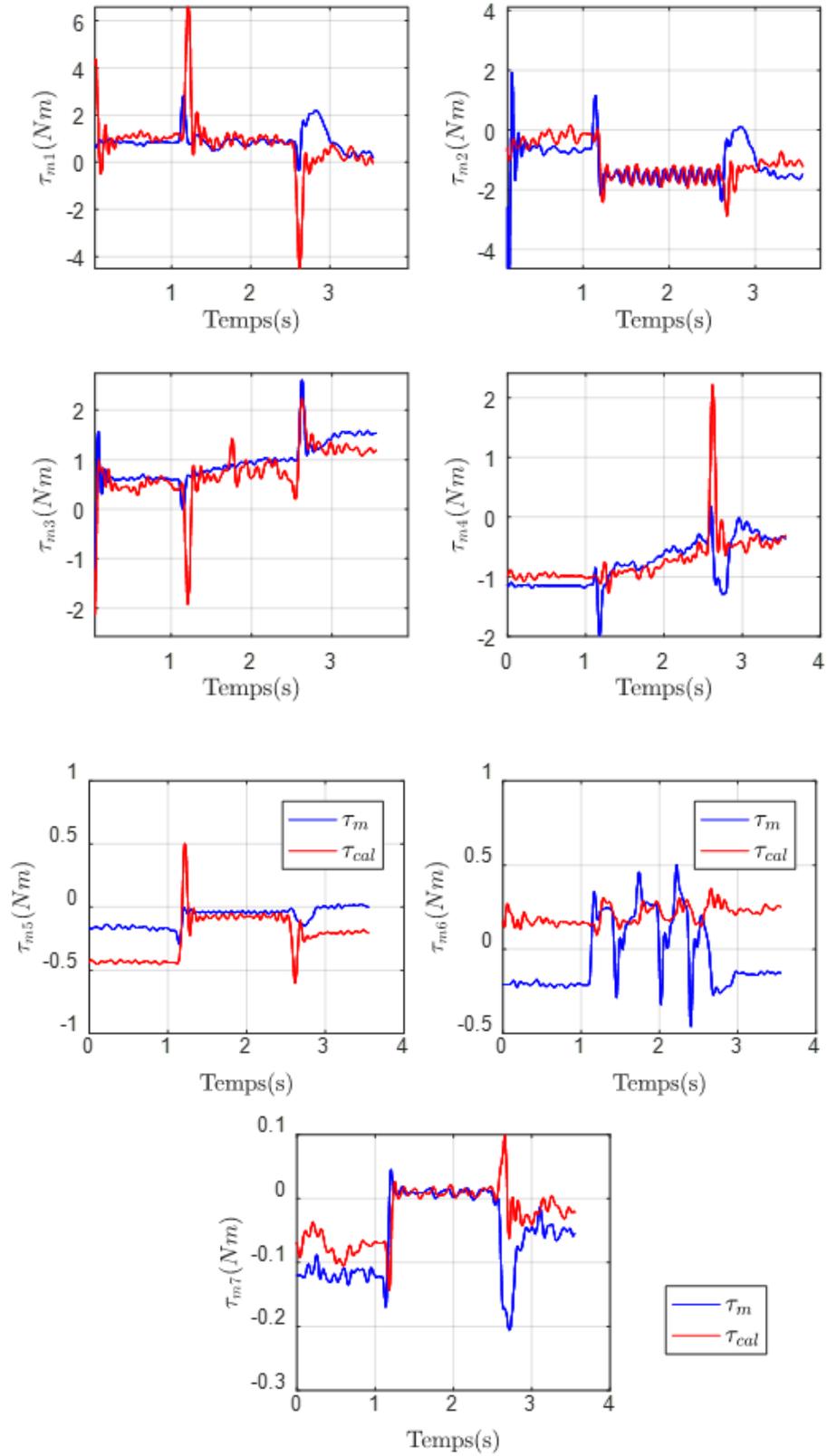


FIGURE 3.11 – Couples externes mesurés et calculés Robot YuMi LCFC

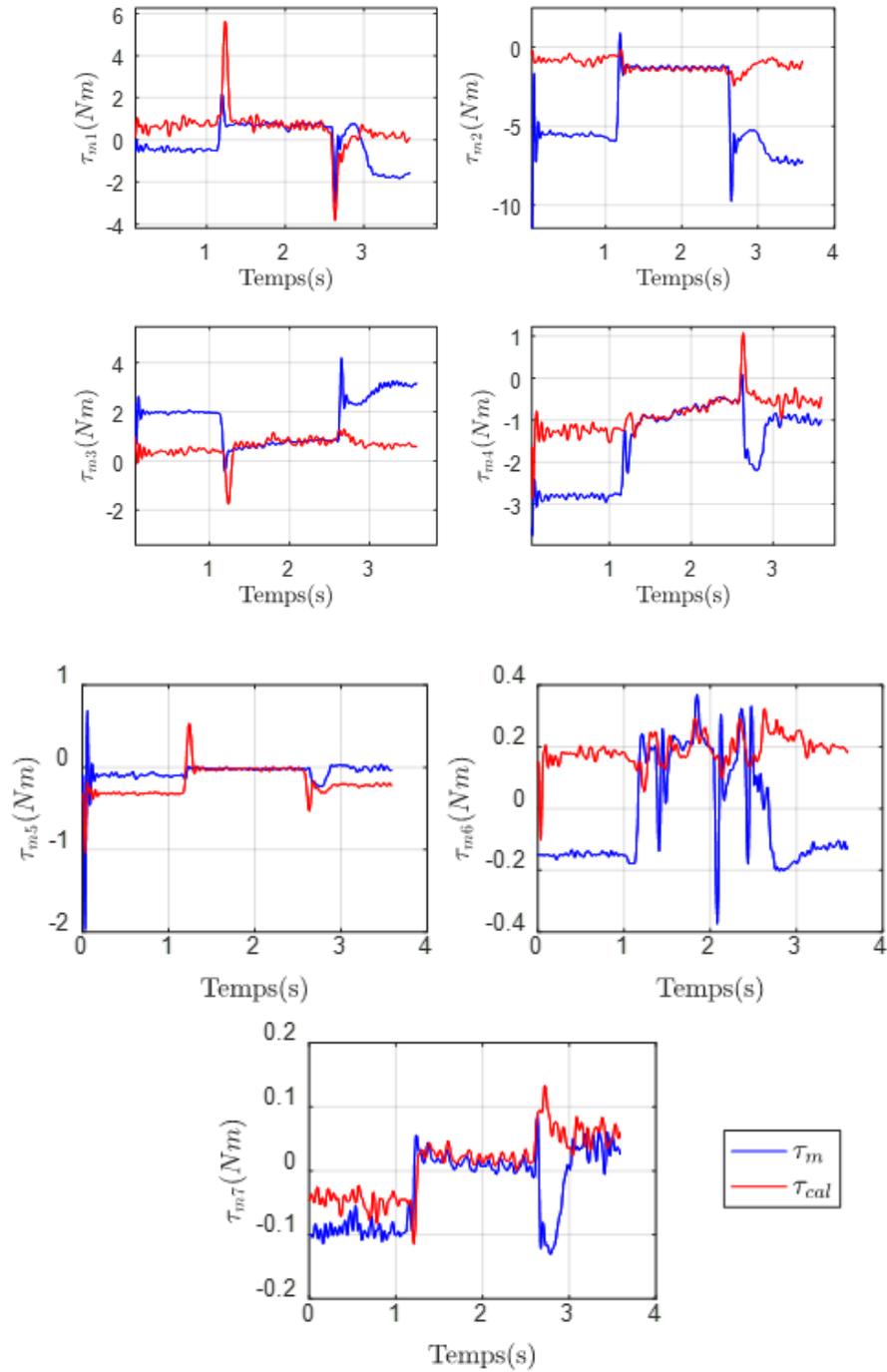
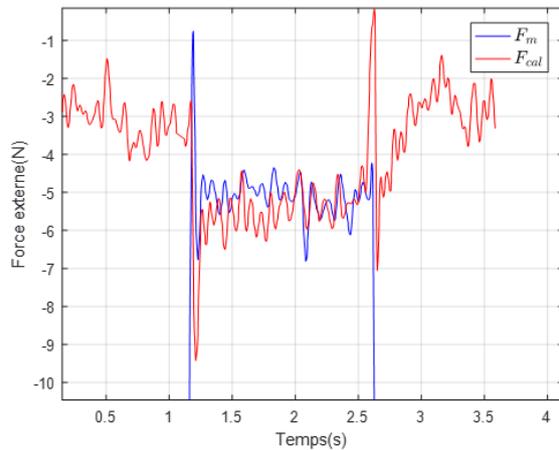
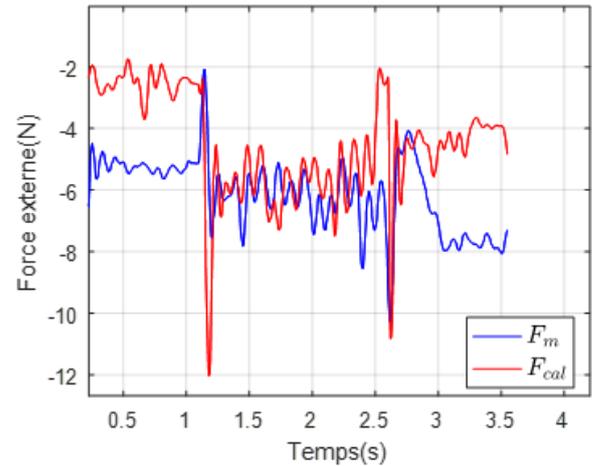


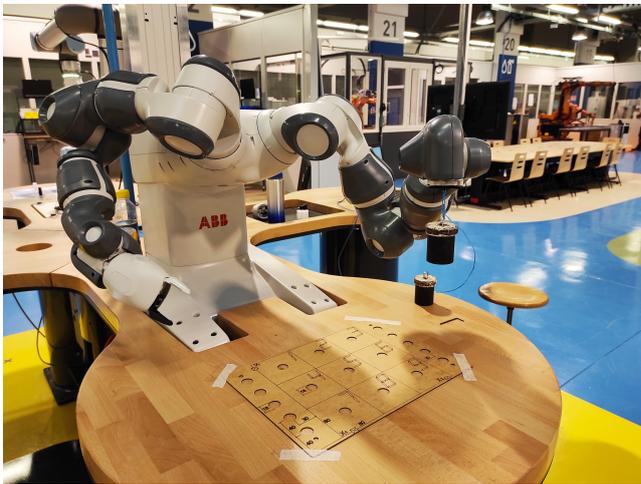
FIGURE 3.12 – Couples externes mesurés et calculés Robot YuMi ant-Technologie



Force externe [N] mesurée et calculée selon  $Z$  du YuMi ant-Technologie



Force externe [N] mesurée et calculée selon  $Z$  du YuMi LCFC



YuMi ant-Technologie



YuMi LCFC

Pour l'évaluation de nos résultats, le robot a été soumis à une force externe appliquée le long de l'axe  $Z$  en utilisant un ressort connecté à un capteur qui calcule la force appliquée le long de l'axe  $Z$ . Ce capteur de force est un capteur à jauge de contrainte qui mesure la force dans une seule direction dans l'espace. La cellule de force est accompagnée d'un amplificateur de force à pont de Wheatstone basé sur un convertisseur analogique-numérique HX711, pour amplifier la mesure des forces appliquées à la cellule de force. Lorsque le capteur est calibré, il renvoie la valeur en kg (ou en N) selon l'unité choisie lors du calcul des paramètres du capteur. Ce dernier est connecté à une carte Arduino pour interfacer le convertisseur analogique-numérique pour la lecture des cellules de force. Ce processus nous permet de générer la mesure en temps réel de la force appliquée par le robot sur le ressort et par conséquent celle appliquée sur le robot selon l'axe  $z$ . La figure 3.13 montre le processus de mesure décrit. Les forces réelles du robot ont également été mesurées à partir des mesures de couple, de position et de vitesse à l'aide de la communication externe effectuée sur le robot (Chap. 4).

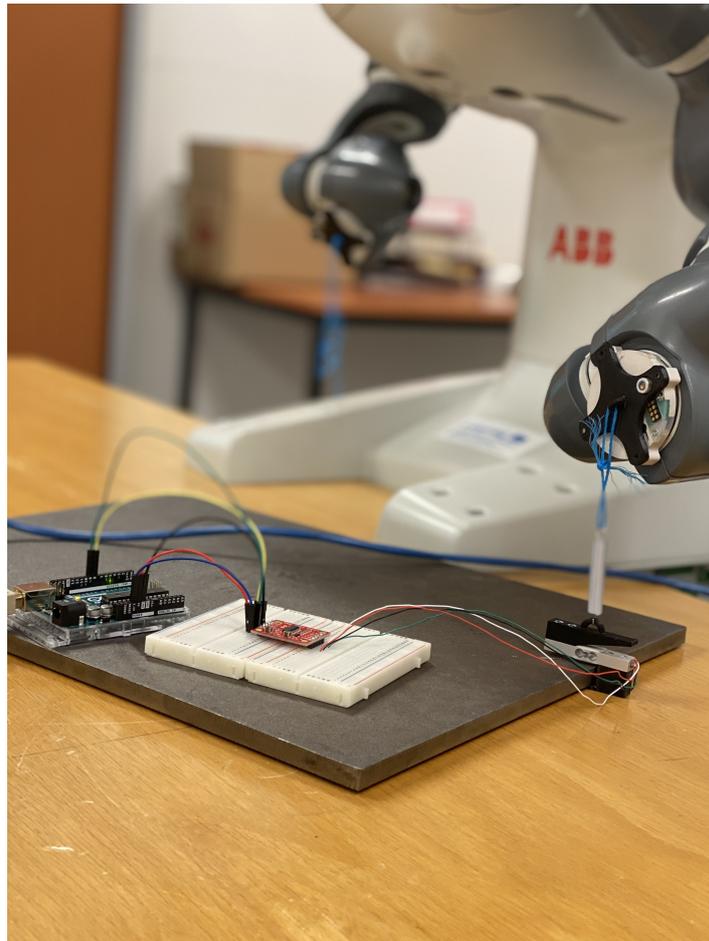
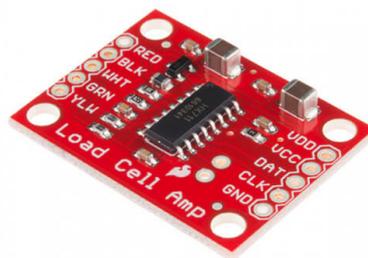


FIGURE 3.13 – Matériel utilisé pour évaluer les couples externes



Arduino UNO

Module amplificateur  
HX711Capteur de charge  
CZL616C

Les figures (Figure 3.14) et (Figure 3.15) présentent les forces mesurées par le capteur et par le robot lui-même comme décrit précédemment pour des forces différentes allant de 0 à 8 N pour le capteur ; celui-ci ne peut peser que 780 g donc au-delà de cette valeur nous considérons que les forces sont fausses ; pour le robot il ne peut manipuler que 500 g par bras. On remarque en effet que pour de petites valeurs de force les mesures sont différentes on trouve par exemple pour 0 N pour le capteur on trouve une valeur de  $-1$  N pour le robot que l'on suppose due à la susceptibilité du robot de sa position aussi et idem pour le capteur. Pour les fortes valeurs, nous voyons par exemple sur le capteur la force maximale équivalente à 7.8 N le robot affiche une force un peu plus proche de 8.8 N donc une différence de 1 N entre les deux. En revanche, pour les valeurs intermédiaires entre 2 et 5 N les valeurs sont plus exactes.

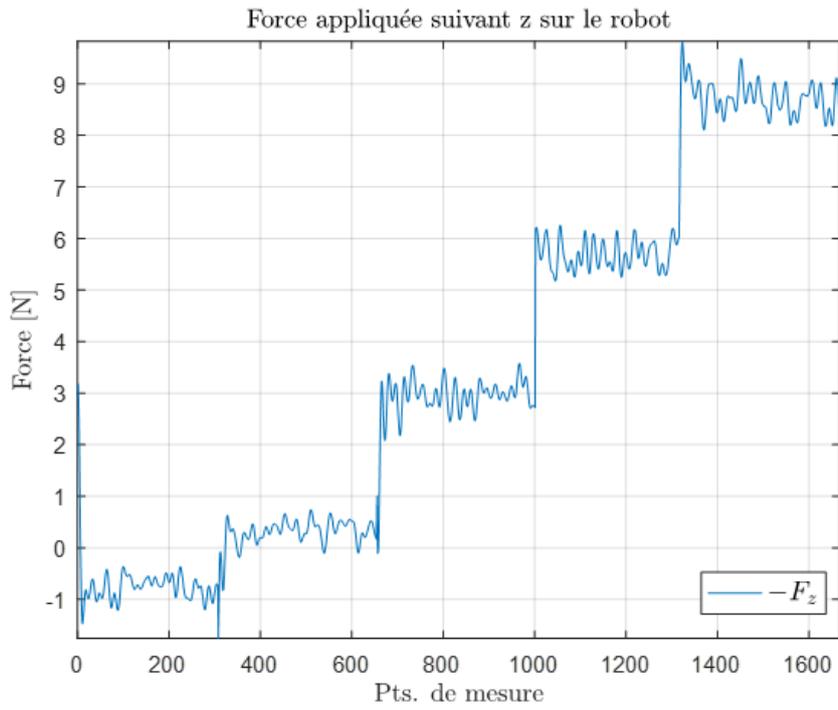


FIGURE 3.14 – Force appliquée suivant  $z$  sur le robot

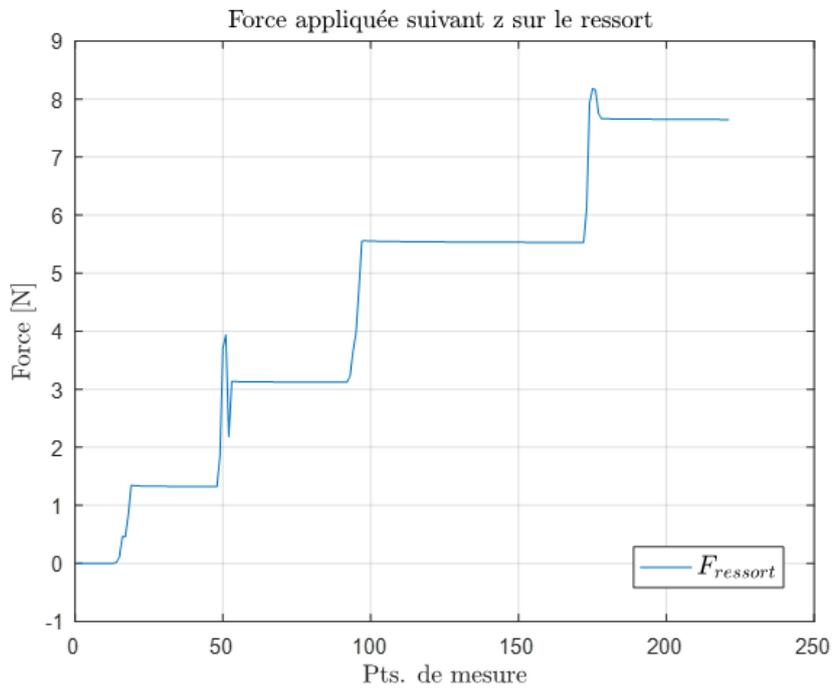


FIGURE 3.15 – Force appliquée suivant  $z$  sur le ressort

## 3.6 Conclusion

Au cours de ce chapitre, nous avons abordé la problématique de l'identification des modèles dynamiques du robot, à savoir l'identification de l'ensemble de ses paramètres dynamiques non-fournis par le constructeur. Nous avons d'abord effectué une brève bibliographie sur l'identification dans le contexte de la robotique. Par la suite, nous avons élaboré notre méthode d'identification étape par étape, en partant des mesures jusqu'à l'optimisation et le calcul des erreurs des paramètres identifiés. Nous avons également exposé les propriétés des différents signaux utilisés dans ce cadre, à savoir les trajectoires d'excitation.

Le traitement des signaux mesurés afin de les rendre utilisables a également été exposé. La technique d'optimisation utilisée est basée sur un critère quadratique (moindres carrés) sous contraintes. Ces contraintes imposées sur les paramètres à caractère physique permettent de s'assurer que la matrice d'inertie est positive quelle que soit la configuration du robot.

Cette procédure nous a permis de déterminer le vecteur de paramètres qui assure une bonne restitution des couples en présence de vitesses constantes ou variables dans le temps. Les résultats de l'identification et les valeurs des paramètres dynamiques, de l'offset et du frottement ont par ailleurs été communiqués. Ceci nous a permis de conclure que la matrice d'inertie calculée est approximativement égale à la matrice d'inertie réelle. De fait, nous pouvons nous baser sur le modèle obtenu pour implémenter sur le robot réel, une commande basée sur le modèle identifié.

Dans un deuxième temps, nous avons réalisé notre simulateur de robot sur Matlab/Simulink, en prenant en compte la structure du contrôleur P/PI en cascade pour l'asservissement de la position et de la vitesse du robot réel; en identifiant ses différents paramètres de manière à avoir les mêmes comportements que le robot réel; en plus du bloc de l'algorithme DNL qui calcule la dynamique non linéaire du robot et les trajectoires des positions articulaires.

La dernière partie de ce chapitre a été consacrée à l'estimation du couple externe et à la vérification des forces résultantes afin de les commander dans le chapitre 5. Une comparaison des résultats de l'identification et du calcul du couple externe a également été détaillée ainsi que la procédure de validation du calcul des forces externes.

# Chapitre 4

## Communication externe du robot

Une commande externe sur un robot industriel exige impérativement la possibilité de communiquer avec ce dernier en temps réel. Toutefois, de nombreux fabricants, soucieux de préserver leurs secrets de fabrication, ne fournissent pas de protocole de communication avec un robot ou le limitent suffisamment pour préserver leur produit. Dans le contexte de la recherche, travailler avec un robot industriel implique la nécessité d'établir une communication. Différents protocoles de communication ont été élaborés. Pour notre thèse, bien que tout fonctionne correctement en simulation, cette dernière n'est pas suffisante pour nous assurer son bon fonctionnement concret sur le robot. Voilà pourquoi nous présentons dans ce chapitre le protocole de communication réalisé sur le robot pour mettre en place notre propre loi de commande.

### 4.1 Introduction

Les robots industriels ont été conçus initialement pour des applications de fabrication et, en toute logique, les outils logiciels de programmation et de contrôle qui les accompagnent répondent à ces besoins. En principe, ces robots sont contrôlés par des contrôleurs embarqués afin d'obtenir une plus grande précision en termes de position et de temps. Ces contrôleurs sont plus rapides, plus compacts et intègrent même des informations sensorielles telles que la force et la vision. Ces évolutions technologiques ont conduit au développement d'un large éventail d'applications pour ce type de robot [27], notamment le fonctionnement continu après l'installation du robot, adapté aux opérations par lots (production et fabrication) ; un programme est alors exécuté qui donne des instructions au robot pour une opération spécifique. De plus, l'accès au contrôleur du robot est restreint, ce qui renforce la dépendance de l'utilisateur envers le fournisseur du robot et la rend persistante [54].

Des recherches plus avancées proposent que les robots puissent travailler dans des processus complexes qui nécessitent la récupération d'informations externes à leur environnement de travail. Dans notre exemple de thèse, nous devons mettre en œuvre une loi de contrôle externe position/force pour manipuler des objets, exercer des forces sur eux et effectuer des tâches de compliance. Dans ces cas, l'ordinateur principal doit traiter les informations de force et de position, qui doivent être envoyées au contrôleur en temps réel. Il est en effet indispensable que ces informations soient envoyées au contrôleur en temps réel pour que le robot puisse saisir l'objet dans la position et avec la force souhaitées. Si les objets sont nombreux ou si la force est élevée, il est plus difficile d'accomplir la tâche requise d'où le rôle important d'une très bonne fréquence de communication.

Deux freins s'opposent toutefois à une innovation significative, tout d'abord, l'apprentissage de l'utilisation du robot, qui nécessite une formation technique approfondie dans les détails de sa construction et de son fonctionnement. Ensuite, l'exploration de nouvelles applications pour le robot exige un effort supplémentaire sur l'échelle technologique, car elle implique souvent le développement de systèmes de contrôle adaptés qui permettent à l'utilisateur de contrôler le robot d'une manière différente de celle pour laquelle il a été conçu. Une maîtrise des particularités de robot en question est toutefois nécessaire et les solutions développées deviennent difficiles à généraliser à des familles de robots plus larges.

Ce problème n'est pas nouveau et les chercheurs sont constamment à la recherche de solutions pour le résoudre. Dans [53] un contrôleur industriel a été conçu sur la base du robot C3G 9000 de Comau, ils ont développé un nouveau contrôleur industriel pour le robot pour remplacer le contrôleur commercial. L'équipe a incorporé des cartes de capteurs dans le système, connectées aux unités de contrôle principales via un bus commun. Les deux unités, le contrôleur du robot et les servomoteurs, comprennent plusieurs microprocesseurs et processeurs de signaux numériques (DSP). Toutefois, ils ne rapportent aucun résultat sur les performances du contrôleur de robot développé. Par ailleurs, le coût d'une solution multiprocesseur est bien plus élevé que celui obtenu par une technique reconfigurable. Le développement d'un système de commande de mouvement basé sur un FPGA (field programmable gate array) dans le cadre d'un système de commande à architecture ouverte et indépendant du constructeur est testé sur un Fanuc S420F dans [102].

Une autre approche du contrôle en temps réel est présentée dans [68]. Cette fois, il s'agit d'adapter le contrôleur commercial existant basé sur le langage d'assemblage Victor (VAL) du robot PUMA 560 et d'intégrer de nouveaux matériels et logiciels, de sorte que le contrôle par rétroaction et la programmation hors-ligne sont possibles. Le système développé n'implique pas le temps réel et ne peut être utilisé qu'avec des robots actionnés par des moteurs à courant continu CC. Dans [111] une plate-forme expérimentale permettant l'implémentation d'algorithmes de contrôle avancés et l'interaction facile avec des capteurs externes dans le robot industriel Stäubli est abordée. Dans [142] une autre plate-forme matérielle et un environnement de programmation développé sur MATLAB/Simulink pour la conception, l'analyse et la mise en œuvre de robots industriels sont présentés. La plate-forme est composée de deux robots industriels et d'une architecture logicielle de calcul distribué basée sur un PC. La génération automatique du code de programme est effectuée par les boîtes à outils Real-Time Workshop et Real-Time Windows Target fournies par MATLAB.

La majorité des méthodes utilisant des ordinateurs pour contrôler des robots se heurtent au problème du système d'exploitation du robot. L'utilisation du système Windows classique provoque des interruptions en temps réel et crée des problèmes de communication. Pour surmonter ce problème, un système d'exploitation en temps réel doit être utilisé pour contrôler les robots, mais la construction d'un système d'exploitation en temps réel basé sur le système d'exploitation Windows revient cher. À cet effet, le système d'exploitation standard open source Linux présente non seulement plus d'avantages que les systèmes d'exploitation normaux en termes d'économie et de flexibilité, mais fournit également une véritable solution aux chercheurs pour résoudre le problème d'interruption qui se produit normalement lorsque le logiciel de contrôle est basé sur le système d'exploitation Windows. En [115], l'auteur s'est intéressé au contrôle de la trajectoire du robot à 5ddl dans un temps souhaité en utilisant un système d'exploitation moderne en temps réel appelé Linux-Xenomai. La référence [34] a également décrit la mise en œuvre d'une architecture informatique à double système d'exploitation, basée sur Linux stan-

dard et Linux embarqué en temps réel, pour le contrôle d'un robot mobile.

Il existe également des stratégies de communication pour lesquelles il n'est pas nécessaire de modifier le système de commande ni même d'intervenir sur le robot, ils sont basés sur la téléopération robotique avec une procédure standard client-serveur connue sous le nom de TCP / IP socket. La référence [112] traite la capacité d'un système de commande Mitsubishi à communiquer avec un système externe sans modification du matériel. Toutes les communications se font via l'interface TCP/IP en utilisant le protocole UDP. Les données en temps réel telles que la position et l'orientation de l'effecteur, les couples des articulations, sont transmises du contrôleur Mitsubishi à une machine externe et observées par le logiciel LabView.

La référence [16] propose une communication open source pour la surveillance et le contrôle du robot industriel IRB2000 via une interface basée sur le web. Un serveur développé sous Solaris<sup>1</sup> se compose de deux modules : le premier est utilisé pour envoyer/recevoir des commandes série vers/ depuis le contrôleur du robot ; et le second est utilisé pour gérer la transmission des flux vidéo sur un protocole standard TCP/IP avec une bande passante de transmission importante.

Dans [26] un contrôle pour une cellule robotique avec trois robots industriels est proposé, où un robot Mitsubishi a été choisi comme maître et deux robots ABB comme esclaves, sur un réseau TCP/IP avec une méthode de communication socket compatible pour chaque robot de la cellule. Une étude comparative entre deux méthodes compatibles avec les contrôleurs ABB a également été présentée, la première solution étant basée sur un module de développement logiciel pour PC (PC-SDK) disponible pour les développeurs des applications ABB [120], écrit en langage de programmation C, et la seconde ayant été développée comme une méthode de programmation socket basée sur LabView [134]. Dans [140] un programme open source pour la téléopération d'un robot industriel avec la méthode de communication par socket est proposé.

De cette revue d'ensemble, on peut constater que le problème de la communication avec les robots n'est pas nouveau et que les chercheurs sont à ce jour à la recherche des méthodes de communication simples, efficaces et adaptées. Néanmoins, toutes les méthodes développées jusqu'à présent sont malheureusement spécifiques à des marques et des types de robots particuliers.

## 4.2 Définition de la méthode de communication

Le contrôle des robots industriels est lié à des aspects tels que la mécanique des robots, les outils cinématiques, la configuration des effecteurs et les langages de programmation. Cette partie se penche principalement sur les problèmes de mise en réseau des interfaces de communication existantes. Afin de simplifier le protocole de communication, les notions techniques sont contenues dans la programmation avec le langage RAPID<sup>2</sup> [143] et le langage Python, la mise en réseau LAN (Local Area Network) et la programmation des sockets pour l'utilisation du protocole TCP (Transmission Control Protocol). Cette approche de communication est répartie en trois volets et elle est inspirée du travail [124] :

---

1. *Solaris : système d'exploitation UNIX [9]*
2. *RAPID : (Robot Application Programming Interface Dialog) langage de programmation de haut niveau destiné à la commande des robots industriels de ABB. Introduit en 1994 par ABB pour succéder au langage de programmation ARLA et son éditeur de texte est intégré dans le logiciel RobotStudio. Les programmes peuvent être écrits dans RobotStudio, puis téléchargés vers le robot réel.*

### 4.2.1 Premier volet : Création d'une interface client/serveur

Afin d'établir une communication en direct entre le PC et le contrôleur du robot, les données sont transmises en continu du PC au contrôleur du robot IRC5. Pour le flux de données, une application client-serveur dans laquelle le PC agit comme un client et le robot comme un serveur. Une telle application client-serveur est nécessaire pour établir un réseau entre deux plates-formes existantes. La connexion entre le client et le serveur dans ce réseau se fait soit sur un réseau local, soit sur un réseau étendu tel que l'internet. Ce service de réseau et le flux de données sont assurés par la mise en place de la phase de transport qui comprend deux types de protocoles, UDP et TCP. La programmation des sockets est utilisée comme interface pour ces protocoles. On peut voir dans la figure (Figure 4.1) comment les sockets sont créés dans une application client-serveur avec le protocole TCP. Il est donc essentiel de déterminer le type de réseau et de protocole pour transmettre les données.

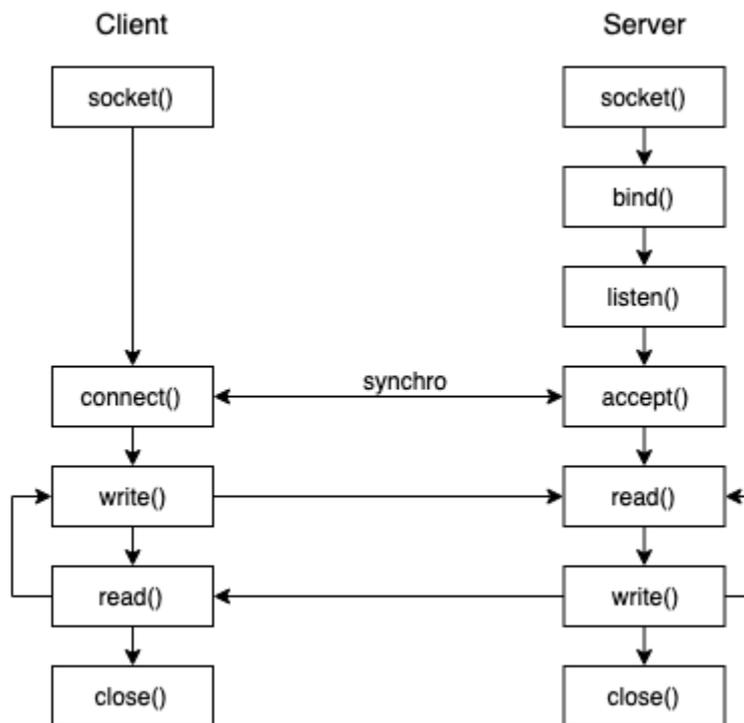


FIGURE 4.1 – Schéma du principe d'une communication client-serveur avec le protocole TCP

### 4.2.2 Deuxième volet : Protocoles UDP et TCP

La possibilité de sélectionner TCP ou UDP comme interface de transmission dépend de chaque système. Certains systèmes et environnements peuvent ne pas prendre en charge la programmation de socket via TCP ou UDP. L'UDP est un protocole qui transmet des paquets de données indépendants, appelés data-grammes, d'un ordinateur à un autre sans aucune garantie de la réception de données et de la fréquence [12]. Il suppose que le contrôle et la correction des erreurs, ne sont pas importants ou ne sont pas effectués dans l'ensemble du processus, pour éviter une surcharge de traitement au niveau de l'interface réseau. Lorsque la transmission des données, est effectuée via un réseau local, le délai et l'ordre de transmission de données ne posent désormais aucun problème. D'autre part, le contrôleur de robot YuMi IRC5 est programmé pour fonctionner uniquement avec TCP. Le protocole TCP est un protocole de connexion, ce qui signifie qu'une connexion est établie et maintenue jusqu'à ce que les programmes d'application

de chaque extrémité achèvent d'échanger toutes les données. Il garantit l'intégrité des données transmises sur le réseau, quelle que soit la quantité de données envoyées. Si deux interfaces ne peuvent pas communiquer par un protocole mutuel, une troisième interface est insérée dans la transmission des données en tant que médiateur.

### 4.2.3 Troisième volet : Python, une interface intermédiaire

Une interface intermédiaire entre le logiciel de conception et le contrôleur du robot est développée en Python. Cette interface est destinée à créer deux réseaux distincts et agit à la fois comme un serveur et un client. Dans le premier réseau, il s'agit d'un serveur sur le réseau local pour recevoir des données sur UDP du logiciel de conception et dans le second, il s'agit d'un client pour transmettre des données sur TCP/IP au contrôleur du robot IRC5. Les avantages de l'insertion d'un tel médiateur dans le flux de travail sont importants, car il permet aux utilisateurs de personnaliser le flux de travail en fonction de leurs propres besoins et rend le flux de données vers l'interface de commande plus flexible pour être indépendant du logiciel de conception. Selon [124], les utilisateurs ont le choix de retenir d'autres logiciels de conception, à condition que, le logiciel sélectionné soit capable de prendre en charge les protocoles UDP ou TCP. Bien que les logiciels prennent en charge le flux de données sur TCP/IP, le maintien de l'interface de la couche intermédiaire permet aux utilisateurs qui ne souhaitent pas coder en RAPID de programmer en python .

Dans notre thèse, dans cette partie, nous utilisons l'interface python de Yumipy [10] pour interagir avec le robot. Cette interface est utilisée pour communiquer via ROS avec le robot. A noter toutefois que dans notre projet, seule la partie python est retenue. Tous nos programmes de contrôle sont réalisés avec python. La figure 4.2 illustre comment on peut créer un socket en langage RAPID programmé pour une interface client/serveur.

```
MODULE Rob Talk
VAR socketdev ServSock; !Socket for server
VAR socketdev InSock; !Socket for client
VAR string received_string;
VAR string stRecieved; ! String from socket
VAR string dientIP;
VAR bool listen = FALSE;
PROC RobComm()

    SocketCreate ServSock;
    S = socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)
    SocketBind ServSock, "128.2.109.20",4997;
    SocketListen ServSock;
    SocketAccept ServSock, InSock,
    WHILE TRUE DO

        !variable data inserted here
        SocketRecieve InSock
        str-stRecieved;
        TPWRITE stReceived

    ENDWHILE
ENDMODULE
ENDPROC
```

FIGURE 4.2 – Programmation des sockets en langage RAPID [124]

*Remarque : nous avons réussi à faire une communication simple entre Matlab et Robotstudio avec le protocole TCP/IP en mettant Matlab comme client et Robotstudio comme serveur mais, cette communication n'est pas assez rapide à cause de système d'exploitation Windows, elle n'envoie que des messages simples et elle ne fonctionne pas sur le robot réel YuMi.*

### 4.3 Résultats de communication

Pour établir la communication nous avons utilisé l'interface Yumipy [10], une interface open source qui permet de communiquer avec le robot YuMi, à savoir cette interface est encore en cours de développement et nous l'avons utilisée pour communiquer avec le robot en temps réel basé sur la méthode de communication décrite auparavant. Dans cette interface certaines fonctions de RAPID, ont été traduites en python pour nous permettre de faire des mouvements avec le robot sans avoir recours à la programmation RAPID ce qui nous a permis de programmer nos problématiques entièrement en python. Cette interface utilise également ROS (Robot Operating System) comme logiciel de contrôle, mais dans notre cas, nous n'avons besoin que de l'interface python pour contrôler notre robot. De plus, la documentation sur l'ensemble du processus d'installation et les exigences relatives des versions du logiciel et du système d'exploitation ne sont pas très claires étant donné que l'interface est encore en cours de développement. Pour utiliser cette interface, nous avons besoin de :

- Un PC avec un système d'exploitation linux (Ubuntu 16.04 64 bits).
- L'installation du module Yumipy sur linux et du module autolab-core d'AutoLab [10].

Tout d'abord, nous avons établi un réseau local entre le robot et le PC (Figure 4.3). Le réseau local (LAN) est un réseau qui inter-connecte des dispositifs dans une zone limitée et ces dispositifs se reconnaissent et se connectent entre eux grâce à des types spécifiques d'adresses IP. Dans ce contexte, le port du service X23 du YuMi et le port de service ethernet du PC, sont reliés par un câble Ethernet. En outre, dans la partie réseau du protocole TCP/IP, l'adresse de protocole Internet (adresse IP) est une étiquette numérique attribuée à chaque dispositif connecté à un réseau qui utilise le protocole Internet pour communiquer. Elle a deux fonctions principales, l'identification de l'interface réseau et la localisation. Toutes les adresses IP utilisées dans notre cas sont des adresses IP statiques et le service DHCP du PC est désactivé pour éviter le double envoi de données. Concrètement, le robot et le PC, se reconnaissent mutuellement lorsqu'ils sont configurés avec des adresses IP, des masques de sous-réseau et des passerelles appropriés. Comme le montre le tableau (Table 4.1).

	Robot YuMi (RAPID)	Interface Yumipy PC
Adresse IP	192.168.125.1	192.168.125.5
Masque de sous-réseau	255.255.255.0	255.255.255.0
Passerelle	192.168.125.1	192.168.125.1

TABLE 4.1 – La configuration appropriée des composants dans le réseau local

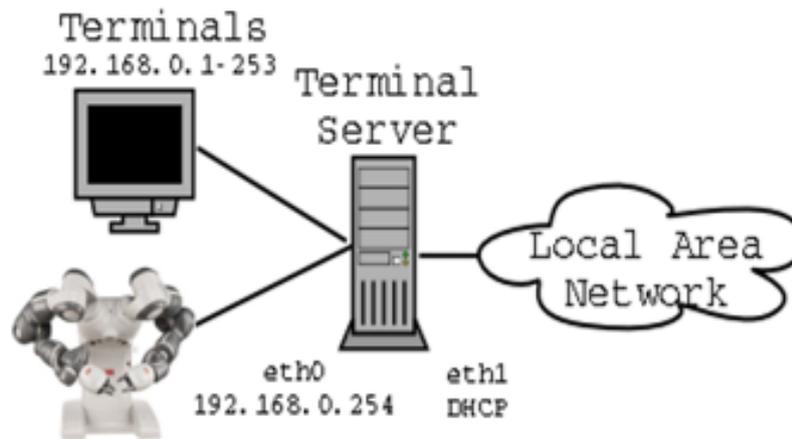


FIGURE 4.3 – LAN construit entre YuMi et l’ordinateur

Ces paramètres garantissent que le robot et l’ordinateur se trouvent dans le même réseau local, condition indispensable pour qu’ils puissent communiquer entre eux [145]. L’exécution d’un programme de surveillance nous permet d’échanger des flux de données entre l’ordinateur et le robot, comme indiqué par la figure (Figure 4.4). Le délai moyen de la communication établie n’est pas supérieur à  $3ms$  et le processus de surveillance n’entraîne pas d’interruption des données, ce qui prouve que la communication est très stable et fiable et que le taux de transmission est élevé.

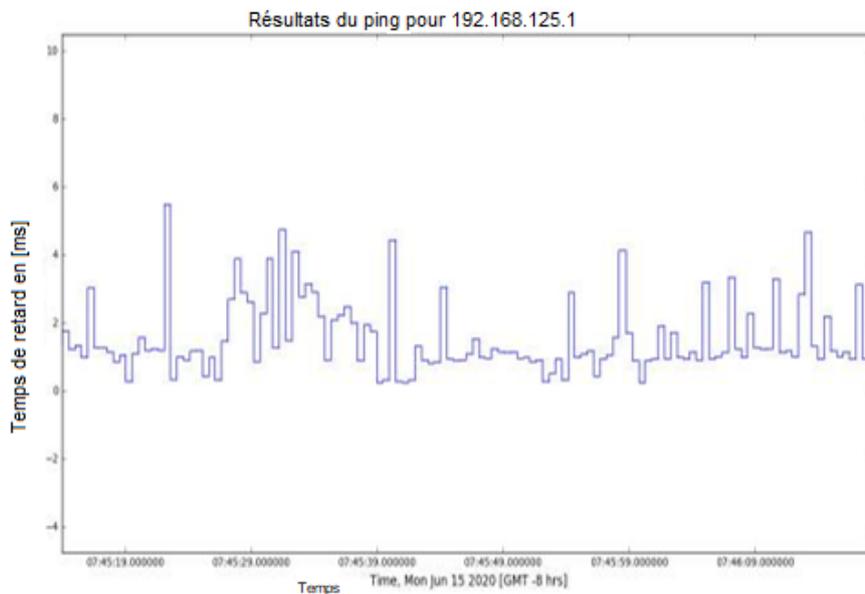


FIGURE 4.4 – Surveillance du retard d’échange de données entre le robot et l’ordinateur

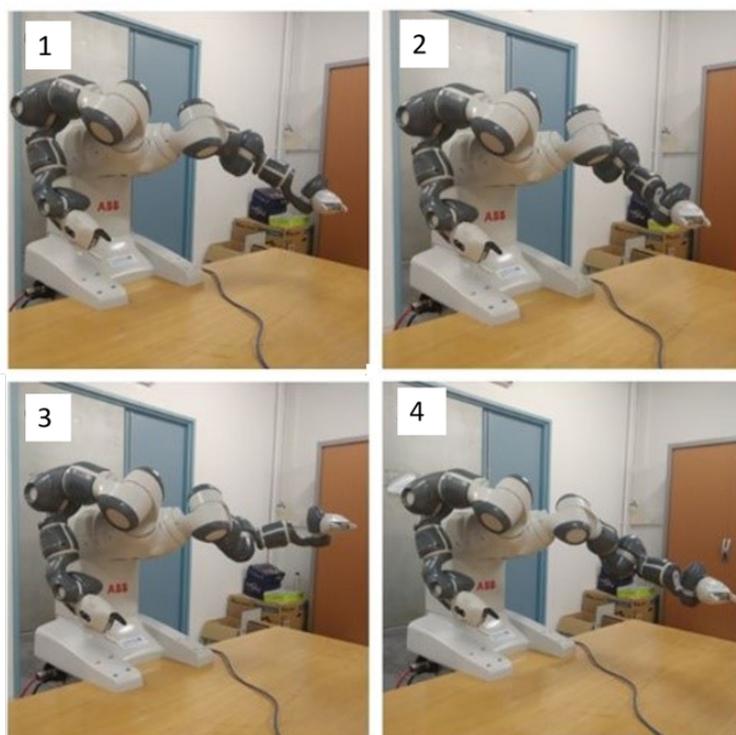


FIGURE 4.5 – Exemple de démonstration du mouvement du bras gauche du robot YuMi, la séquence des mouvements se fait de gauche à droite, de haut en bas sur la figure

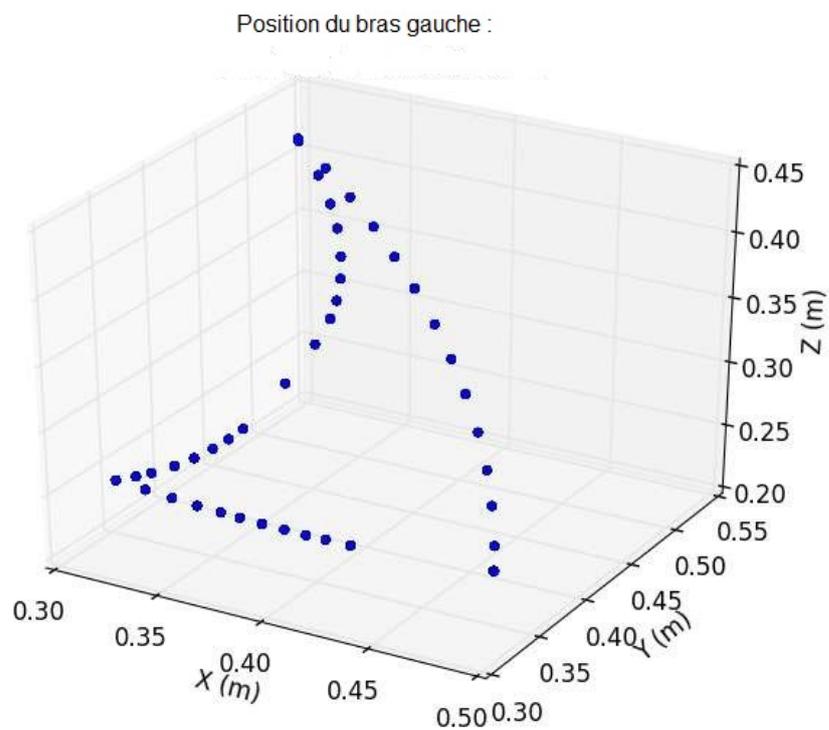


FIGURE 4.6 – La trajectoire de l'effecteur dans l'espace tridimensionnel

Pour tester le bon fonctionnement de l'interface, nous avons fait en sorte que le bras gauche de YuMi suive une trajectoire pour se déplacer dans les directions X, Y et Z, puis revient au point initial dans le sens de la diagonale (Figure 4.5), et la position de l'effecteur correspondante mesurée est illustrée dans la figure (Figure 4.6).

Nous avons été amenés à modifier les programmes de l'interface Yumipy pour les adapter à nos besoins, notamment pour la mesure des couples articulaires. Le couple de l'articulation sept n'était pas mesurable et la mesure de la position de l'axe était toujours constante (erronée). Or, comme nous l'avons détaillé dans le chapitre précédent, nous avons besoin de récupérer impérativement les positions et les couples articulaires exactes du robot.

## 4.4 Conclusion

En absence d'une communication en temps réel stable et sans interruption, il était impossible de mettre en œuvre notre commande sur le robot. Nous vous avons présenté dans ce chapitre un état de l'art sur les types de communication réalisables sur un robot, allant du remplacement du contrôleur initial jusqu'au protocole de communication sans modification du robot (procédure de téléopération). En effet, dans un premier temps, nous avons expliqué et détaillé la méthode utilisée pour réaliser la communication avec notre robot YuMi.

Cette communication est basée sur le protocole TCP/IP. Pour y parvenir, nous avons procédé à une interface client-serveur. Étant donné que le robot n'est pas apte à communiquer qu'avec le protocole TCP, une interface intermédiaire appelée Yumipy a été utilisée pour établir la communication. Nous avons établi des modifications sur l'interface Yumipy pour l'adapter à nos besoins ; celle-ci ne permet pas de récupérer tous les couples articulaires. Nous considérons sur cette interface que la partie Python, la partie du ROS, n'a pas été utilisée pour des raisons de version et de documentation insuffisante, de plus, seule cette partie nous est nécessaire pour notre travail de thèse. Les résultats de la communication sont présentés et la fréquence de transmission des données, est suffisamment performante pour établir une loi de commande sur le robot en temps réel. Cette communication nous a évidemment permis de programmer notre loi de commande mise dans le simulateur et de l'implémenter de manière concrète sur le robot. Le chapitre suivant aborde en détail le volet commande.



# Chapitre 5

## Commande du robot YuMi

### 5.1 État de l'art

Pour l'exécution des tâches robotiques nécessitant une interaction avec l'environnement, il est nécessaire de commander la force exercée sur le contact en plus de la position du manipulateur. Cette commande des interactions permet de surmonter les problèmes propres au contrôle basé sur la position, où le robot agit sans force extérieure sur son environnement. En effet, pour les tâches nécessitant un contact entre le robot et son environnement, la commande en position peut être inappropriée. Plus généralement, pour les opérations impliquant une interaction entre un robot et son environnement, un haut degré de précision concernant les éléments en contact est requis. En pratique, cette précision reste limitée et spécifique au robot considéré. Pourtant, la prise en compte de ces forces lors de la réalisation de la tâche permet de modifier le comportement du robot, d'agir sur ses mouvements et d'améliorer ses capacités.

La première stratégie de commande de base pour exercer une force désirée sur l'environnement est la commande par raideur [121]. Le gain proportionnel dans la commande de position définit la raideur du robot, qui doit être correctement réglée en fonction de la raideur de l'environnement. La commande de la rigidité est utilisée pour concevoir un comportement d'interaction statique. Pour parvenir au comportement dynamique recherché de l'interaction, il faut tenir compte de l'inertie et de l'amortissement au niveau du contact, outre la rigidité, ce qui mène au contrôle par impédance [76]. Selon [28], le contrôle d'impédance est appliqué au problème de la coordination, en particulier à la co-manipulation. Cette étude présente un schéma qui englobe une stratégie centralisée de commande d'impédance, visant à obtenir un comportement flexible au niveau de l'objet manipuler et également une commande d'impédance décentralisée, appliquée au niveau de l'effecteur terminal, pour réduire les forces internes de l'objet. Le schéma de commande est basé sur une configuration à deux boucles, une boucle de position interne simple proportionnelle-intégrale-dérivée et une boucle externe, utilisant les mesures de force et de moment de l'effecteur du robot pour imposer les comportements d'impédance souhaités. En 2008, José de Gea a proposé la mise en œuvre d'une méthode de commande d'impédance cartésien pour réguler les forces d'interaction entre un bras robotisé et l'environnement. Ce type de commande permet de régler le comportement dynamique du robot manipulateur lors de la mise en contact de l'environnement ainsi que le contrôle de la raideur et de l'amortissement de l'interaction plutôt que celui des forces de contact [65]. L'étude dans [86] aborde une loi de commande appelée commande d'impédance synchrone de l'espace cartésien (CSSIC) appliquée à un système de plusieurs manipulateurs. Cette dernière combine la commande synchrone et la commande en impédance. Elle consiste à regrouper les mouvements des manipulateurs saisissant le même objet avec et sans perturbation afin d'éviter la chute de l'objet en cas d'obstacle

ou de rupture provoquée par erreur. Une proposition de mise en œuvre d'une commande d'impédance sur un système à deux bras a été développée [99]. Cette dernière considère les deux manipulateurs comme étant un seul manipulateur, en effectuant une ABT (Asymmetric bimanual task) à l'aide de la technique de la jacobienne relative. Avec cette méthode, la description de tâche devient plus simple et intuitive grâce au mouvement relatif entre les deux effecteurs. Un autre moyen de faire interagir le robot avec son environnement serait la commande hybride position/force [119]. Elle permet d'utiliser à la fois les informations de force et de position pour satisfaire les contraintes de position et les contraintes de force sur les trajectoires. Les directions dont la position est contrainte sont contrôlées en force, et celles dont la force est contrainte sont contrôlées en position (Figure 5.1). Dans cette commande, la force de l'effecteur est déterminée en agissant directement sur les couples des articulations du robot en fonction de l'erreur de force. Dans [144], un état de l'art de la commande de manipulateurs robotiques dans des tâches à mouvement contraint a été exposé, dont les différentes méthodes de commande de force ont été examinées en discutant leurs avantages et leurs inconvénients.

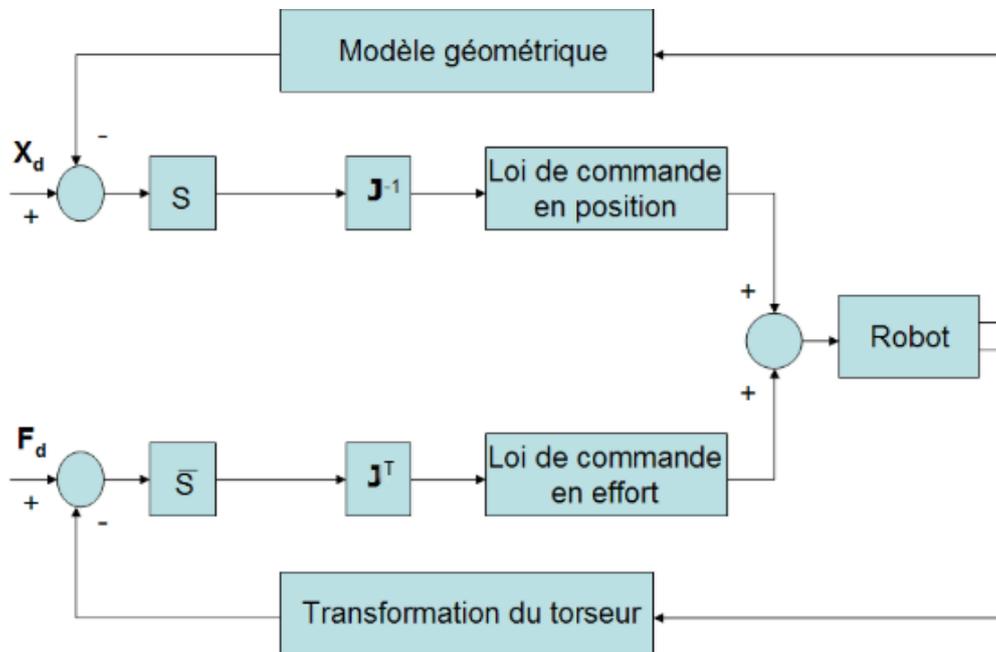


FIGURE 5.1 – Principe de commande hybride position/force

Ces méthodes classiques reposent depuis des années sur le montage de capteurs de force sur l'effecteur final pour détecter la force. Cela a entraîné des coûts importants en termes de prix, de réparation et de maintenance, sans compter d'autres inconvénients tels que le bruit excessif, les directions de détection limitées et l'état instable également provoqué par les bandes passantes étroites des informations de force [50] [46]. À ce propos, la recherche a réussi à développer des algorithmes de commande pour supprimer certains outils comme les capteurs, afin de simplifier l'ensemble du mécanisme et d'en réduire le coût. En effet, les observateurs de perturbations représentent une méthode répandue pour éliminer les capteurs de force.

Au cours des vingt dernières années, les observateurs de perturbations ont fait l'objet de divers travaux de recherche et ont été développés dans de nombreuses applications. Citons par exemple dans [90], un observateur de perturbations pour le contrôle de la force d'un manipulateur à 6 DDL a été décrit et approuvé par des résultats expérimentaux sur la base d'une comparaison avec un capteur de force. En outre, dans [96], une analyse de la stabilité d'un

observateur de perturbations subissant des variations d'inertie a été menée. Ces observateurs sont, pour la plupart, conçus sur la base de systèmes linéaires.

Ces observateurs sont conçus soit sur la base de systèmes linéaires, soit sur la base de systèmes non linéaires. En effet, dans le premier cas, il convient de compenser toutes les non-linéarités avant de les intégrer [108]. Dans le deuxième cas, en revanche, les travaux de conception d'observateurs de perturbations sont basés sur des systèmes non linéaires utilisant un modèle précis de frottement et de dynamique du robot. Une estimation de la dynamique du robot dans différentes phases sans forces de contact, puis l'utilisation des observateurs de perturbations avec le modèle précis identifié pour contrôler la force de contact a été exposé dans [126]. Les observateurs de perturbations permettent d'estimer la force externe comme une perturbation, mais présentent des difficultés de traitement des incertitudes [107].

Une autre approche est possible, il s'agit de la commande adaptative. De nombreux chercheurs l'ont utilisé pour des critères de contrôle de mouvement. On retrouve dans [41] une procédure d'estimation adaptative des perturbations basée sur le suivi de position. Son principe est donc la compensation de la force perturbatrice à l'aide d'une loi adaptative basée sur la position qui met à jour la force estimée de la loi de contrôle. En outre, une combinaison d'un filtre de Kalman étendu pour estimer les états et d'une loi adaptative pour estimer la force dans le contrôle de la position du manipulateur est proposée dans [89]. En [46], une procédure de contrôle hybride force/position sans capteur a été développée. Celle-ci utilise un observateur de force adaptatif pour estimer les forces externes du manipulateur dans une boucle de contrôle interne de la position/vitesse et une boucle de contrôle adaptative de suivi de la force externe qui estime la conformité de l'environnement et modifie la trajectoire du manipulateur dans les directions contrôlées par la force. Toutefois, ces observateurs de force sont peu utilisés pour le contrôle de la force et ils estiment une architecture ouverte de contrôleur de robot.

L'application des méthodes complètes citées dans la bibliographie sur un robot industriel, c'est-à-dire un comportement qui nécessite de réagir avec les couples-moteurs du robot et de les contrôler, nécessite un contrôleur de robot ouvert peu fréquent sur le marché, l'ouverture préalable du contrôleur n'est pas forcément souhaitable sur un robot onéreux ou encore l'ajout des logiciels pour gérer le comportement du robot [29]. Les travaux utilisant des lois de commande classiques sont pour la plupart développés sur des robots prototypes ou sur des robots avec un contrôleur à architecture ouverte. Cela permet d'interagir avec les couples-moteurs, de connaître tous les paramètres et d'avoir accès au générateur de trajectoire.

En revanche, ces études n'abordent pas le cas des architectures relativement fermées, ce qui est le cas de tous les robots industriels sur le marché aujourd'hui [21]. Par exemple, les principaux fabricants de robots utilisent aujourd'hui leur propre langage de programmation, comme le langage RAPID chez ABB, KRL chez KUKA ou encore TPE chez FANUC. Ces langages de programmation utilisent un modèle géométrique du robot et les fonctions spécifiques des autres modèles cinématiques et dynamiques utilisés par le contrôleur ne sont pas accessibles par le langage de programmation classique. En réalité, seule la position est contrôlée sur un robot industriel, c'est-à-dire que pour programmer une trajectoire entre différentes positions, il faut définir différents points de référence. Le repère de base et le repère effecteur sont intégrés par défaut dans le contrôleur du robot et toute position définie est référencée au repère de base, ce qui permet au contrôleur de calculer en temps réel la trajectoire de position et de vitesse pour générer un mouvement entre les deux positions en question. Les couples à envoyer aux moteurs du robot sont obtenus en temps réel en mesurant la différence entre la trajectoire souhaitée et la trajectoire réelle.

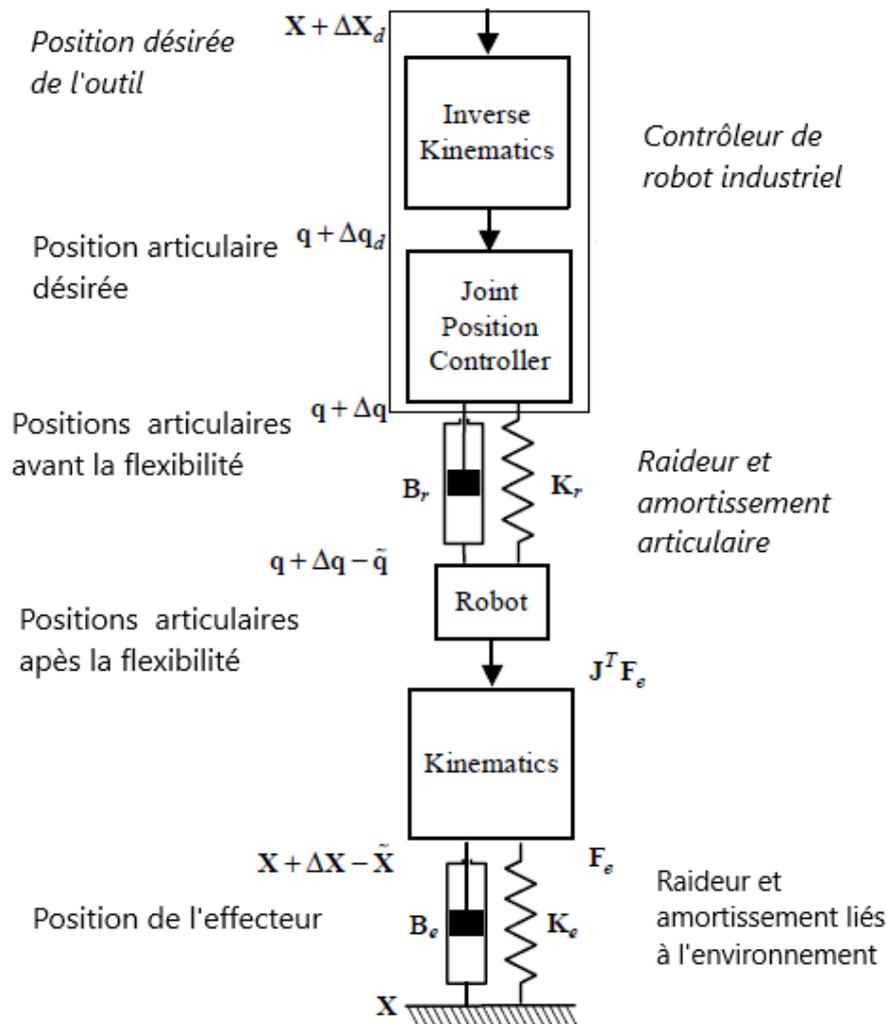


FIGURE 5.2 – Mouvement d'un robot en contact avec l'environnement [20]

Malheureusement, les commandes de force implémentées sont basées sur une structure en cascade, où la commande de position originale, non modifiable par l'utilisateur, est une boucle interne de la commande de force [101]. Ce problème constitue une véritable entrave au développement d'applications d'interaction, d'où la nécessité de développer des lois de contrôle externes. La commande externe développée dans [44] reste une loi de commande valide dans ce cas en utilisant des capteurs de force externes. Dans [20], la conception d'un contrôleur de force pour un robot industriel KUKA est présentée. Le modèle prend en compte la cinématique complète du système et la réponse en position du contrôleur du robot industriel (Figure 5.2).

Dans notre cas, la loi de contrôle proposée est une loi dédiée au dénudage de petits fils nécessitant une faible force; or, comme nous l'avons déjà mentionné, le robot YuMi est un robot industriel dédié à la manipulation de petites pièces. De plus, pour réaliser des tâches robotiques impliquant une interaction avec l'environnement comme le dénudage de fils, il est nécessaire de contrôler la force externe exercée sur l'effecteur du bras en plus de la position du manipulateur. Lorsque le robot est en contact, la position de l'effecteur final est contrainte dans certaines directions par l'environnement, ce qui nécessite un comportement approprié du manipulateur pour faciliter l'interaction et accomplir la tâche. La force que nous voulons

contrôler dans notre cas est la force nécessaire pour tirer le fil sans le détruire, c'est-à-dire la force suffisante pour retirer uniquement la gaine du fil.

## 5.2 Définition de la loi de commande

Nous avons choisi la commande externe position/force en utilisant le modèle dynamique du robot sans utiliser de capteurs externes. Cette dernière consiste à organiser au préalable deux boucles de contrôle imbriquées; une boucle de contrôle de force interne et une seconde boucle de contrôle de position externe. A la différence de la commande hybride classique qui consiste à organiser ces deux boucles simultanément. La figure 5.3 présente le schéma du principe de commande pour un seul bras de YuMi.

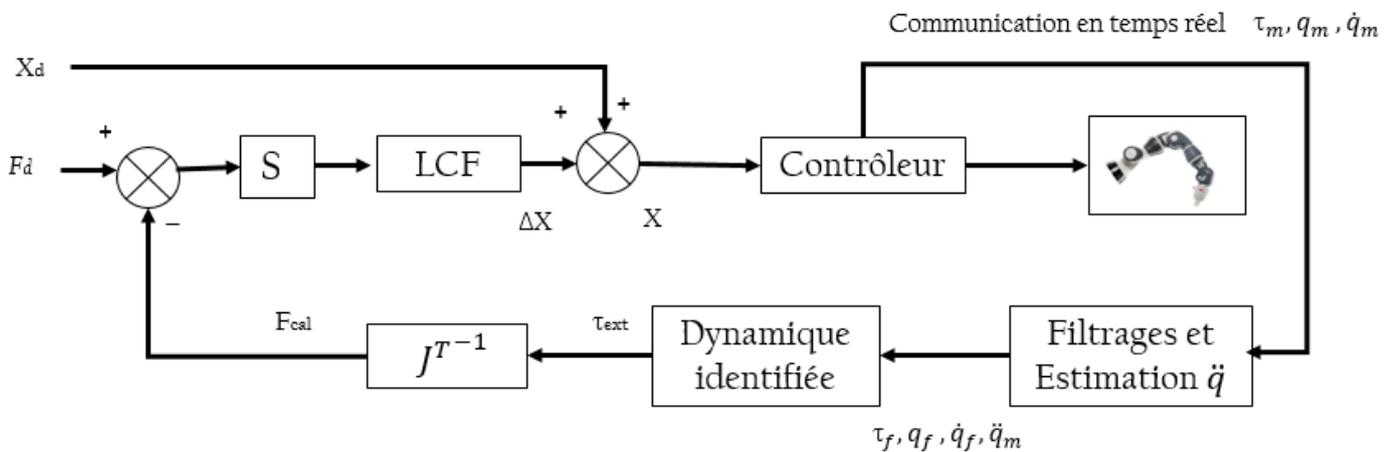


FIGURE 5.3 – Principe de commande proposée pour un bras de YuMi

Les forces externes souhaitées  $F_d$  et les forces calculées  $F_{cal}$  appliquées à l'effecteur du robot sont utilisées pour calculer un écart de position  $\Delta X$  pour ensuite l'ajouter à la consigne de position souhaitée  $X_d$  pour former la nouvelle consigne de position de l'effecteur  $X$  transmise au robot. Grâce à cela, on impose une force désirée constante et ensuite, avec la matrice de sélection  $S$ ; matrice diagonale composée de 1 ou 0, on choisit les directions à commander en force. En effet, si le terme de la matrice est égal à 1, l'asservissement en force de la direction correspondante est réalisé. La variation de position  $\Delta X$  résultante de la différence  $(F_d - F_{cal})$  est obtenue à partir de la loi de commande en force  $LCF$  à définir en fonction de la tâche à réaliser et des résultats souhaités. A la différence de la commande hybride, les contraintes de force n'agissent pas directement sur le couple articulaire des actionneurs mais sur la position opérationnelle du robot. Le grand avantage de ce type de commande est la possibilité de l'appliquer à des architectures de commande relativement "fermées", comme dans le cas de notre robot YuMi. Ce dernier possède déjà son propre contrôleur de position et n'est pas adapté à l'implémentation d'un autre contrôle de position.

Cela nous permet de tirer parti du contrôleur du robot ainsi que de la nouvelle commande externe. La complexité de cette loi peut donc être particulièrement élevée en fonction du modèle dynamique identifié du robot, du filtrage de données et de la fréquence des mesures, c'est-à-dire



mande proposée.

Dans des cas réels, il y a forcément des éléments de distorsion qui s'ajoutent aux forces réelles. Pour cela, nous avons reproduit ces phénomènes en ajoutant un bruit à la force calculée par les blocs d'environnement (Figures 5.7 et 5.8). Un bruit blanc à bande limitée permettant de générer des nombres aléatoires normalement distribués. Les résultats de la simulation obtenus sont bien meilleurs au niveau du contrôle de la force ainsi qu'au niveau des positions. Nous pouvons donc en déduire qu'avec la méthode de contrôle proposée, nous avons réussi à contrôler les forces externes appliquées aux robots et qu'il nous reste à la valider concrètement sur le robot réel.

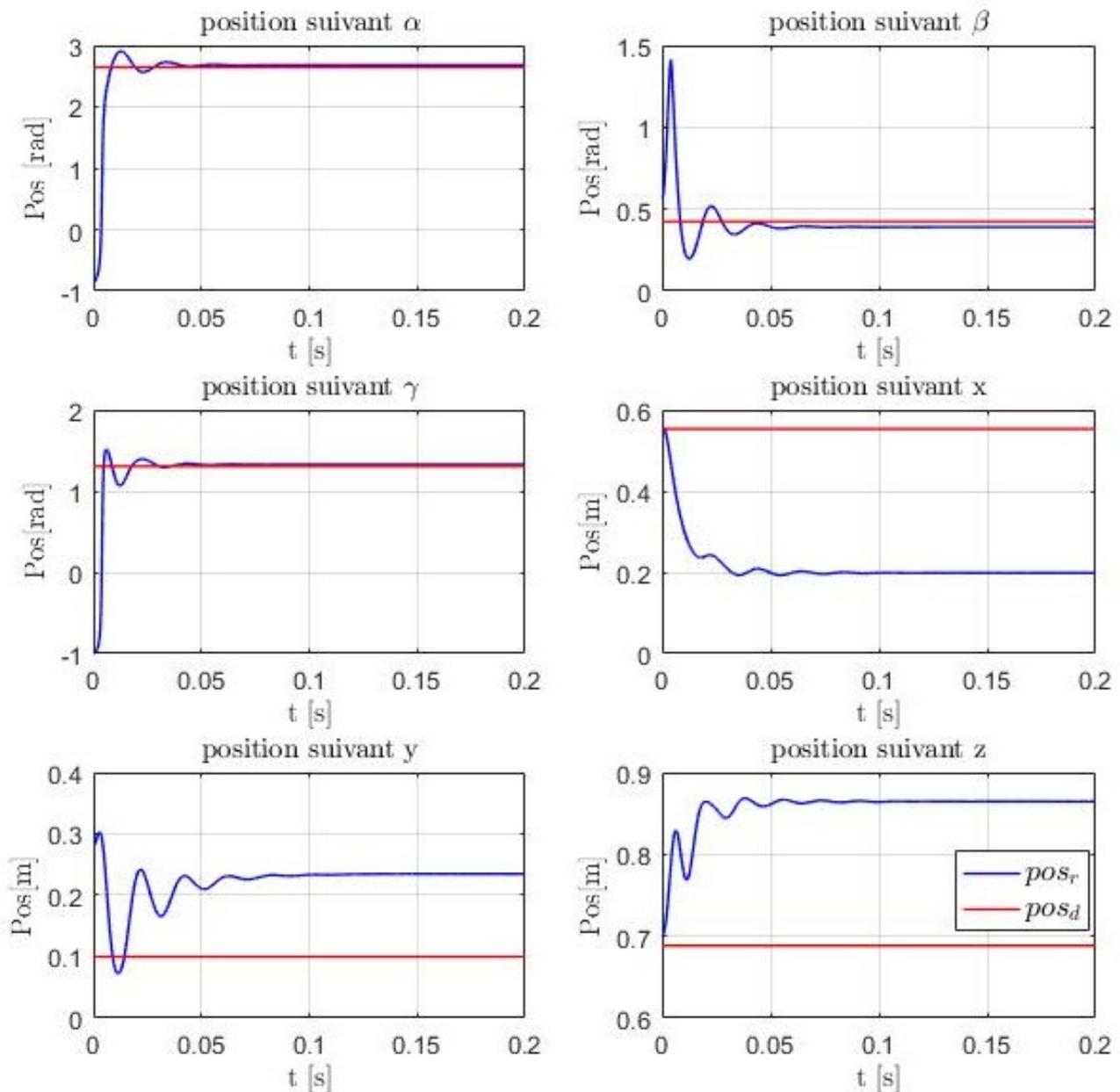


FIGURE 5.5 – Positions désirées et réelles du simulateur sans bruit

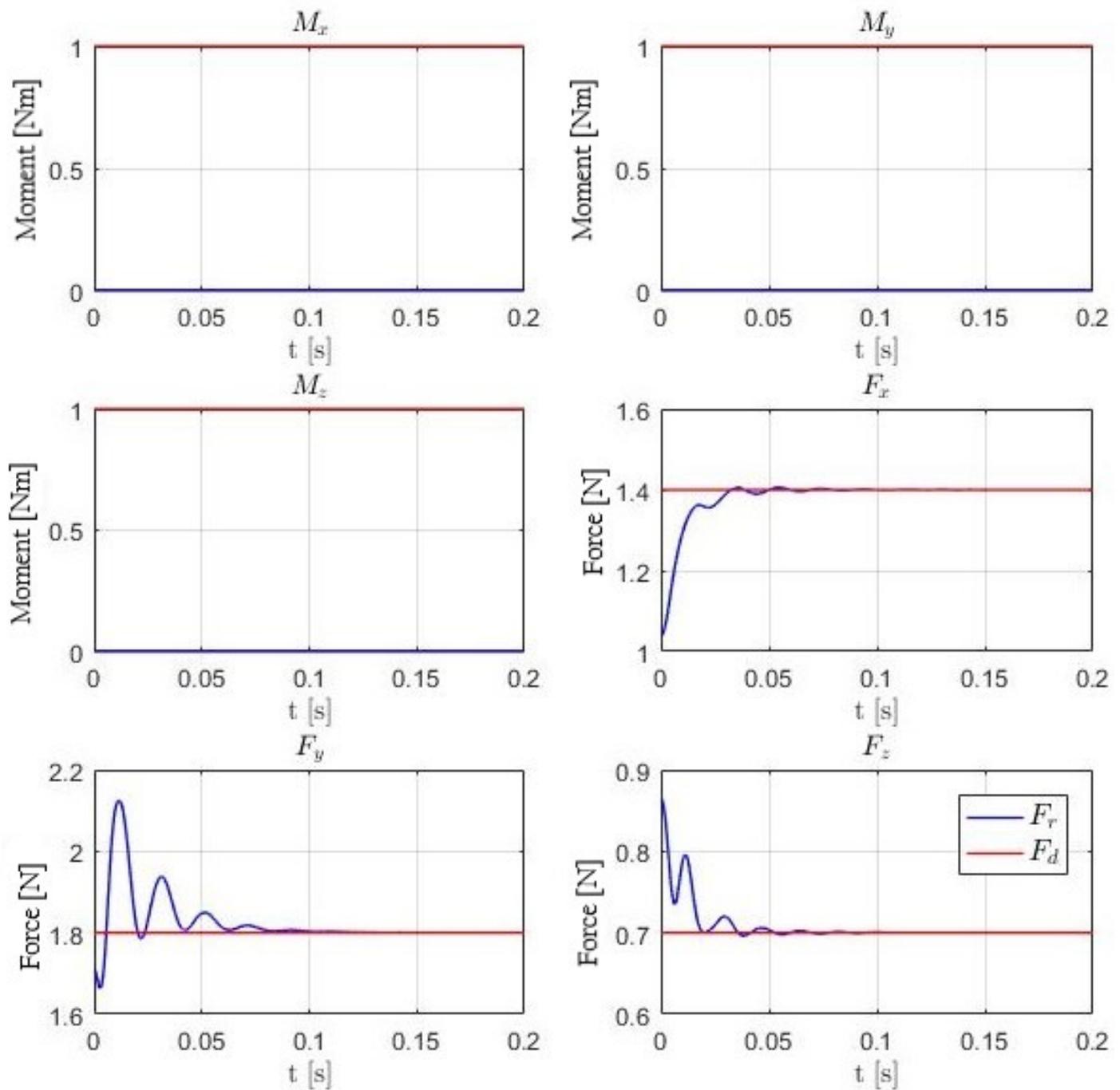


FIGURE 5.6 – Forces et moments désirés et réels du simulateur avec bruit

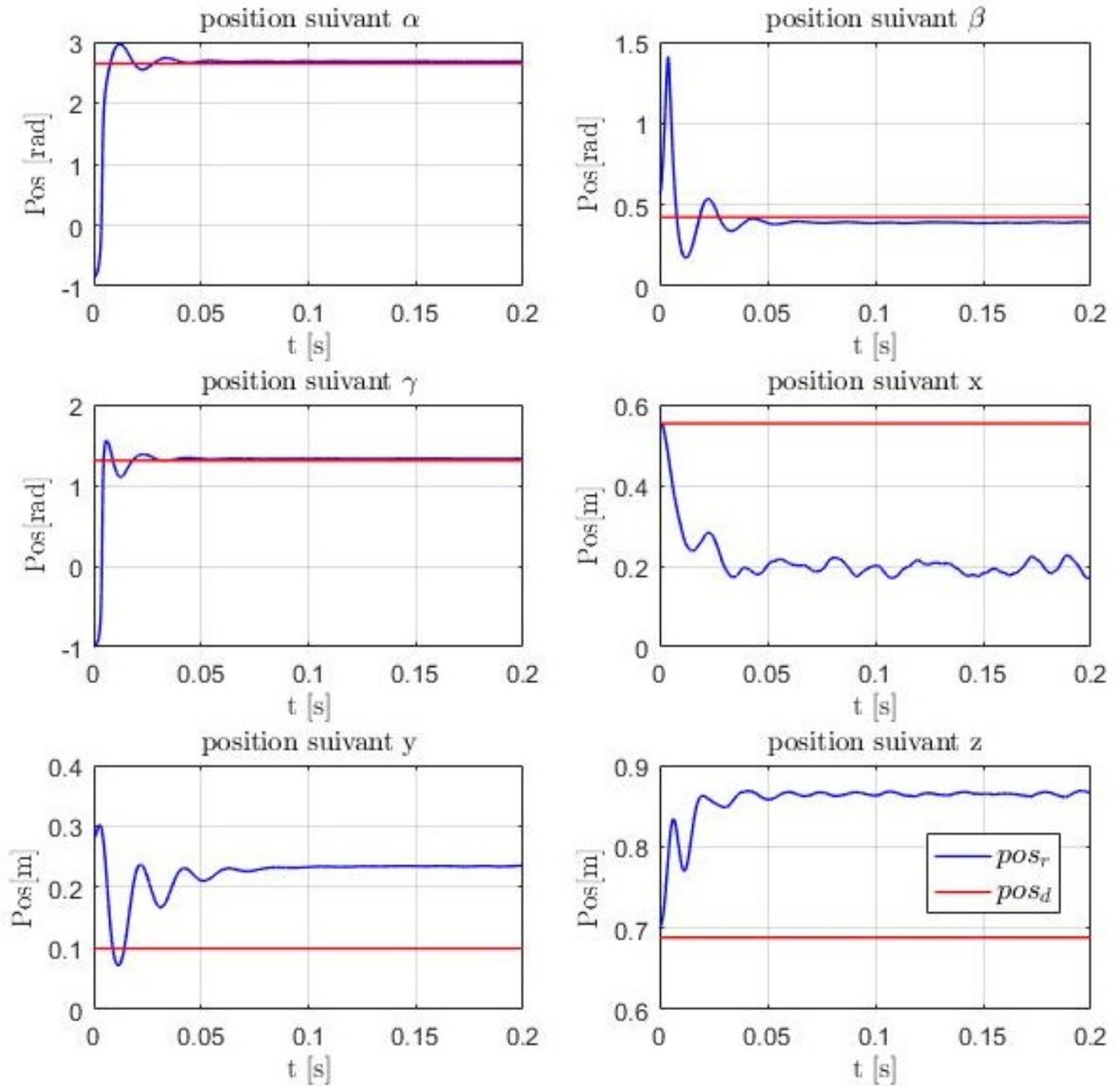


FIGURE 5.7 – Positions désirées et réelles du simulateur avec bruit

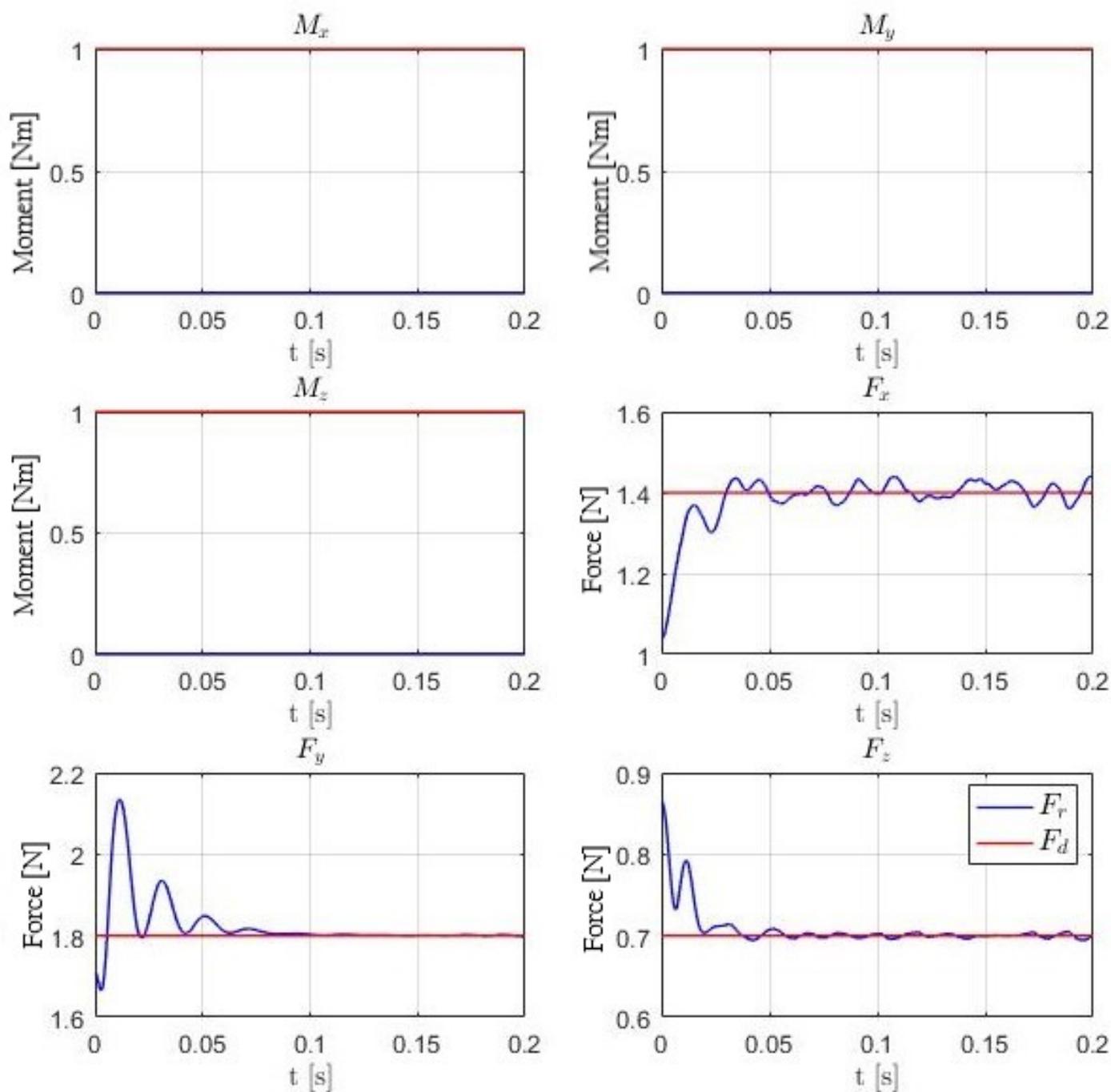


FIGURE 5.8 – Forces et moments désirés et réels du simulateur avec bruit

## 5.4 Résultats expérimentaux de la commande sur YuMi

Dans notre commande, les écarts de position ajoutés à la consigne de position résultent d'une loi de contrôle des forces. Le correcteur retenu peut se révéler d'une complexité variable en fonction du résultat souhaité et des capacités de traitement disponibles auprès de l'ordinateur, de la communication en temps réel et du contrôleur du robot. Lors des essais réalisés sur le robot, nous avons opté pour un contrôleur de force  $PID$  et nous avons analysé le comportement du robot en prenant en compte chacun des paramètres de ce correcteur.

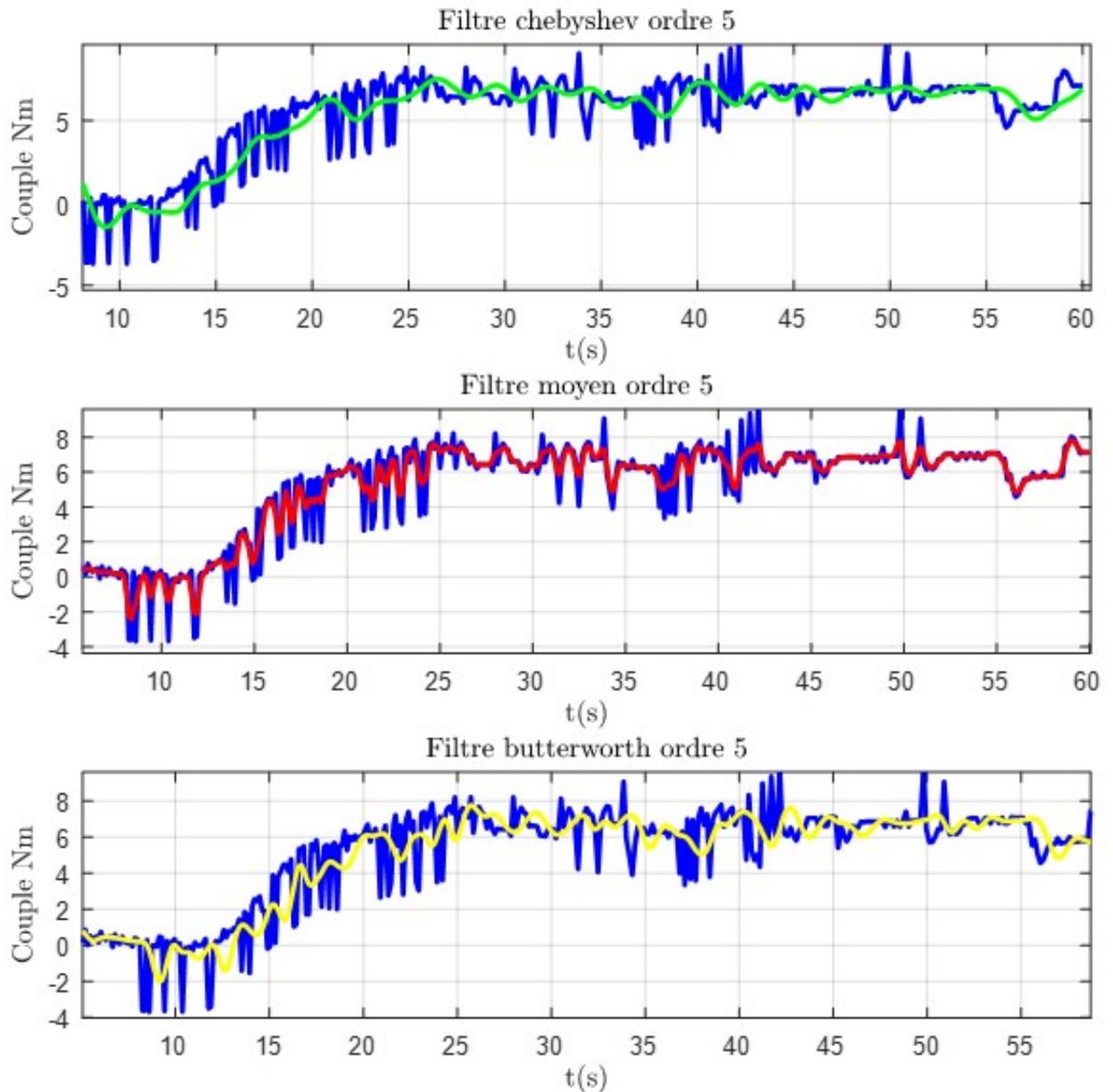


FIGURE 5.9 – Couple d'axe 1 avec différents type de filtrage

Avant de procéder à l'expérimentation de la commande, une étape cruciale pour assurer la stabilité du système est la phase de filtrage. En effet, les mesures reçues par la communication sont excessivement bruitées. Cela nous a amené à filtrer ses données en temps réel. Par conséquent, nous avons testé les résultats du contrôle avec trois différents types de filtrage : le filtre moyen d'ordre 5 numérique, le filtre de Chebyshev type 1 et le filtre de Butterworth dont les réponses en gain (ou en amplitude),  $G_n(\omega)$  sont respectivement Eq. 5.2 et Eq. 5.1. La figure 5.9 présente les résultats du filtrage du couple du premier axe du bras droit du robot.

Filtre Butterworth :

$$G_n(\omega) = |H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}} \quad (5.1)$$

Filtre Chebyshev :

$$G_n(\omega) = |H(j\omega)|^2 = \frac{1}{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)} \quad \varepsilon < 1 \quad (5.2)$$

où  $\omega_0$  est la fréquence de coupure,  $H(\omega)$  est la fonction de transfert du filtre,  $\varepsilon$  est le facteur d'ondulation, et  $T_n$  est un polynôme de Chebyshev d'ordre  $n$  :  $T_n(\omega) = 2\omega \cdot T_{n-1}(\omega) - T_{n-2}(\omega)$  (pour plus de détails, voir l'appendice E).

Les résultats obtenus avec les trois filtres étaient plus stables que ceux obtenus sans filtrage et le bras du robot ne tremblait pas lorsqu'il parvenait à la valeur de force et de position souhaitée. Néanmoins, certains résultats étaient plus performants que d'autres. En effet, nous avons obtenu de très bons résultats avec le filtre Tchebychev d'ordre 5 avec une période d'échantillonnage de 0.03s. Ce filtrage a été donc retenu pour les résultats de commande. Dans ce qui suit, nous exposons différents exemples de résultats de la commande des bras du robot YuMi.

Dans ces expériences, les bras du robot ne comportait pas les pinces, on a simplement installé un outil de poids négligeable dans les bras pour nous aider à fixer le fil à tirer entre les deux bras du robot, ou un bras et contrôler en position suivant toutes les directions ce qui permet de ne pas influencer sur les résultats de l'autre bras contrôler en force suivant la direction désirée. Dans le premier exemple, nous avons défini la matrice de sélection  $S$  de manière à ce que seule la force le long de l'axe  $z$  soit prise en compte dans le calcul de la correction de position ; dans le second exemple, seul l'axe  $Y$  est contrôlé. Pour l'axe  $x$ , nous présenterons les résultats avec l'application de dénudage dans le prochain paragraphe. Dans le simulateur Simulink, nous avons considéré un contrôleur  $PI$ , car nous avons constaté que l'ajout d'un terme dérivé entraîne des dépassements importants, par conséquent nous avons supprimé ce terme du contrôleur ; la même observation a été signalée lors des expérimentations. Par ailleurs, l'introduction du terme intégral améliore considérablement la réponse du système et permet d'obtenir la force désirée, ceci explique la décision de retenir un contrôleur  $PI$ .

### Premier exemple :

Dans cet exemple, seule la force le long de l'axe  $z$  est considérée. Lors de l'expérience, le bras droit du robot doit tirer sur un fil relâché jusqu'à ce qu'il parvient à le tendre et à appliquer la force souhaitée. La position et les forces souhaitées sont les suivantes :

La figure 5.10 on a  $X_d = [0, 300 \ 0, 160 \ 0, 180 \ 0, 067 \ -1, 23 \ -1, 6]$  et  $F_d = [-5 \ 4, 5 \ -3 \ 0 \ 0 \ 0]$   
 La figure 5.11 on a  $X_d = [0, 300 \ 0, 160 \ 0, 180 \ 0, 067 \ -1, 23 \ -1, 6]$  et  $F_d = [-5 \ 4, 5 \ -5 \ 0 \ 0 \ 0]$  Dans

cet exemple, pour exercer une force de  $-5\text{N}$  sur le câble, le robot a dû tirer un peu plus que lorsqu'il ne devait appliquer que  $-3\text{N}$ ; celui-ci s'est arrêté après 20cm de décalage de position, alors qu'avec une force de  $-5\text{N}$ , il a continué à tirer au bout de 25cm. Le gain proportionnel est  $K_p = 0,02\text{m}\cdot\text{N}^{-1}$  et le gain intégral est  $K_i = 5\cdot 10^{-5}\text{m}\cdot\text{s}^{-1}\cdot\text{N}^{-1}$ .

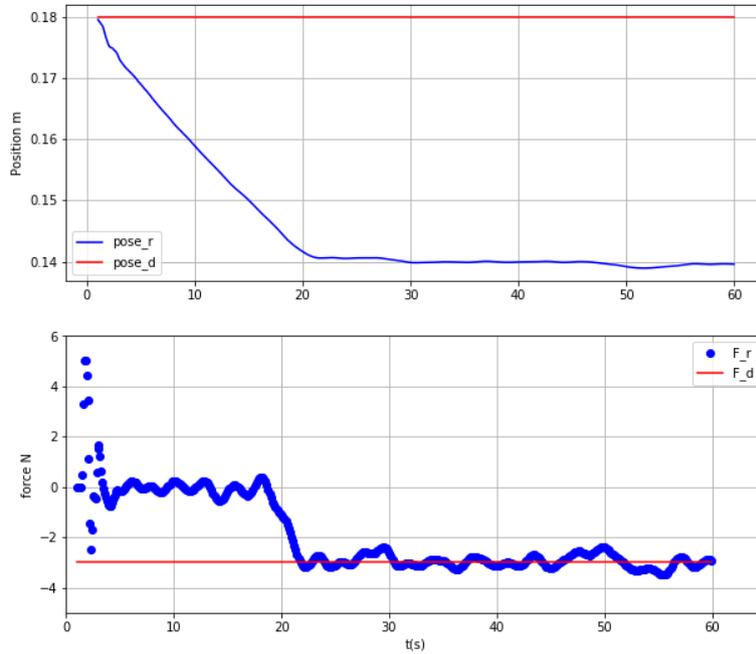


FIGURE 5.10 – Position désirée et réelle du robot selon  $z$

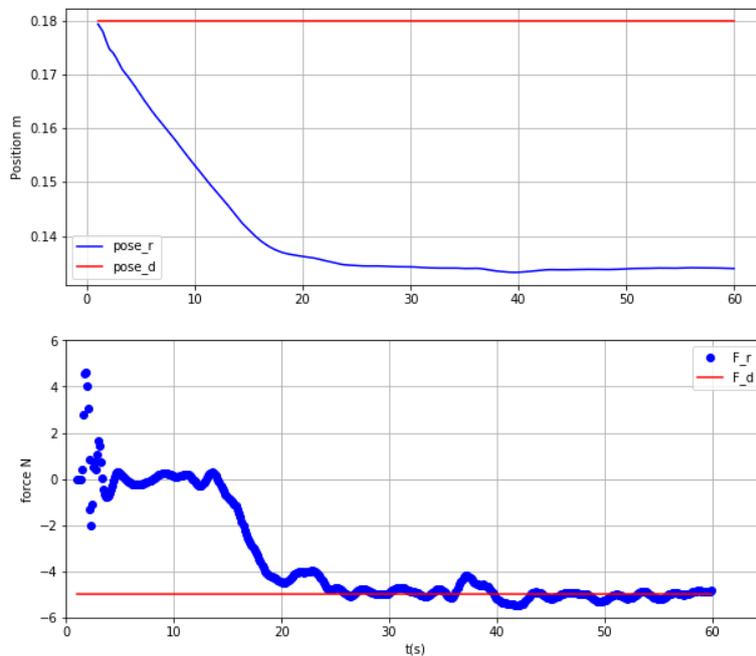


FIGURE 5.11 – Force désirée et réelle du robot selon  $z$

Deuxième exemple :

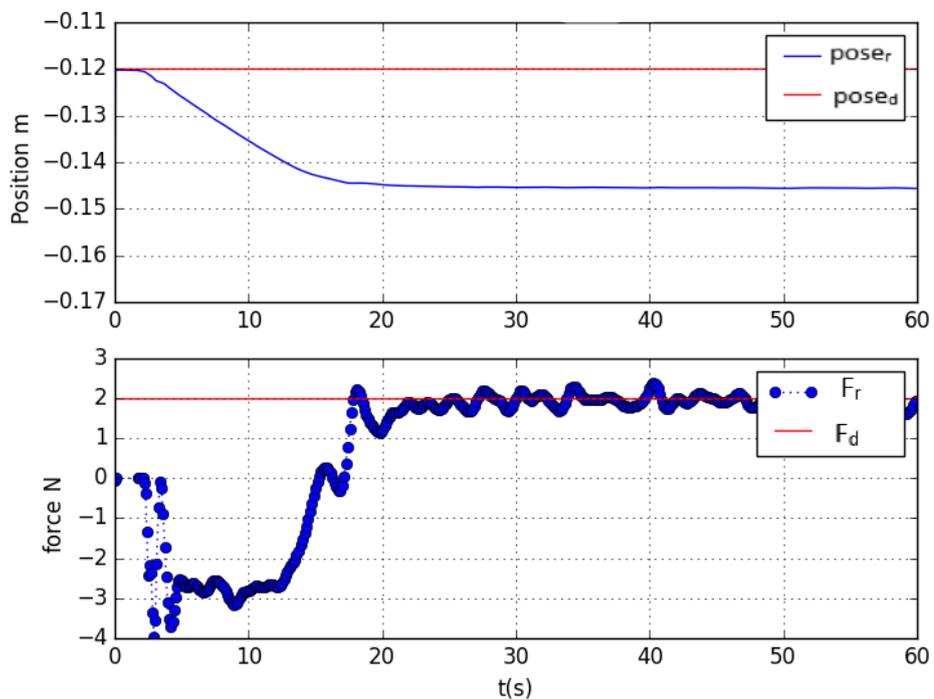


FIGURE 5.12 – Position désirée et réelle du robot selon  $y$

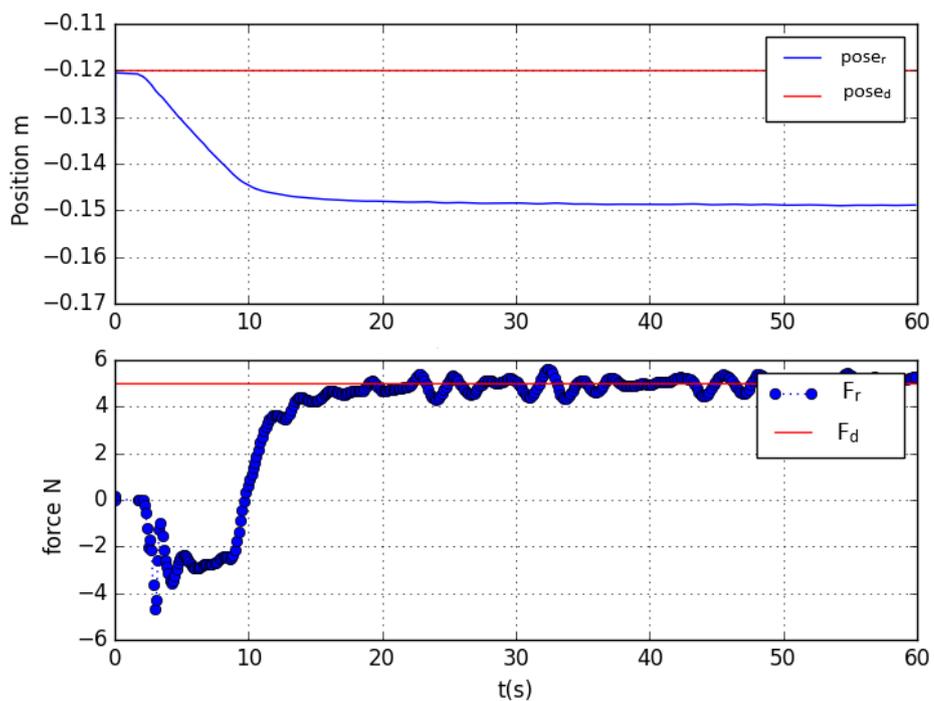


FIGURE 5.13 – Force désirée et réelle du robot selon  $y$

Dans le deuxième exemple, la force suivant  $y$  est commandée pour une valeur désirée de 2N et 5N, on constate également que pour atteindre la force 5N le bras du robot a été obligé de tirer plus que dans le cas de 2N de 10mm de plus pour atteindre la force 5N.

## 5.5 Application au dénudage

### 5.5.1 Technique de dénudage adoptée

Afin de dénuder un fil électrique, il est nécessaire d'exécuter au préalable une technique adéquate afin de respecter certaines contraintes. En effet, il s'agit par exemple de ne pas endommager l'âme conductrice, qui pourrait alors se rompre lors de la manipulation et provoquer des dysfonctionnements, et de ne pas endommager la gaine de protection sur la longueur restante du câble pour éviter de provoquer des courts-circuits. Ceci étant, le dénudage s'étale sur plusieurs étapes [105] à savoir :

**1<sup>re</sup> étape : le réglage du calibre**, c'est la première étape de l'opération de dénudage qui consiste à adapter l'outil ou la machine au fil que l'on souhaite dénuder. La caractéristique importante à prendre en compte ici est le plus souvent la section du fil.

**2<sup>me</sup> étape : le serrage du fil**, cette étape est relativement simple puisqu'elle consiste à placer le fil entre les mâchoires de l'outil de dénudage. Lors de cette étape, il est également important de s'assurer que la force de serrage n'endommage pas le capot de protection.

**3<sup>me</sup> étape : l'application de la force pour segmenter l'isolant**, il s'agit d'une étape importante, car c'est ici que tous les paramètres doivent être pris en compte. L'espace laissé entre les pinces de l'outil doit être suffisant et précis pour que la coupe n'endommage pas l'âme conductrice. De même, la segmentation obtenue doit être suffisante pour faciliter le retrait de l'isolant à l'étape suivante.

**4<sup>me</sup> étape : le retrait de la gaine de protection**, c'est la dernière étape de l'opération de dénudage, qui consiste à retirer l'isolant d'une portion donnée de fil. Les outils de dénudage automatiques et les machines industrielles permettent de retirer l'isolant sur la partie concernée directement après la segmentation en une seule action. Pour les outils manuels, elle doit se faire en tirant sur le fil ou en séparant le fil de sa gaine (dénudage).

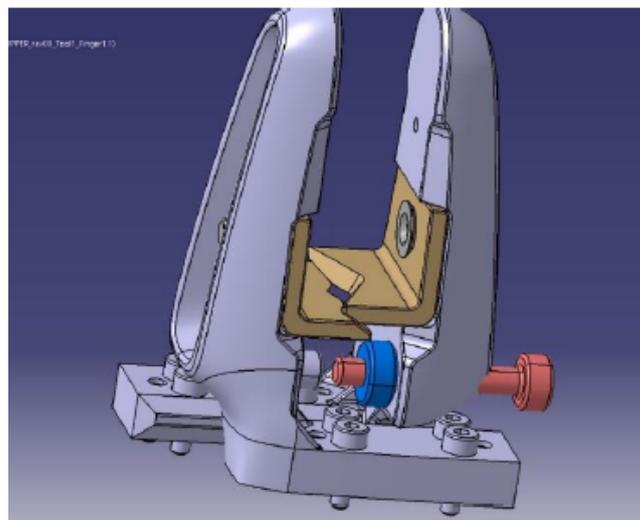


FIGURE 5.14 – Modèle CAO proposé pour la pince avec des lames identiques en forme de V

Dans le cadre de notre projet, nous avons conçu un outillage pour le robot YuMi afin de dénuder des câbles électriques en ayant recours à une technologie adéquate qui prend en compte les contraintes rencontrées dans l'exécution du dénudage par le robot et qui respecte les spécifications techniques requises [137]. Il s'agit donc d'une paire de pinces avec des lames identiques en forme de V (voir Figure 5.14), une solution inspirée des pinces à dénuder classiques qui ont des mâchoires en forme de V et une vis de blocage pour ajuster la section du câble à couper.

Une rainure est faite dans chaque doigt du YuMi pour loger la base de chaque lame. Cette rainure possède des congés de rayon  $r = 0,8\text{mm}$  qui correspond à deux fois la valeur minimale imposée par l'impression 3D. Un trou lisse est également percé dans chaque doigt et dans chaque lame. Le maintien en position est assuré par une vis FHC M3x8 et un écrou M3. Ce type de vis a été préféré pour des raisons d'encombrement parce que cette vis peut être totalement noyée dans la lame sans poser de problèmes à la conception globale.

Sur chaque doigt, des arrondis de  $r \geq 0,4\text{ mm}$  ont été créés sur toutes les nervures dans le but de supporter les efforts. Une vis de blocage, de type vis moletée M3  $\times$  20, et un écrou moleté M3 peuvent également être aperçus sur un doigt tandis que sur l'autre, un prolongement a été créé dans le but de réduire la longueur de la vis de blocage utilisée et de respecter les standards pour les vis moletées. Le trou créé, qui s'appuie sur un sommet du doigt pour des questions de stabilité et de surface d'appui, est doté d'un taraudage de pas  $p = 0,25\text{mm}$  et de diamètre  $d = 2,5\text{mm}$ . Le système de blocage moleté s'explique ici par le choix du réglage manuel qui a été pris en compte.

Les lames sont issues du pliage d'une tôle de caractéristiques  $L \times l \times e = 25\text{ mm} \times 20\text{ mm} \times 2$ . Un chanfrein a été créé sur la base de la lame pour faciliter son maintien en position dans la rainure du doigt. Une encoche chanfreinée de  $120^\circ$  a également été faite dans la lame pour créer la partie coupante. Les deux lames ont été conçues pour être montées dos à dos. Cela étant, des chanfreins de longueur  $1\text{ mm} \times 60^\circ$  ont été ajoutés sur les extrémités des lames pour apporter un chevauchement sans jeu des lames et leur réglage en position autonome. Ce chevauchement sans jeu est utile car il renforce l'opération de coupe des lames. La figure 5.15 illustre la solution proposée montée sur le robot.



FIGURE 5.15 – Pince proposée montée sur le robot

### 5.5.2 Résultats expérimentaux

Avant de présenter les premiers résultats du dénudage avec la loi de commande, il convient de préciser que nous avons tenté, avec les mêmes expériences, de dénuder des câbles uniquement avec la commande du robot, c'est-à-dire avec un simple programme RAPID commandant au robot d'aller d'une position à l'autre en tenant le câble entre les pinces, mais qu'il s'est avéré que le robot n'était pas en mesure de dénuder et que celui-ci s'est arrêté car il se considère comme étant en collision et, par conséquent, il désactive les moteurs et s'arrête en affichant le message " Surveillance de mouvement ". De plus, nous avons également noté que le robot était capable de soutenir beaucoup plus que la force externe sur ses bras, allant jusqu'à 40 N selon une seule direction, ce qui confirme que les performances d'un robot peuvent être considérablement améliorées par l'ajout de commandes supplémentaires.

Nous abordons désormais les premiers résultats obtenus lors du dénudage. Ces résultats sont le fruit d'une expérience faite avec un fil de 1mm de diamètre que nous avons dénudé avec le bras droit du robot et un outil de serrage qui permet de tenir le fil sans risque de dérapage contrairement à ce qui se passe lorsque nous utilisons l'autre bras étant donné que nous n'avons pas encore développé un outil qui permet de tenir le câble et de bien le dénuder. De plus, en changeant à chaque fois la force de serrage et la force de traction contrôlée par la loi de commande développée sur le robot. Nous avons considéré pour ces essais la même longueur à dénuder de 5mm.

Lors du premier résultat, nous avons appliqué une force de serrage maximale réalisable par le robot de 20N et nous avons commandé le robot à 30N pour la force de tirage. On remarque sur la figure 5.16 que le robot a atteint la force désirée c'est-à-dire 30N, et sur le plan expérimental nous avons constaté que l'étape de segmentation a été faite à 15s et que le retrait de la gaine a lieu 3s juste après ce qui explique la diminution de la force à partir de 15s.

En revanche, dans le second cas, nous avons réduit la force de traction à 23N et la force de serrage à 18N car nous avons remarqué que le dénudage pouvait se faire avec moins de force et la figure 5.17 le prouve car le dénudage est nettement mieux fait que dans l'exemple précédent. Pour les figures 5.18 et 5.19, nous avons donc maintenu les mêmes forces de serrage et nous avons modifié la force de traction pour connaître son influence sur le dénudage. Avec une force de 35N, le robot n'a pas pu atteindre la force et il a dénudé avant avec une force de 30N et pour la segmentation il a eu besoin d'une force entre 15N et 17N pour finaliser l'opération de dénudage.

Dans le dernier essai, nous avons diminué les deux forces. Le dénudage a été parfaitement réalisé avec 18N et 8N respectivement de segmentation et de serrage, la figure 5.20 illustre les résultats. Dans cet essai, le dénudage est effectué en une seule étape avec moins de force à exercer.

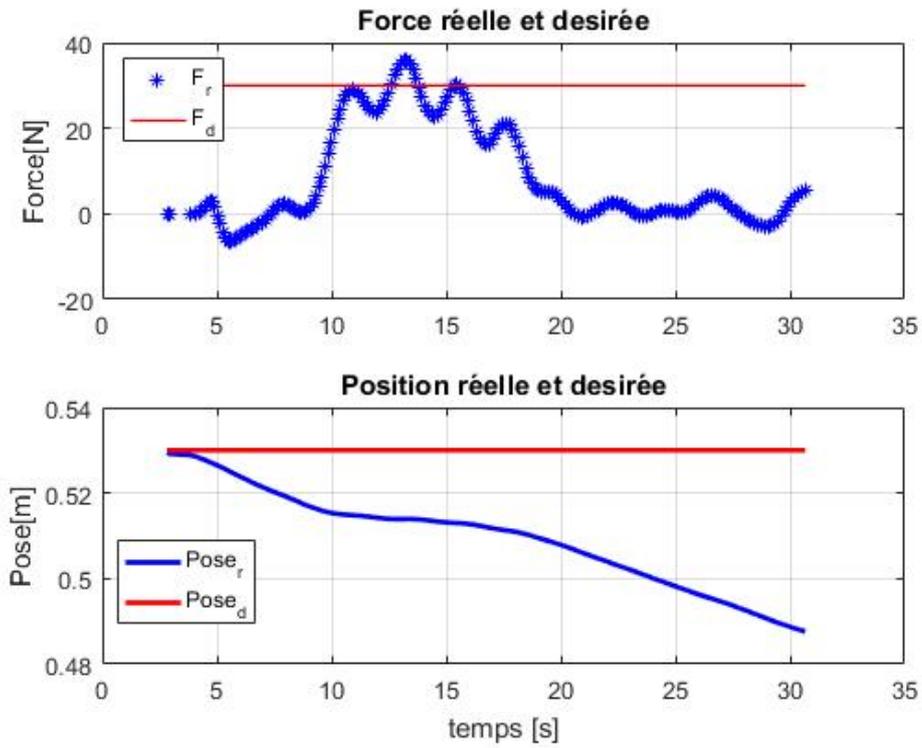


FIGURE 5.16 – Résultats de dénudage avec force de serrage 20N et force contrôlée 30N

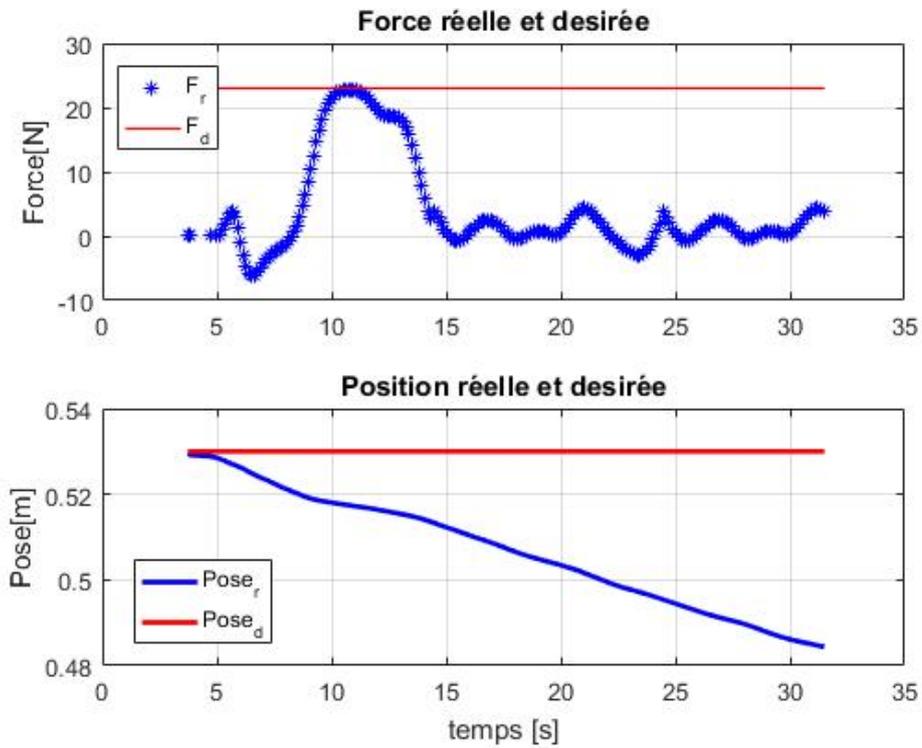


FIGURE 5.17 – Résultats de dénudage avec force de serrage 18N et force contrôlée 23N

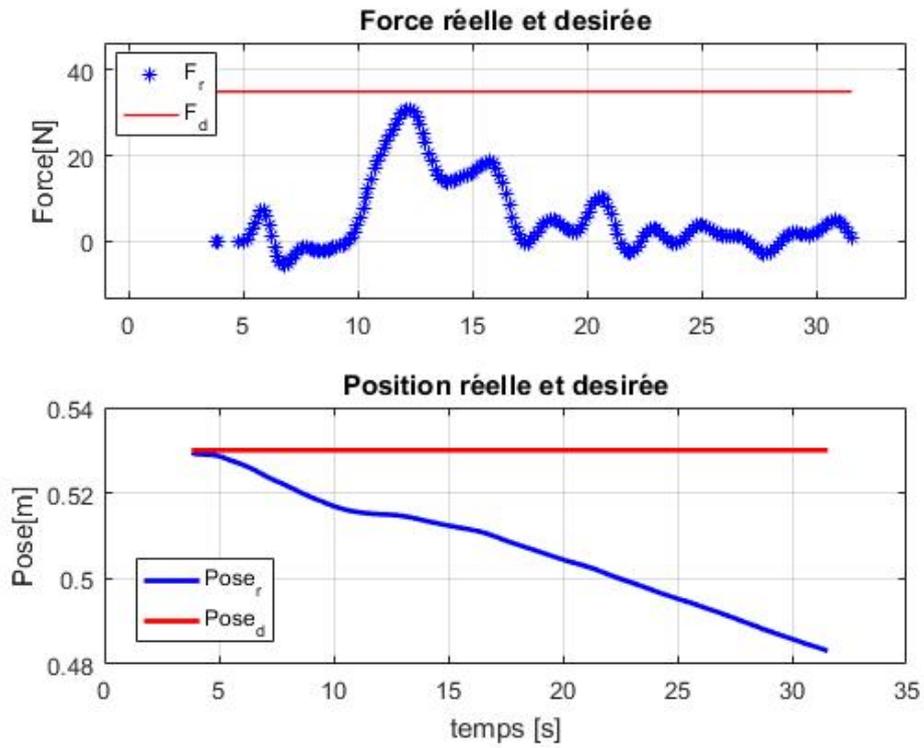


FIGURE 5.18 – Résultats de dénudage avec force de serrage 15N et force contrôlée 35N

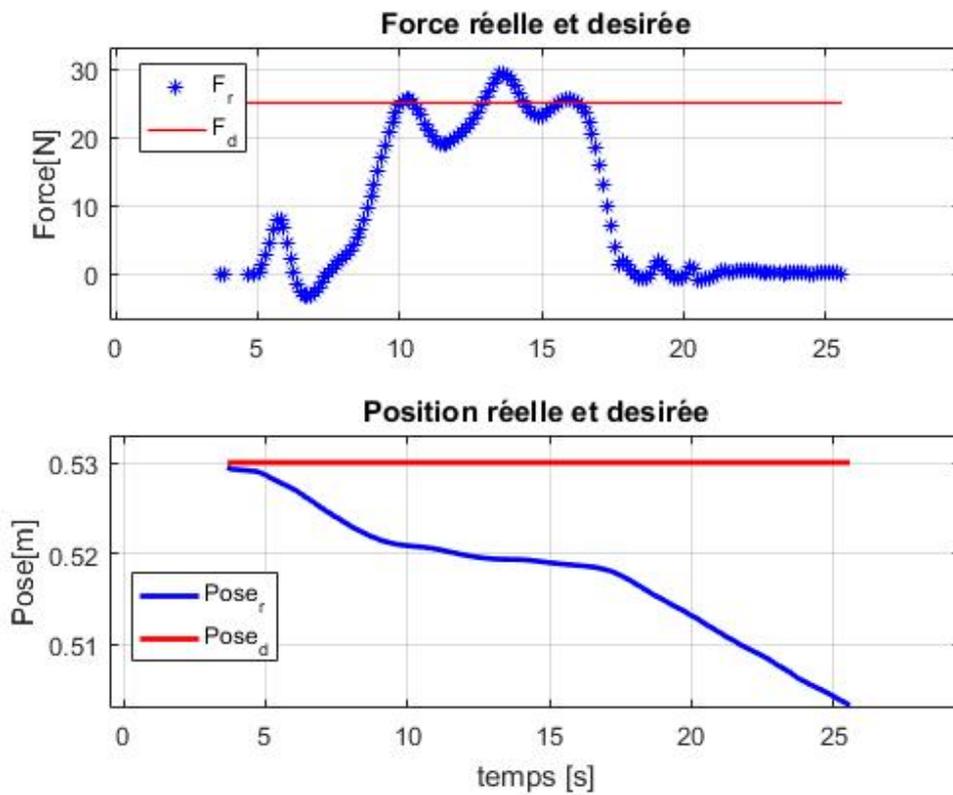


FIGURE 5.19 – Résultats de dénudage avec force de serrage 15N et force contrôlée 25N

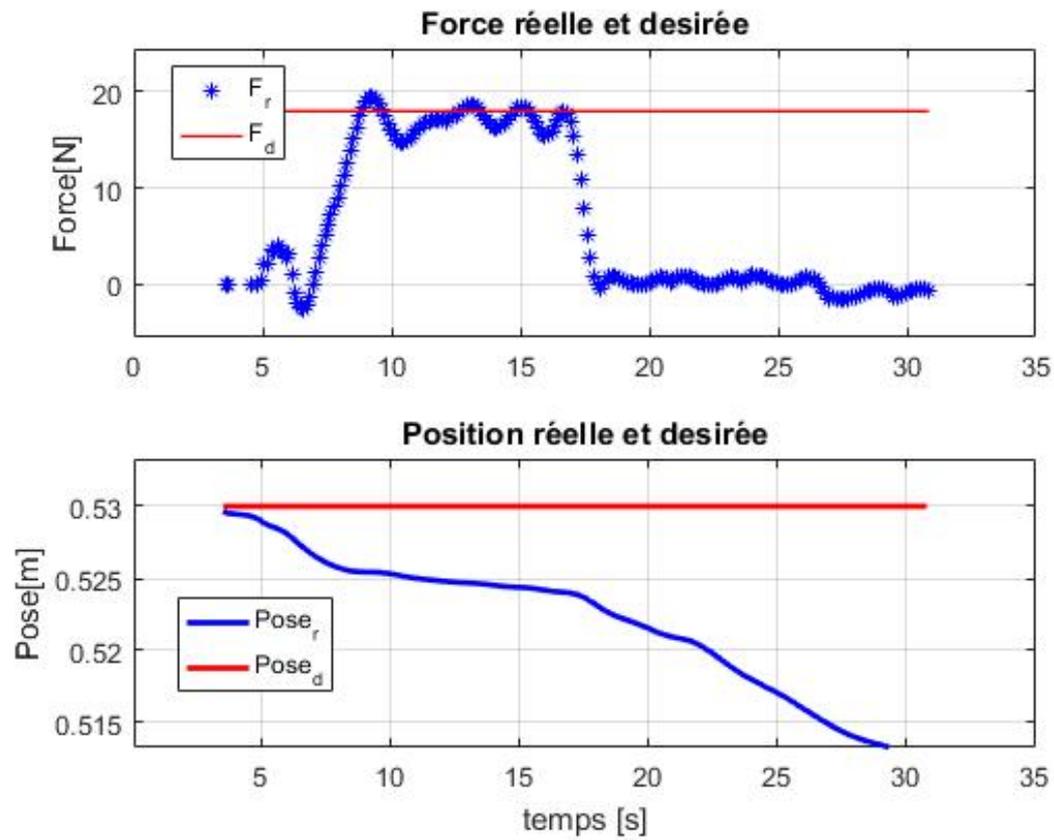


FIGURE 5.20 – Résultats de dénudage avec force de serrage 8N et force contrôlée 18N

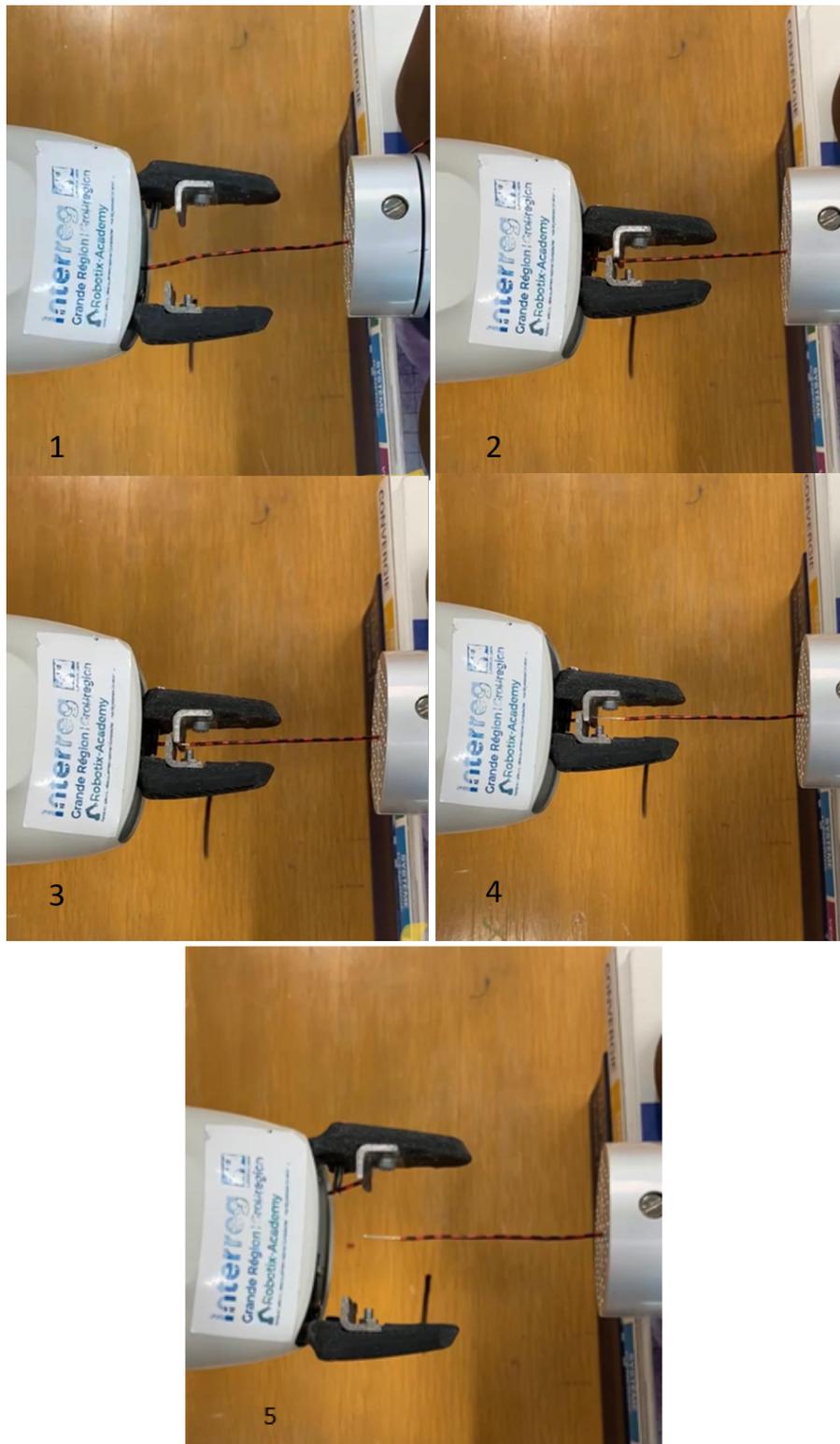


FIGURE 5.21 – Les étapes de dénudage avec YuMi

Les résultats de commande présentés sont réalisés sans avoir ajouté les pinces sur les robots, le fil utilisé n'est pas dénudable, et de plus, l'autre bras tient bien le fil avec un outil adapté, puisqu'il est contrôlé en position. Nous pouvons constater que par rapport à l'expérience de dénudage, la convergence du système est plus performante. Ceci est lié à différents facteurs à prendre en compte dans la perspective de ce travail, y compris le modèle de la force de dénudage. Les premiers résultats de dénudage avec le robot confirment que pour robotiser l'application

du dénudage, il est nécessaire de connaître à l'avance le modèle de force à appliquer pour une longueur et un diamètre spécifique du câble. Nous avons donc à la fin de cette thèse, proposé un modèle de force permettant de connaître à l'avance les deux forces à appliquer sur le câble pour voir un dénudage réussi et qui respecte les normes définies.

Pour y parvenir, nous avons choisi un simple fil électrique. Pour dénuder ce fil, il faut maîtriser les paramètres suivants (Figure 5.22) :

- $l$  la longueur de la partie de fil électrique à dénuder .
- $d_e$  la profondeur de la partie à dénuder.

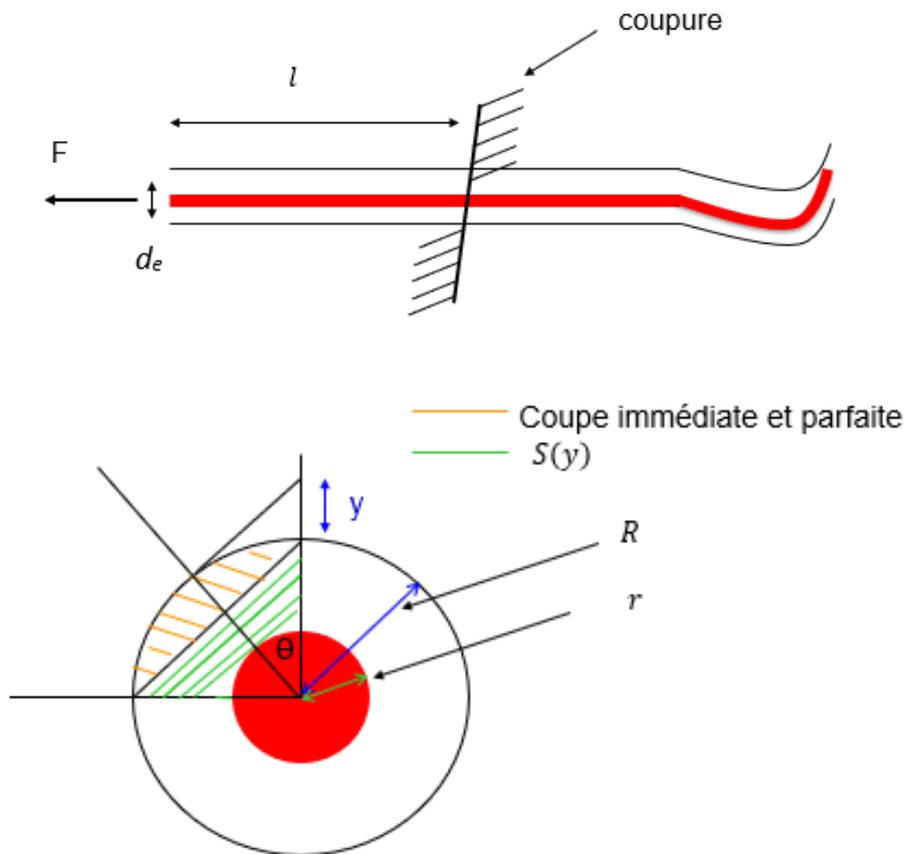


FIGURE 5.22 – Position du plan de coupure sur le fil

De ce fait, on a établi le modèle de force de dénudage suivant :

$$F = S(y)\sigma + f.\pi.d.l.\tau \quad (5.3)$$

Avec :

- $\sigma$  : contrainte indépendante de  $y$ .
- $f$  : coefficient de frottement gaine/fil.
- $d$  et  $l$  : diamètre et longueur du fil dénudé.
- $\tau$  : contrainte de cisaillement entre la gaine et le fil.
- $S(y)$  : surface à dénuder avec  $S(y) = \frac{1}{2} \cdot \left[ y \cos(\theta) \cdot R \cdot \sin\left(\frac{y \cdot \arccos(\theta)}{R}\right) \right] - \frac{r^2 \theta}{2}$

Cette modélisation demeure un premier trait d'identification et de validation dans la perspective de ce travail de thèse.

## 5.6 Conclusion

Au travers de ce chapitre, nous avons dans un premier temps effectué un état de l'art sur les différentes stratégies de commande existantes. En effet, il ressort que les principales stratégies de commande sont très anciennes tout en étant toujours présentes dans l'état de l'art, des modifications et des améliorations viennent s'ajouter au fil du temps en fonction des problématiques et tout particulièrement en fonction du type de robot et de la tâche à accomplir.

Dans un second temps, nous avons fait un choix de la méthode que nous avons adoptée dans le cadre de notre travail, à savoir la méthode de commande position force externe applicable ici dans le cas de notre robot YuMi. Cette méthode a été bien détaillée dans le deuxième paragraphe de ce chapitre. De manière évidente, avant d'appliquer cette méthode sur le robot, il convient de la valider en simulation ; en sachant que nous avons réussi à établir un simulateur sous Matlab/Simulink capable de reproduire fidèlement le comportement du robot réel. Un exemple sans et avec bruit de cette simulation est présenté dans le troisième paragraphe.

Après la phase de validation arrive la phase de concrétisation où nous avons présenté cette fois les résultats expérimentaux appliqués sur le robot avec des exemples différents expliquant en détail l'étape de filtrage ; une étape primordiale dans notre cas de figure où les résultats obtenus sans filtrage n'étaient pas exploitables. La dernière partie, a présenté la validation du contrôle par l'application de dénudage de fils électriques. Nous avons tout d'abord expliqué la technique adoptée pour effectuer le dénudage et ensuite exposé les premiers résultats de dénudage avec un seul bras de YuMi. Par ailleurs, nous avons présenté à la fin de ce chapitre un modèle de force à identifier et à valider pour achever la robotisation de cette application.



# Conclusion générale et perspectives

## Bilan

Le projet de thèse présenté porte sur l'étude et la mise en œuvre de la commande en effort d'un robot à deux bras, dans le but de dénuder des fils électriques. Cette étude a été menée en considérant deux aspects de la recherche à savoir :

- La modélisation et l'identification dynamique du robot YuMi
- Le contrôle de la force externe exercée sur le robot YuMi afin de robotiser le dénudage de fils électriques.

Je voudrais en conséquence entamer cette conclusion en tirant un bilan des travaux réalisés dans cette optique et pour y parvenir, je propose de revenir sur le contenu de chaque chapitre pour en faire émerger les grandes lignes et les points essentiels.

Le premier chapitre a été voué à l'évocation des trois volets sur lesquels est axé notre sujet de thèse. À ce titre, nous avons abordé l'évolution des robots industriels collaboratifs au cours du temps et leur intérêt dans le cadre de l'Industrie 4.0. Nous avons ensuite exposé des statistiques valorisant l'intérêt d'un tel sujet de recherche. Par ailleurs, les robots à deux bras forment un volet tout aussi intéressant, nous avons donc esquissé les travaux existants traitant ce concept et plus particulièrement le propos de la coordination entre les deux bras et les différentes approches cherchant à décrire l'assemblage et la manipulation à deux bras. Au terme de ce chapitre, nous avons succinctement présenté l'application du dénudage, une application à robotiser à travers ce travail de thèse. Le concept de l'application a été clairement défini ainsi que les techniques et les outils déjà existants dans l'industrie pour réaliser de telles applications ont été cités.

Dans le second chapitre, une synthèse des principes de base de la modélisation des robots, et en particulier des robots manipulateurs, a été présentée. Nous avons ensuite présenté le robot YuMi IRB14000, le robot retenu pour notre projet, et pour lequel toutes les simulations et travaux expérimentaux de cette thèse sont accomplis. Afin de pouvoir identifier ou contrôler ce robot, il était en effet nécessaire de le modéliser au préalable. Nous avons opté pour la méthode Denavit-Hartenberg modifiée pour la modélisation géométrique directe et inverse du robot. Cette méthode a été décrite en profondeur, et nous avons également déterminé les repères de transformations de base des bras du robot. Cette modélisation a été validée par trois méthodes différentes (par modélisation théorique, par la Toolbox Robotic de Matlab ainsi que par le modèle fourni par le logiciel Symoro). Concernant la modélisation cinématique, nous l'avons calculée en déterminant la jacobienne et la pseudo-inverse de la jacobienne du robot ; des hypothèses ont été prises en compte dans cette dernière. Enfin, le modèle dynamique restant une discipline essentielle pour modéliser le comportement des robots, nous avons établi symboliquement le modèle dynamique en tenant compte du modèle de frottement, ainsi que

la modélisation linéaire au regard des paramètres de base (en principe les seuls paramètres identifiables du robot).

La problématique de l'identification des modèles dynamiques de robots a fait l'objet du troisième chapitre, soit une identification de l'ensemble des paramètres dynamiques non communiqués par le constructeur du robot. Pour ce faire, une étude bibliographique sur l'identification dans le contexte de la robotique a été menée. Ensuite, nous avons développé en détail notre méthode d'identification, en partant des mesures effectuées jusqu'aux dernières phases d'optimisation et de calcul des erreurs des paramètres identifiés. Les propriétés des différents signaux utilisés dans ce cadre, à savoir les trajectoires d'excitation et le traitement des signaux mesurés à exploiter, ont également été exposées. Notre technique d'optimisation est basée sur un critère quadratique contraint. En effet, des contraintes sont imposées sur les paramètres physiques afin que la matrice d'inertie et d'autres paramètres restent positifs quelle que soit la configuration du robot. Une procédure qui nous a permis de déterminer le vecteur de paramètres qui assure une bonne restitution des couples. Les résultats de l'identification et les valeurs des paramètres dynamiques, de l'offset et du frottement ont été obtenus. L'identification nous a permis de concevoir un simulateur du robot sur Matlab/Simulink. Un simulateur fondé sur la structure du contrôleur en cascade P/PI pour l'asservissement de la position et de la vitesse du robot réel ; ceci en identifiant ses principaux paramètres. Les mêmes comportements que le robot réel ont été réalisés et exploités pour simuler le fonctionnement et la conduite du robot sous une loi de commande externe. L'estimation du couple externe et la vérification des forces résultantes afin de les contrôler ont fait l'objet de la dernière partie de ce chapitre. Une comparaison des résultats de l'identification et du calcul du couple externe a été présentée en détail, ainsi que la procédure de validation du calcul des forces externes.

La mise en œuvre d'une loi de commande sur le robot industriel, nécessite une communication en temps réel stable et ininterrompue. Dans le chapitre quatre, nous avons fait le bilan de l'état de l'art sur les différents types de communication réalisables sur un robot, à savoir du remplacement du contrôleur initial au protocole de communication de téléopération. Tout d'abord, nous avons expliqué et détaillé la méthode utilisée pour réaliser la communication avec notre robot YuMi. Le protocole TCP/IP est à la base de cette communication. Pour cela, nous avons procédé à une interface client-serveur. Le robot n'étant pas capable de communiquer qu'avec le protocole TCP, une interface intermédiaire appelée Yumipy a été utilisée pour établir la communication. Nous avons établi des modifications sur l'interface Yumipy afin de l'adapter à nos vrais besoins ; elle ne permet pas de récupérer tous les couples articulaires. Les résultats de la communication sont présentés et la fréquence de transfert des données permet d'établir une loi de commande sur le robot en temps réel. Grâce à cette communication, nous avons été en mesure de valider notre loi de commande sur le simulateur et de l'implémenter de manière concrète sur le robot.

Dans le dernier chapitre, la méthode de commande choisie dans notre projet a été sélectionnée parmi différentes stratégies de commande de l'état de l'art. Il s'agit de la méthode de commande position/force externe adaptée au robot YuMi, dont le principe est amplement expliqué et détaillé. Avant d'appliquer cette méthode sur le robot, il convient *a priori* de la valider en simulation ; d'autant plus que nous avons réussi à établir un simulateur sous Matlab/Simulink capable de restituer avec une grande rigueur le fonctionnement du robot réel. Nous avons présenté un exemple sans et avec bruit de cette simulation. La phase de simulation est complétée par la phase de réalisation où nous avons présenté cette fois les résultats expérimentaux appliqués sur le robot avec différents exemples illustrant en détail l'étape de filtrage.

Cette étape était essentielle dans notre cas, car les résultats obtenus sans filtrage n'étaient pas exploitables. Pour finir, la validation du contrôle par l'application du dénudage des fils a été réalisée. La technique adoptée pour réaliser ce dénudage a d'abord été expliquée, puis nous avons présenté les premiers résultats du dénudage avec un seul bras de YuMi. En outre, à la fin de ce chapitre, nous avons proposé un modèle de force à identifier et à valider afin d'achever la robotisation de cette application.

Le bilan de cette étude est bien positif. Tout d'abord, nous avons apporté une solution à la fois pour l'identification des paramètres dynamiques du robot YuMi et pour son contrôle externe position/force. Nous avons ainsi atteint l'objectif que nous nous étions fixé : contrôler les forces externes appliquées au robot via une commande utilisant la communication TCP/IP. De plus, l'étude réalisée sur la commande nous a permis de procéder à la robotisation d'une application industrielle telle que le dénudage de fil électrique, une solution très prenante dans l'industrie électronique et informatique. Un des avantages de notre approche, est que nos travaux de recherche peuvent être directement exploités dans le milieu industriel et peuvent être la source d'une série de robotisation d'autres applications telles que la soudure de fils après dénudage qui nécessite également un contrôle d'effort très précis dans l'industrie. Je pense que mon travail de recherche a inclus les trois composantes importantes de la manipulation robotique : une composante théorique, une composante de validation expérimentale et une composante d'application industrielle.

## Perspectives

En termes de travaux prospectifs, nous projetons de poursuivre nos travaux à différents niveaux :

- Au niveau de la recherche, en gardant toujours la même structure de commande, nous pensons à développer et implémenter des algorithmes plus sophistiqués permettant d'utiliser les deux bras du robot en coordination pour réaliser la tâche de dénudage et de reproduire le comportement humain tout en l'améliorant, c'est-à-dire avec plus de précision et d'efficacité. Notons que ces algorithmes robustes peuvent *a priori* être implémentés sur la commande développée dans le présent travail et ceci sans aucun problème.
- Au niveau du développement, comme nous l'avons évoqué auparavant, nous cherchons à rendre l'application de dénudage robotisée, en utilisant le principe de la commande externe, suffisamment fiable avec des durées de cycle suffisamment réduites pour être utilisée en production. Au vu des résultats actuels et du modèle proposé, cette étape est encore simple à réaliser. Il est donc nécessaire d'identifier les paramètres du modèle proposé et de le valider par des tests expérimentaux. Confiants du succès de notre démarche et, étant donné le grand potentiel de la solution de contrôle des forces externes, nous souhaitons appliquer la solution proposée à d'autres tâches que le dénudage. Ceci nous amène à proposer également d'autres types de pinces en fonction de l'application souhaitée.



# Appendices

# Annexe A

## Modèle géométrique direct

La matrice de transformation du repère 7 au repère de base  $R_0$  est la suivante :

$${}_{R_0}T_7 = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_0$  est la matrice de transformation de base. En effet, pour définir le modèle géométrique du bras droit ou gauche, il suffit de changer cette matrice de transformation respectivement par  $R_{0r}$  ou  $R_{0l}$ ; les valeurs des composantes de ces matrices sont définies dans l'annexe C.

$$R_0 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

$$T_{11} = -S_7 \cdot (C_5 \cdot (S_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) - C_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) - S_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14})) - C_7 \cdot (S_6 \cdot (C_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}) - S_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})))) - C_6 \cdot (S_5 \cdot (S_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) - C_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + C_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}))))$$

$$T_{12} = S_7 \cdot (S_6 \cdot (C_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}) - S_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})))) - C_6 \cdot (S_5 \cdot (S_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) - C_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + C_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14})))) - C_7 \cdot (C_5 \cdot (S_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) - C_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) - S_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}))))$$

$$T_{13} = -C_6 \cdot (C_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}) - S_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})))) - S_6 \cdot (S_5 \cdot (S_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) - C_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + C_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}))))$$

$$T_{14} = R_{14} - R_{11} \cdot d_1 - d_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) - R_{14} \cdot r_1 + d_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11})) - r_3 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}) + d_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + d_5 \cdot (C_4 \cdot (C_3 \cdot (R_{14} \cdot S_2 - C_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}))) + S_3 \cdot (R_{11} \cdot S_1 - C_1 \cdot R_{12})) + S_4 \cdot (S_2 \cdot (R_{12} \cdot S_1 - C_1 \cdot R_{11}) + C_2 \cdot R_{14}))))$$



$$S_3.(R_{31}.S_1 - C_1.R_{32})) - S_6.(S_5.(S_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) - C_3.(R_{31}.S_1 - C_1.R_{32})) + C_5.(C_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32})) + S_4.(S_2.(R_{32}.S_1 - C_1.R_{31}) + C_2.R_{33})))$$

$$\begin{aligned} T_{34} = & R_{34} - R_{31}.d_1 - d_2.(R_{32}.S_1 - C_1.R_{31}) - R_{33}.r_1 + d_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) - \\ & r_3.(S_2.(R_{32}.S_1 - C_1.R_{31}) + C_2.R_{33}) + d_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) + S_3.(R_{31}.S_1 - \\ & C_1.R_{32})) + d_5.(C_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32})) + S_4.(S_2.(R_{32}.S_1 - \\ & C_1.R_{31}) + C_2.R_{33})) + r_5.(C_4.(S_2.(R_{32}.S_1 - C_1.R_{31}) + C_2.R_{33}) - S_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - \\ & C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32}))) + d_6.(S_5.(S_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) - C_3.(R_{31}.S_1 - \\ & C_1.R_{32})) + C_5.(C_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32})) + S_4.(S_2.(R_{32}.S_1 - \\ & C_1.R_{31}) + C_2.R_{33}))) - d_7.(S_6.(C_4.(S_2.(R_{32}.S_1 - C_1.R_{31}) + C_2.R_{33}) - S_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - \\ & C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32}))) - C_6.(S_5.(S_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) - C_3.(R_{31}.S_1 - \\ & C_1.R_{32})) + C_5.(C_4.(C_3.(R_{33}.S_2 - C_2.(R_{32}.S_1 - C_1.R_{31})) + S_3.(R_{31}.S_1 - C_1.R_{32})) + S_4.(S_2.(R_{32}.S_1 - \\ & C_1.R_{31}) + C_2.R_{33})))) \end{aligned}$$

# Annexe B

## Modèle géométrique inverse

Comme le bras du robot YuMi est un robot série à 7 DDL, on a forcé la position angulaire de l'articulation 3 à  $q_3 = 0^\circ$  afin de calculer la position angulaire de toutes les autres articulations à partir de la position et de l'orientation de l'effecteur terminal.

Dans cette partie, on restreint évidemment l'espace de travail du robot, mais il s'agit simplement de valider la simulation en prenant en compte les singularités et les configurations réelles du robot.

Considérons la matrice de passage homogène  $U_0$  :

$$U_0 = {}^0T_1 \cdot {}^1T_2 \dots \cdot {}^{n-1}T_n = \begin{bmatrix} S_x & n_x & a_x & x \\ S_y & n_y & a_y & y \\ S_z & n_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

Cette matrice  $U_0$  est une donnée du problème calculée à partir de la position désirée. Elle représente la position de l'organe terminal dans le repère de base du robot  $T_0$ . Les inconnues sont donc les six variables articulaires,  $q_1, q_2, q_4, q_5, q_6$  et  $q_7 (q_3 = 0)$ .

Les axes des trois dernières articulations se coupent en P qui peut donc être choisi comme origine des trois repères R4, R5 et R6. Connaissant la matrice de transformation  ${}^0T_{T_{cp}}$  et  $U_0$ , ces deux données permettent de déterminer directement la position du centre du poignet. De plus, la position de P dans  $T_0$  est une donnée du problème ( ${}^0T_7$ ).

Dans cette partie, on souhaite déterminer  $q_1$  en premier :

$${}^0T_7 = \begin{bmatrix} r_{11} & r_{21} & r_{31} & x \\ r_{21} & r_{22} & r_{32} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

où les éléments de la matrice  ${}^0T_7$  sont connus (des valeurs numériques).

Première démarche : on cherche à faire apparaître  $q_1$  dans la matrice :

$${}^1T_7 = {}^0T_1^{-1} \cdot {}^0T_7 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} & X \\ r_{21} & r_{22} & r_{32} & Y \\ r_{31} & r_{32} & r_{33} & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

avec  ${}^0T_7$  donné par la position et l'orientation de l'effecteur.

${}^1T_7$  peut également prendre la forme suivante :

$${}^1T_7 = \begin{bmatrix} {}^2r_{11} & {}^2r_{21} & {}^2r_{31} & c_1X - s_1Y \\ {}^2r_{21} & {}^2r_{22} & {}^2r_{32} & c_1Y + s_1X \\ {}^2r_{31} & {}^2r_{32} & {}^2r_{33} & Z - r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.4})$$

En se référant aux expressions des éléments de la matrice  ${}^1T_7$

$${}^1T_7 = \begin{bmatrix} {}^1r_{11} & {}^1r_{21} & {}^1r_{31} & r_3s_2 - r_5(c_2s_4 + c_4s_2) \\ {}^1r_{21} & {}^1r_{22} & {}^1r_{32} & 0 \\ {}^1r_{31} & {}^1r_{32} & {}^1r_{33} & r_5(c_2c_4 - s_2s_4) - c_2r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.5})$$

On remarque que l'élément [2,4] de la matrice est égal à zéro, en effet l'expression de l'élément n'est pas nulle, mais avec  $q_3 = 0$  le terme s'annule.

En identifiant les éléments [2,4] des 2 matrices impliquées dans l'égalité matricielle précédente, on a :

$$S_1X + C_1Y = 0 \quad (\text{B.6})$$

Cette équation peut être résolue par substitution de la tangente. On obtient donc :

$$q_1 = \arctan(Y/X) \quad (\text{B.7})$$

Pour la variable  $q_1$ , deux solutions sont donc possibles  $q_1$ .

Connaissant  $q_1$ , la détermination des angles du porteur  $q_2$  et  $q_4$  est possible. Nous effectuons les mêmes étapes que précédemment, en identifiant cette fois les éléments [1,4] et [3,4] de la matrice  ${}^1T_7$  :

$$\begin{aligned} C_1X - S_1Y &= r_3S_2 - r_5(C_2S_4 + C_4S_2) \\ Z - r_1 &= r_5(C_2C_4 - S_2S_4) - c_2r_3 \end{aligned}$$

On pose :

$$\begin{aligned} K_1 &= -r_5S_4 \\ K_2 &= r_3 - r_5C_4 \end{aligned}$$

On obtient donc :

$$\begin{aligned} C_1X - S_1Y &= r_3S_2 - r_5(C_2S_4 + C_4S_2) \\ Z - r_1 &= r_5(C_2C_4 - S_2S_4) - c_2r_3 \end{aligned}$$

Il en résulte alors :

$$\begin{cases} s_2 = \frac{C_1X + S_1Y + (Z - r_1) \frac{k_1}{k_2}}{k_2 + \frac{k_1^2}{k_2}} \\ c_2 = \frac{k_1S_2 - (Z - r_1)}{k_2} \end{cases} \quad (\text{B.8})$$

D'après l'équation (Eq. B.8), on peut déterminer les deux angles  $q_2$  et  $q_4$  :

$$q_2 = \text{atan2}(s_2, c_2) \quad (\text{B.9})$$

$$q_4 = \arccos \frac{A - r_5^2 - r_3^2}{-2r_3r_5} \quad (\text{B.10})$$

avec :  $A = X^2C_1^2 + Y^2S_1^2 + 2S_1C_1XY + Z^2 + r_1^2 - 2Zr_1$

Les variables articulaires  $q_1, q_2$  et  $q_4$  du robot sont calculées à partir de la position de l'organe terminal. On utilise à présent les orientations pour déterminer les angles du poignet  $q_5, q_6$  et  $q_7$ . En effet, nous écrivons :

$${}^4R_7 = {}^0R_4^{-1} \cdot {}^0R_7 = \begin{bmatrix} {}^3r_{11} & {}^3r_{21} & {}^3r_{31} \\ {}^3r_{21} & {}^3r_{22} & {}^3r_{32} \\ {}^3r_{31} & {}^3r_{32} & {}^3r_{33} \end{bmatrix} \quad (\text{B.11})$$

où les éléments de la matrice  $r_{ij}$  sont connus ( ${}^0R_7$ ) et  ${}^0R_4$  est calculée à partir des angles du robot connaissant  $q_1, q_2$  et  $q_4$ . Si nous regardons l'orientation de la matrice  ${}^4T_7$ , nous avons :

$${}^4R_7 = \begin{bmatrix} S_5S_7 + C_5C_6C_7 & C_7S_5 - C_5C_6S_7 & -C_5S_6 \\ -C_7S_6 & S_6S_7 & -C_6 \\ C_5S_7 - C_6C_7S_5 & C_5C_7 + C_6S_5S_7 & S_5S_6 \end{bmatrix} \quad (\text{B.12})$$

Après l'identification des éléments, on obtient :

$$\begin{aligned} q_5 &= \text{atan2}({}^3r_{33}, -{}^3r_{31}) \\ q_6 &= \text{atan2}(\sqrt{{}^3r_{21}^2 + {}^3r_{22}^2}, -{}^3r_{32}) \\ q_7 &= \text{atan2}({}^3r_{22}, -{}^3r_{21}) \end{aligned}$$

# Annexe C

## Matrices de transformation de base des deux bras

La matrice de transformation de base pour le bras gauche s'écrit comme suit :

$${}^{base}T_{bras} = R_0 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

$$\begin{aligned} R_{11} &= CA_{1l} \cdot CA_{2l} \\ R_{21} &= CA_{2l} \cdot SA_{1l} \\ R_{31} &= -SA_{2l} \\ R_{12} &= -(CA_{3l} \cdot SA_{1l}) + CA_{1l} \cdot SA_{2l} \cdot SA_{3l} \\ R_{22} &= CA_{1l} \cdot CA_{3l} + SA_{1l} \cdot SA_{2l} \cdot SA_{3l} \\ R_{32} &= CA_{2l} \cdot SA_{3l} \\ R_{13} &= CA_{1l} \cdot CA_{3l} \cdot SA_{2l} + SA_{1l} \cdot SA_{3l} \\ R_{23} &= CA_{3l} \cdot SA_{1l} \cdot SA_{2l} - CA_{1l} \cdot SA_{3l} \\ R_{33} &= CA_{2l} \cdot CA_{3l} \\ R_{14} &= x_{0l} \\ R_{24} &= y_{0l} \\ R_{34} &= z_{0l} \\ R_{34} &= z_{0l} \end{aligned}$$

D'où :

$$R_{0l} = \begin{bmatrix} 0.5534 & 0.1157 & 0.8248 & -0.0040 \\ -0.6384 & 0.6949 & 0.3309 & 0.0543 \\ -0.5349 & -0.7097 & 0.4584 & 0.3875 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_{0r} = \begin{bmatrix} 0.5534 & -0.1157 & 0.8248 & -0.0040 \\ 0.6384 & 0.6949 & -0.3309 & -0.0543 \\ -0.5349 & 0.7097 & 0.4584 & 0.3875 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Annexe D

## Jacobienne du robot YuMi

Afin de calculer la matrice Jacobienne du robot YuMi, nous avons utilisé trois différentes méthodes pour la valider ; la première est basée sur le modèle symbolique fourni par le logiciel SYMORO+, la seconde utilise la simulation sur la Toolbox robotique de Matlab et la troisième fait appel à la méthode classique utilisée en robotique détaillée ci-dessous :

Nous avons en cas général :

$${}^c\dot{X} = \begin{pmatrix} V_p \\ \omega \end{pmatrix} = J\dot{q} \quad \text{avec} \quad J = [j_1, j_2, \dots, j_n]_{n \times 1}$$

Avec :

En cas d'articulation rotoïde :  $J_i = \begin{pmatrix} {}^0Z_i \times {}^0l_i \\ {}^0Z_i \end{pmatrix}$

En cas d'articulation prismatique :  $J_i = \begin{pmatrix} {}^0Z_i \\ 0 \end{pmatrix}$

Pour le calcul des colonnes de la matrice jacobienne on utilise le modèle géométrique :

$${}^iT_j = \begin{bmatrix} & {}^0A_i & & {}^0P_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

$${}^0Z_i = {}^0A_i(:, 3)$$

$${}^0l_i = {}^0T_{Tcp}(1:3, 4) - {}^0T_i(1:3, 4)$$

(1 :3,4) : lignes 1 à 3 de la colonne 4 de la matrice A.

La matrice Jacobienne du robot est la suivante avec  $R_0$  qui représente la matrice de transformation de la base du bras à la base du robot calculée auparavant .

$$J(1, 1) = R_0.d_5.(C_{1+2}.C_4+S_{1+2}.C_3.S_4)-d_7.(R_0.S_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)+R_0.C_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))+R_0.d_6.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))+r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4 + S_{1+2}.S_3.S_5 - S_{1+2}.C_3.C_4.C_5) - R_0.S_6.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4)) + R_0.d_3.C_{1+2} -$$

$$R_0.r_7.S_7.(S_{1+2}.C_5.S_3 - C_{1+2}.S_4.S_5 + S_{1+2}.C_3.C_4.S(q_5)) - R_0.r_3.S_{1+2}.C_3 - R_0.d_4.S_{1+2}.S_3 + R_0.r_5.C_5.(C_{1+2}.S_4 - S_{1+2}.C_3.C(q_4)) + R_0.r_5.S_{1+2}.S_3.S(q_5)$$

$$J(1, 2) = R_0.d_5.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4) - d_7.(R_0.S_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) + R_0.C_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) - R_0.d_6.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) - R_0.S_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) + R_0.d_3.S_{1+2} + R_0.r_7.S_7.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + R_0.r_3.C_{1+2}.C_3 + R_0.d_4.C_{1+2}.S_3 + R_0.r_5.C_5.(S_{1+2}.S_4 + C_{1+2}.C_3.C(q_4)) - R_0.r_5.C_{1+2}.S_3.S(q_5)$$

$$J(1, 3) = 0$$

$$J(1, 4) = 0$$

$$J(1, 5) = 0$$

$$J(1, 6) = 1$$

$$J(2, 1) = R_0.d_5.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4) - d_7.(R_0.S_6.(C_{1+2}.C_5.S_4 + S_{1+2}.S_3.S_5 - S_{1+2}.C_3.C_4.C_5) + R_0.C_6.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4)) + R_0.d_6.(S_{1+2}.C_5.S_3 - C_{1+2}.S_4.S_5 + S_{1+2}.C_3.C_4.S(q_5)) + r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4 + S_{1+2}.S_3.S_5 - S_{1+2}.C_3.C_4.C_5) - R_0.S_6.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4)) + R_0.d_3.C_{1+2} - R_0.r_7.S_7.(S_{1+2}.C_5.S_3 - C_{1+2}.S_4.S_5 + S_{1+2}.C_3.C_4.S(q_5)) - R_0.r_3.S_{1+2}.C_3 - R_0.d_4.S_{1+2}.S_3 + R_0.r_5.C_5.(C_{1+2}.S_4 - S_{1+2}.C_3.C(q_4)) + R_0.r_5.S_{1+2}.S_3.S(q_5)$$

$$J(2, 2) = R_0.d_5.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4) - d_7.(R_0.S_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) + R_0.C_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) - R_0.d_6.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) - R_0.S_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) + R_0.d_3.S_{1+2} + R_0.r_7.S_7.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + R_0.r_3.C_{1+2}.C_3 + R_0.d_4.C_{1+2}.S_3 + R_0.r_5.C_5.(S_{1+2}.S_4 + C_{1+2}.C_3.C(q_4)) - R_0.r_5.C_{1+2}.S_3.S(q_5)$$

$$J(2, 3) = 0$$

$$J(2, 4) = 0$$

$$J(2, 5) = 0$$

$$J(2, 6) = 1$$

$$J(3, 1) = R_0.d_5.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4) - d_7.(R_0.S_6.(C_{1+2}.C_5.S_4 + S_{1+2}.S_3.S_5 - S_{1+2}.C_3.C_4.C_5) + R_0.C_6.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4)) + R_0.d_6.(S_{1+2}.C_5.S_3 - C_{1+2}.S_4.S_5 + S_{1+2}.C_3.C_4.S(q_5)) + r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4 + S_{1+2}.S_3.S_5 - S_{1+2}.C_3.C_4.C_5) - R_0.S_6.(C_{1+2}.C_4 + S_{1+2}.C_3.S_4)) - R_0.r_7.S_7.(S_{1+2}.C_5.S_3 - C_{1+2}.S_4.S_5 + S_{1+2}.C_3.C_4.S(q_5)) - R_0.d_4.S_{1+2}.S_3 + R_0.r_5.C_5.(C_{1+2}.S_4 - S_{1+2}.C_3.C(q_4)) + R_0.r_5.S_{1+2}.S_3.S(q_5)$$

$$J(3, 2) = R_0.d_5.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4) - d_7.(R_0.S_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) + R_0.C_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) - R_0.d_6.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4 - C_{1+2}.S_3.S_5 + C_{1+2}.C_3.C_4.C_5) - R_0.S_6.(S_{1+2}.C_4 - C_{1+2}.C_3.S_4)) + R_0.r_7.S_7.(C_{1+2}.C_5.S_3 + S_{1+2}.S_4.S_5 + C_{1+2}.C_3.C_4.S(q_5)) + R_0.d_4.C_{1+2}.S_3 + R_0.r_5.C_5.(S_{1+2}.S_4 + C_{1+2}.C_3.C(q_4)) - R_0.r_5.C_{1+2}.S_3.S(q_5)$$

$$J(3, 3) = 0$$

$$J(3, 4) = 0$$

$$J(3, 5) = 0$$

$$J(3, 6) = 1$$

$$J(4, 1) = R_0.d_5.(C_{1+2}.C_4+S_{1+2}.C_3.S_4)-d_7.(R_0.S_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)+R_0.C_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))+R_0.d_6.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))+r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)-R_0.S_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))-R_0.r_7.S_7.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))+R_0.r_5.C_5.(C_{1+2}.S_4-S_{1+2}.C_3.C(q_4))+R_0.r_5.S_{1+2}.S_3.S(q_5)$$

$$J(4, 2) = R_0.d_5.(S_{1+2}.C_4-C_{1+2}.C_3.S_4)-d_7.(R_0.S_6.(S_{1+2}.C_5.S_4-C_{1+2}.S_3.S_5+C_{1+2}.C_3.C_4.C_5)+R_0.C_6.(S_{1+2}.C_4-C_{1+2}.C_3.S_4))-R_0.d_6.(C_{1+2}.C_5.S_3+S_{1+2}.S_4.S_5+C_{1+2}.C_3.C_4.S(q_5))+r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4-C_{1+2}.S_3.S_5+C_{1+2}.C_3.C_4.C_5)-R_0.S_6.(S_{1+2}.C_4-C_{1+2}.C_3.S_4))+R_0.r_7.S_7.(C_{1+2}.C_5.S_3+S_{1+2}.S_4.S_5+C_{1+2}.C_3.C_4.S(q_5))+R_0.r_5.C_5.(S_{1+2}.S_4+C_{1+2}.C_3.C(q_4))-R_0.r_5.C_{1+2}.S_3.S(q_5)$$

$$J(4, 3) = 0$$

$$J(4, 4) = 0$$

$$J(4, 5) = 0$$

$$J(4, 6) = 1$$

$$J(5, 1) = R_0.d_6.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))-d_7.(R_0.S_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)+R_0.C_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))+r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)-R_0.S_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))-R_0.r_7.S_7.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))$$

$$J(5, 2) = r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4-C_{1+2}.S_3.S_5+C_{1+2}.C_3.C_4.C_5)-R_0.S_6.(S_{1+2}.C_4-C_{1+2}.C_3.S_4))-R_0.d_6.(C_{1+2}.C_5.S_3+S_{1+2}.S_4.S_5+C_{1+2}.C_3.C_4.S(q_5))-d_7.(R_0.S_6.(S_{1+2}.C_5.S_4-C_{1+2}.S_3.S_5+C_{1+2}.C_3.C_4.C_5)+R_0.C_6.(S_{1+2}.C_4-C_{1+2}.C_3.S_4))+R_0.r_7.S_7.(C_{1+2}.C_5.S_3+S_{1+2}.S_4.S_5+C_{1+2}.C_3.C_4.S(q_5))$$

$$J(5, 3) = 0$$

$$J(5, 4) = 0$$

$$J(5, 5) = 0$$

$$J(5, 6) = 1$$

$$J(6, 1) = r_7.C_7.(R_0.C_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)-R_0.S_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))-d_7.(R_0.S_6.(C_{1+2}.C_5.S_4+S_{1+2}.S_3.S_5-S_{1+2}.C_3.C_4.C_5)+R_0.C_6.(C_{1+2}.C_4+S_{1+2}.C_3.S_4))-R_0.r_7.S_7.(S_{1+2}.C_5.S_3-C_{1+2}.S_4.S_5+S_{1+2}.C_3.C_4.S(q_5))$$

$$J(6, 2) = r_7.C_7.(R_0.C_6.(S_{1+2}.C_5.S_4-C_{1+2}.S_3.S_5+C_{1+2}.C_3.C_4.C_5)-R_0.S_6.(S_{1+2}.C_4-C_{1+2}.C_3.S_4))$$

$$C_3 \cdot S_4)) - d_7 \cdot (R_0 \cdot S_6 \cdot (S_{1+2} \cdot C_5 \cdot S_4 - C_{1+2} \cdot S_3 \cdot S_5 + C_{1+2} \cdot C_3 \cdot C_4 \cdot C_5) + R_0 \cdot C_6 \cdot (S_{1+2} \cdot C_4 - C_{1+2} \cdot C_3 \cdot S_4)) + R_0 \cdot r_7 \cdot S_7 \cdot (C_{1+2} \cdot C_5 \cdot S_3 + S_{1+2} \cdot S_4 \cdot S_5 + C_{1+2} \cdot C_3 \cdot C_4 \cdot S(q_5))$$

$$J(6, 3) = 0$$

$$J(6, 4) = 0$$

$$J(6, 5) = 0$$

$$J(6, 6) = 1$$

$$J(7, 1) = 0$$

$$J(7, 2) = 0$$

$$J(7, 3) = 0$$

$$J(7, 4) = 0$$

$$J(7, 5) = 0$$

$$J(7, 6) = 1$$

# Annexe E

## Filtrage

Un filtre est notamment défini par sa fonction de transfert  $H(p)$  qui permet de connaître :

- La réponse temporelle :  $X(t) \rightarrow Y(t)$  avec  $Y(t) = L^{-1}[H(p) \cdot X(p)]$

avec  $X$  signal d'entrée variant dans le temps,  $Y$  signal de sortie et  $L^{-1}$  la transformée unilatérale de Laplace

- La réponse harmonique ( $p = j\omega$ )

— Module et arguments pour représenter la réponse harmonique du filtre

— Gabarit pour représenter les spécifications du filtre

Un filtre est dit linéaire s'il ne fait apparaître aucune composante spectrale dans le signal de sortie

### Filtre Butterworth

Le filtre de Butterworth est un type de filtre de traitement du signal destiné à avoir une réponse en fréquence plate (pas d'ondulation) dans la bande passante et une réponse nulle dans la bande d'arrêt (voir Figure E.1). Ces filtres sont les plus utilisés dans l'analyse du mouvement et dans les signaux audio. Ils sont rapides et simples à utiliser et dont la fonction de transfert est définie comme suit .

$$|H(j\omega)| = \frac{H_0}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} = \frac{1}{B_n(j\omega)}$$

Avec :

—  $n$  est l'ordre du filtre

—  $\omega_c$  est la fréquence de coupure

—  $H_0$  est le gain à fréquence nulle

—  $B_n(p)$  est un polynôme de Butterworth

Les propriétés principales polynôme de Butterworth sont :

Ordre du filtre	Dénominateur (le numérateur est à 1)
2	$p^2 + \sqrt{2}p + 1$
3	$(p^2 + p + 1)(p + 1)$
4	$(p^2 + 1.8477p + 1)(p^2 + 0.7653p + 1)$
5	$(p^2 + 1.6180p + 1)(p^2 + 0.6180p + 1)(p + 1)$
6	$(p^2 + 1.9318p + 1)(p^2 + \sqrt{2}p + 1)(p^2 + 0.5176p + 1)$

TABLE E.1 – Polynômes de Butterworth normalisés

–  $|B_n(j\omega)| = \sqrt{2} \quad \forall \quad n \quad \text{pour } \omega = \omega_c$

–  $|B_n(j\omega)|$  est une fonction strictement croissante

Le tableau E.1 présente Polynômes de Butterworth normalisés.

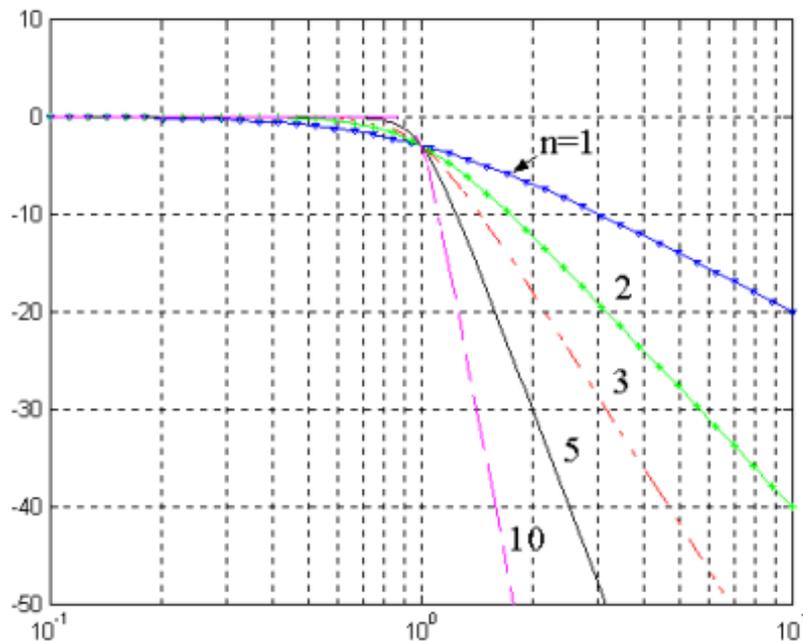


FIGURE E.1 – Réponse en amplitude des filtres de Butterworth (différents ordres)[55]

## Filtre Chebychev

Pour certaines applications, une réponse en amplitude très plate dans la bande passante n'est pas requise. On peut alors privilégier la caractéristique du filtre Tchebychev, qui offre une coupure plus raide dans la bande passante (voir Figure E.2) en comparaison avec la caractéristique de Butterworth.

On distingue deux types de filtres de Chebyshev et par conséquent deux types de fonction de transfert pour les filtres passe-bas :

— **Chebyshev de type 1** : qui présente des oscillations dans la bande passante :

$$|H(j\omega)| = \frac{H_0}{\sqrt{1 + \varepsilon^2 C_n^2(\omega/\omega_c)}}$$

— **Chebyshev de type 2** : qui présente des oscillations dans la bande d'arrêt :

$$|H(j\omega)| = H_0 \frac{\varepsilon \cdot C_n(\omega_c/\omega)}{\sqrt{1 + \varepsilon^2 C_n^2(\omega_c/\omega)}}$$

où  $C_n$  est le polynôme de Chebyshev d'ordre  $n$  défini par :  $C_{n-1}(\omega) - C_{n-2}(\omega)$ ,  $\varepsilon$  est le facteur d'ondulation et  $\omega_c$  est la fréquence de coupure. Nous avons utilisé dans notre projet le type 1 dont les polynômes normalisés sont présentés dans Table E.2.

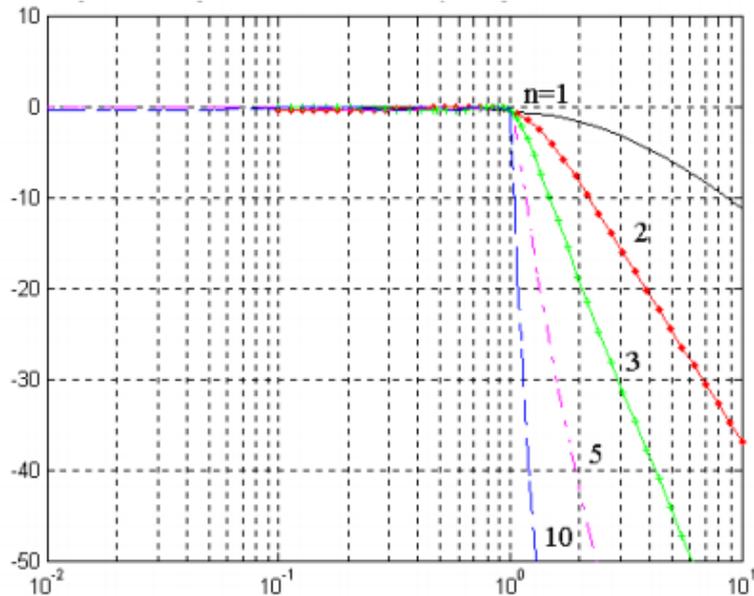


FIGURE E.2 – Réponse en amplitude des filtres Chebyshev pour 0.5 dB d'ondulation [55]

Ordre du filtre	Dénominateur (le numérateur est à 1)
2	$0.9070p^2 + 0.9956p + 1$
3	$(1.0058p^2 + 0.497p + 1)(2.023p + 1)$
4	$(1.0136p^2 + 0.2828p + 1)(3.5791p^2 + 2.4113p + 1)$
5	$(2.3293p^2 + 1.0911p + 1)(1.0118p^2 + 0.1610p + 1)(3.454p + 1)$

TABLE E.2 – Polynômes normalisés de filtre Chebyshev de type 1 pour  $\varepsilon = 1dB$



# Bibliographie

- [1] 2018. URL : <http://www.interreg-gr.eu/fr/presentation%3E>.
- [2] 2019. URL : <https://m.boutique.afnor.org/xml/1989134/false> (visité le 2021).
- [3] 2021. URL : <https://ifr.org/ifr-press-releases/news/record-2.7-million-robots-work-in-factories-around-the-globe> (visité le 05/05/2021).
- [4] 2021. URL : [http://https://www.phoenixcontact.com/assets/downloads\\_ed/global/web\\_dwl\\_promotion/52006038\\_Werkzeug\\_fr-int\\_LoRes.pdf](http://https://www.phoenixcontact.com/assets/downloads_ed/global/web_dwl_promotion/52006038_Werkzeug_fr-int_LoRes.pdf) (visité le 2021).
- [5] 2021. URL : <https://www.marketsandresearch.biz/report/113071/global-wire-stripping-machine-market-2020-by-manufacturers-regions-type-and-application-forecast-to-2025> (visité le 2021).
- [6] 2021. URL : <https://cableriedaumesnilblog.com/fils-electriques-et-cables-electriques-quelles-differences> (visité le 2021).
- [7] 2021. URL : <https://www.lintech.fr/produits-et-solutions/cable-fil/denudage-laser/> (visité le 2021).
- [8] 2021. URL : <https://new.abb.com/products/robotics/sv/robotstudio> (visité le 20/03/2021).
- [9] 2021. URL : <https://www.oracle.com/solaris/> (visité le 20/04/2021).
- [10] 2021. URL : <https://berkeleyautomation.github.io/yumipy/> (visité le 20/03/2021).
- [11] B.V. ADORNO, P. FRAISSE et S. DRUON. “Dual position control strategies using the cooperative dual task-space framework”. In : *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, p. 3955-3960.
- [12] K. AFSARI, M.E. SWARTS et T.R. GENTRY. “Integrated generative technique for interactive design of brickworks”. In : (2014).
- [13] P. AKELLA, M. PESHKIN, E. COLGATE, W. WANNASUPHOPRASIT, N. NAGESH, J. WELLS, S. HOLLAND, T. PEARSON et B. PEACOCK. “Cobots for the automobile assembly line”. In : *Proceedings 1999 IEEE International Conference on Robotics and Automation*. T. 1. IEEE. 1999, p. 728-733.
- [14] A. ALBU-SCHÄFFER, S. HADDADIN, C. OTT, A. STEMMER, T. WIMBÖCK et G. HIRZINGER. “The DLR lightweight robot : design and control concepts for robots in human environments”. In : *Industrial Robot : an international journal* (2007).
- [15] A. ALCOCER, A. ROBERTSSON, A. VALERA et R. JOHANSSON. “Force estimation and control in robot manipulators”. In : *IFAC Proceedings Volumes 36.17* (2003). 7th IFAC Symposium on Robot Control (SYROCO 2003), Wroclaw, Poland, 1-3 September, 2003, p. 55-60. ISSN : 1474-6670.

- [16] A.J. ALVARES, G.C. de CARVALHO, L.F. de AGUILAR PAULINYI et S.C.A. ALFARO. “Te-lerobotics : through-the-Internet teleoperation of the ABB IRB 2000 industrial robot”. In : *Telem manipulator and Telepresence Technologies VI*. T. 3840. International Society for Optics et Photonics. 1999, p. 259-269.
- [17] CH. ANA et C.G. HOLLERBACH. “Estimation of inertiel parameters of rigid body of manipulators”. In : *IEEE Conference on desicion and control*. IEEE, 1985.
- [18] O. BENDEL. “Co-robots from an Ethical Perspective”. In : *Business Information Systems and Technology 4.0*. Springer, 2018, p. 275-288.
- [19] C. BIDARD, C. LIBERSA, D. ARHUR, Y. MEASSON, J.P FRICONNEAU et J.D PALMER. “Dynamic identification of the hydraulic Maestro manipulator—Relevance for monitoring”. In : *Fusion Engineering and Design* 75-79 (2005). Proceedings of the 23rd Symposium of Fusion Technology, p. 559-564.
- [20] P. BIGRAS, M. LAMBERT et C. PERRON. “New formulation for an industrial robot force controller : Real-time implementation on a KUKA robot”. In : *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2007, p. 2794-2799.
- [21] P. BIGRAS, M. LAMBERT et C. PERRON. “Robust force controller for industrial robots : Optimal design and real-time implementation on a KUKA robot”. In : *IEEE transactions on control systems technology* 20.2 (2011), p. 473-479.
- [22] R. BLOSS. “Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications”. In : *Industrial Robot : An International Journal* (2016).
- [23] I-C. BOGDAN et G. ABBA. “Identification of the Servomechanism used for micro-displacement”. In : *International Conference on Intelligent Robots and Systems*. St. Louis, MO, USA : IEEE, oct. 2009, p. 1986-1991.
- [24] I.C. BOGDAN et G. ABBA. “Identification of mechanical parameters at low velocities for a micropositioning stage using a velocity hysteresis model”. In : *International Conference on Robotics and Automation*. Saint Paul, MN, USA., 2012, p. 430-435.
- [25] R. BOGUE. “Europe continues to lead the way in the collaborative robot business”. In : *Industrial Robot : An International Journal* 43(1) (2016), p. 6-11.
- [26] J. BOHUSLAVA, J. MARTIN et H. IGOR. “TCP/IP protocol utilisation in process of dynamic control of robotic cell according industry 4.0 concept”. In : *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE. 2017, p. 000217-000222.
- [27] T. BROGÅRDH. “Present and future robot control development—An industrial perspective”. In : *Annual Reviews in Control* 31.1 (2007), p. 69-79.
- [28] F. CACCAVALE, P. CHIACCHIO, A. MARINO et L. VILLANI. “Six-dof impedance control of dual-arm cooperative manipulators”. In : *IEEE/ASME Transactions On Mechatronics* 13.5 (2008), p. 576-586.
- [29] F. CACCAVALE, C. NATALE, B. SICILIANO et L. VILLANI. “Integration for the next generation : embedding force control into industrial robots”. In : *IEEE Robotics & Automation Magazine* 12.3 (2005), p. 53-64.
- [30] D.G CALDWELL, N.G TSAGARAKIS, S. KOUSIDOU, N. COSTA et I. SARAKOGLU. “" Soft" exoskeletons for upper and lower body rehabilitation—design, control and testing”. In : *International Journal of Humanoid Robotics* 4.03 (2007), p. 549-573.

- [31] C. CANUDAS DE WIT, H. OLSSON et K.J. ASTROM. “A new model for control of systems with friction”. In : *IEEE Transactions on Automatic Control* 40.3 (mar. 1995), p. 419-425.
- [32] F.J. CARRILLO, A. BAYSSE et A. HABBADI. “Output error identification algorithms for continuous-time systems operating in closed-loop”. In : *IFAC Proceedings Volumes* 42.10 (), p. 408-413.
- [33] P. CHIACCHIO, S. CHIAVERINI et Bruno SICILIANO. “Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system”. In : (1996).
- [34] B.W. CHOI, D.G. SHIN, J.H. PARK, S.Y. YI et S. GERALD. “Real-time control architecture using Xenomai for intelligent service robots in USN environments”. In : *Intelligent Service Robotics* 2.3 (2009), p. 139-151.
- [35] Y. CHU, Z. HUANG et J. HAHN. “Improving prediction capabilities of complex dynamic models via parameter selection and estimation”. In : *Chemical Engineering Science* 64.19 (2009), p. 4178-4185.
- [36] B. COHEN, S. CHITTA et M. LIKHACHEV. “Search-based planning for dual-arm manipulation with upright orientation constraints”. In : *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, p. 3784-3790.
- [37] J.E. COLGATE, J. EDWARD, M.A. PESHKIN et W. WANNASUPHOPRASIT. “Cobots : Robots for collaboration with human operators”. In : *International Mechanical Engineering Congress and Exhibition* 58 (1996), p. 433-439.
- [38] A. COLOMÉ, D. PARDO, G. ALENYA et C. TORRAS. “External force estimation during compliant robot manipulation”. In : *2013 IEEE international conference on robotics and automation*. IEEE. 2013, p. 3535-3540.
- [39] M. DAEMI et B. HEIMANN. “Identification and compensation of gear friction for modeling of robots”. In : *ROMANSY 11*. Springer, 1997, p. 89-96.
- [40] P.R. DAHL. “A Solid Friction Model”. In : *Technical rept , AEROSPACE CORP EL SEGUNDO CA* (1968).
- [41] M. DANESH, F. SHEIKHOLESLAM et M. KESHMIRI. “External force disturbance rejection in robotic arms : An adaptive approach”. In : *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 88.10 (2005), p. 2504-2513.
- [42] P. DAUCHEZ, X. DELEBARRE, Y. BOUFFARD et E. DEGOULANGE. “Task modeling and force control for a two-arm robot”. In : *Proceedings. 1991 IEEE international conference on robotics and automation*. IEEE. 1991, p. 1702-1707.
- [43] J. DE SCHUTTER, T. DE LAET, J. RUTGEERTS, W. DECRÉ, R. SMITS, E. AERTBELIËN, K. CLAES et H. BRUYNINCKX. “Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty”. In : *The International Journal of Robotics Research* 26.5 (2007), p. 433-455.
- [44] J. DE SCHUTTER et H. VAN BRUSSEL. “Compliant robot motion I. A formalism for specifying compliant motion tasks”. In : *The International Journal of Robotics Research* 7.4 (1988), p. 3-17.
- [45] E. DEGOULANGE. “Effort control of a two-armed manipulator robot applied to the control of the deformation of a closed kinematic chain”. Thèse de doct. University of Montpellier, Janvier 1993.

- [46] S.A.M. DEGHAN, M. DANESH et F. SHEIKHOESLAM. "Adaptive hybrid force/position control of robot manipulators using an adaptive force estimator in the presence of parametric uncertainty". In : *Advanced Robotics* 29.4 (2015), p. 209-223.
- [47] J. DENAVIT et R.S HARTENBERGH. "A Kinematic notation for lower-pair mechanism based on matrices". In : *Journal of Applied Mechanism* 22 (1955), p. 215-221.
- [48] A.M DJURIC, R.J. URBANIC et JL. RICKLI. "A framework for collaborative robot (Co-Bot) integration in advanced manufacturing systems". In : *SAE International Journal of Materials and Manufacturing* 9.2 (2016), p. 457-464.
- [49] S. EL ZAATARI, M. MAREI, W. LI et Z. USMAN. "Cobot programming for collaborative industrial tasks : An overview". In : *Robotics and Autonomous Systems* 116 (2019), p. 162-180.
- [50] K.S. EOM, I.H. SUH, W.K. CHUNG et S.R OH. "Disturbance observer based force control of robot manipulator without force sensor". In : *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*. T. 4. IEEE. 1998, p. 3012-3017.
- [51] S. FANG, X. HUANG, H. CHEN et N. XI. "Dual-arm robot assembly system for 3C product based on vision guidance". In : *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2016, p. 807-812.
- [52] Å. FAST-BERGLUND, F. PALMKVIST, P. NYQVIST, S. EKERED et M. ÅKERMAN. "Evaluating cobots for final assembly". In : *Procedia CIRP* 44 (2016), p. 175-180.
- [53] G. FERRETTI, G. MAGNANI, P. PUTZ et P. ROCCO. "The structured design of an industrial robot controller". In : *Control Engineering Practice* 4.2 (1996), p. 239-249.
- [54] W.E FORD. "What is an open architecture robot controller?" In : *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*. IEEE. 1994, p. 27-32.
- [55] M. FRÉNEA. *Filtrage analogique : généralités*. University Antenne de Bretagne. 2003.
- [56] J.G. GARCIA, A. ROBERTSSON, J. ORTEGA et R. JOHANSSON. "Generalized contact force estimator for a robot manipulator". In : *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE. 2006, p. 4019-4024.
- [57] H. GARNIER, M. MANSLER et A. RICHARD. "Continuous-time model identification from sampled data : Implementation issues and performance evaluation". In : 76 (2003), p. 1337-1357.
- [58] M. GAUTIER et S. BRIOT. "Global identification of drive gains parameters of robots using a known payload." In : *EEE International Conference on Robotics and Automation 2* (2012), p. 2812-2817.
- [59] M. GAUTIER, A. JANOT et Vandanjon. P.O. "A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics". In : 21 (2001), p. 428-444.
- [60] M. GAUTIER et W. KHALIL. "Direct calculation of minimum set of inertial parameters of serial robots". In : (1990), p. 368-373.
- [61] M. GAUTIER et P. POIGNET. "Nonlinear identification continues in closed loop physical parameters of mechatronic systems by inverse model and least squares error input," in : (2001), p. 28-44.
- [62] Ph. GAUTIER M. Poignet. "Extended Kalman filtering and weighted least squares dynamic identification of robot control". In : 9 (2001), p. 1361-1372.

- [63] F. GAZ C. Flacco et A. DE LUCA. “Identifying the dynamic model used by the KUKA LWR : A reverse engineering approach”. In : (2014), p. 1386-1392.
- [64] J. GE, Z. CHI et Q. LI. “Small Part Assembly with Dual Arm robot and Smart Camera”. In : *ISR/Robotik 2014 ; 41st International Symposium on Robotics*. 2014, p. 1-6.
- [65] J. DE-GEA et F. KIRCHNER. “Modelling and simulation of robot arm interaction forces using impedance control”. In : *IFAC Proceedings Volumes 41.2* (2008), p. 15589-15594.
- [66] M. GHARBI, J. CORTÉS et T. SIMEON. “A sampling-based path planner for dual-arm manipulation”. In : *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE. 2008, p. 383-388.
- [67] K. GOLDBERG. “Robots and the return to collaborative intelligence”. In : *Nature Machine Intelligence* 1.1 (2019), p. 2-4.
- [68] A. GOLDENBERG et L. CHAN. “An approach to real-time control of robots in task space. Application to control of PUMA 560 without VAL-II”. In : *IEEE Transactions on Industrial Electronics* 35.2 (1988), p. 231-238.
- [69] M. GROTHJAHN et B. HEIMANN. “Determination of dynamic parameters of robots by base sensor measurements”. In : *IFAC Proceedings Volumes 33.27* (2000), p. 279-284.
- [70] M. GSFVERT. “Comparisons of Two Dynamic Friction Models”. In : *Proceedings of the 1997 IEEE International Conference on Control Applications* (1997), p. 6.
- [71] M. GUGLIELMI, E. JONKER et JM. PIASCO. “Modelling and identification of a two degrees of freedom SCARA robot using extended Kalman filtering”. In : *International conference on advanced robotics* (1987), p. 137-148.
- [72] P.J. HACKSEL et S.E. SALCUDEAN. “Estimation des forces de l’environnement et des vitesses des corps rigides à l’aide d’observateurs”. In : *Actes de la conférence internationale IEEE 1994 sur la robotique et l’automatisation*, 931-936 vol.2.
- [73] A. HENTOUT, M. AOUACHE, A. MAOUDJ et I. AKLI. “Human-robot interaction in industrial collaborative robotics : a literature review of the decade 2008-2017”. In : *Advanced Robotics* 33.15-16 (2019), p. 764-799.
- [74] N. HEPENG, Z. CHENGRUI et H. TIANLIANG. “A dynamic parameter identification method of industrial robots considering joint elasticity”. In : *International Journal of Advanced Robotic Systems* 16.1 (2019), p. 1729881418825217.
- [75] A. HERNÁNDEZ et V. PETUYA. “Position analysis of planar mechanisms with R-pairs using a geometrical-iterative method”. In : *Mechanism and Machine Theory* 39.2 (2004), p. 133-152.
- [76] N. HOGAN. “Impedance control : An approach to manipulation”. In : *1984 American control conference*. IEEE. 1984, p. 304-313.
- [77] J. HOLLERBACH, W. KHALIL et M. GAUTIER. “Model identification”. In : *Springer handbook of robotics*. Springer, 2016, p. 113-138.
- [78] B. HU, H. YU, M. CHEN et L. HAN. “Dual arm robot astronaut design and coordinated force control”. In : *2017 Chinese Automation Congress (CAC)*. IEEE. 2017, p. 6912-6917.
- [79] J.O. HUCKABY et H.I. CHRISTENSEN. “A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics”. In : *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.
- [80] T. ISHIDA. “Force Control in Coordination of Two Arms.” In : *IJCAI*. Citeseer. 1977, p. 717-722.

- [81] A. JANOT. “Contribution to modeling and identification of haptic interfaces”. Thèse de doct. University of Nantes, Janvier 2007.
- [82] A. JANOT, M. GAUTIER et A. JUBIEN. “Comparison between the CLOE method and the DIDIM method for robot identification”. In : *IEEE Transactions on Control Systems Technology* 22 (2014), p. 1935-1941.
- [83] A. JANOT, P.C. YOUNG et M. GAUTIER. “Identification and control of electro-mechanical systems using state-dependent parameter estimation”. In : *International Journal of Control* 90.4 (2017), p. 643-660.
- [84] O. AL-JARRAH, Y. F ZHENG et K.Y. YI. “Trajectory planning for two manipulators to deform flexible materials using compliant motion”. In : *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. T. 2. IEEE. 1995, p. 1517-1522.
- [85] S. JIANG, M. JIANG, Y. CAO, D. HUA, H. WU, Y. DING et B. CHEN. “A typical dynamic parameter identification method of 6-degree-of-freedom industrial robot”. In : *Proc. Inst. Mech. Eng. I J. Syst. Control Eng.* 231 (2017), p. 740-752.
- [86] M. JIN, Z. ZHANG, F. NI et H. LIU. “Cartesian space synchronous impedance control of two 7-DOF robot arm manipulators”. In : *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, p. 4750-4756.
- [87] A. JUBIEN. “Dynamic identification of robots with articular flexibilities.” Thèse de doct. University of Nantes, 2014.
- [88] W JUN, W. JINSONG et Y. ZHENG. “An overview of dynamic parameter identification of robots”. In : *Robotics and Computer-Integrated Manufacturing* 26 (2010), p. 414-419.
- [89] J JUNG, J LEE et K HUH. “Robust contact force estimation for robot manipulators in three-dimensional space”. In : *Proceedings of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science* 220.9 (2006), p. 1317-1327.
- [90] S. KATSURA, Y. MATSUMOTO et K. OHNISHI. “Modeling of force sensing and validation of disturbance observer for force control”. In : *IEEE Transactions on industrial electronics* 54.1 (2007), p. 530-538.
- [91] W KHALIL et P LEMOINE. “S Y M O R O+ Symbolic Modeling of Robots USER’S GUIDE”. In : (2003).
- [92] W. KHALIL et E. DOMBRE. “Modeling, Identification and Control of Robots”. In : *3rd edition. Taylor and Francis Group* (2002).
- [93] W. KHALIL et E. DOMBRE. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.
- [94] W. KHALIL et J.F KLEINFINGER. “A new geometric notation for open and closed-loop robots”. In : *Proceedings. 1986 IEEE International Conference on Robotics and Automation* (2003), p. 1174-1179.
- [95] O. KHATIB. “Object manipulation in a multi-effector robot system”. In : *Proceedings of the 4th international symposium on Robotics Research*. 1988, p. 137-144.
- [96] H. KOBAYASHI, S. KATSURA et K. OHNISHI. “An analysis of parameter variations of disturbance observer for motion control”. In : *IEEE Transactions on Industrial Electronics* 54.6 (2007), p. 3413-3421.
- [97] O. KORN, G. BIEBER et C. FRON. “Perspectives on Social Robots : From the Historic Background to an Experts’ View on Future Developments”. In : *Proceedings of the 11th Pervasive Technologies Related to Assistive Environments Conference*. 2018, p. 186-193.

- [98] I.D. LANDAU, B.D.O. ANDERSON et F. DE BRUYNE. “Closed-loop output error identification algorithms for nonlinear plants”. In : *EEE Conference on Decision and Control* (1999).
- [99] J. LEE, P.H. CHANG et R.S. JAMISOLA. “Relative impedance control for dual-arm robots performing asymmetric bimanual tasks”. In : *IEEE transactions on industrial electronics* 61.7 (2013), p. 3786-3796.
- [100] T. LEFEBVRE, J. XIAO, H. BRUYNINCKX et G. DE GERSEM. “Active compliant motion : a survey”. In : *Advanced Robotics* 19.5 (2005), p. 479-499.
- [101] S. LESSARD, P. BIGRAS, L-G DURAND, G. SOULEZ, G. CLOUTIER et J.A. DE GUISE. “Robust position/force controller design on an industrial robot for medical application using LMI optimization”. In : *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*. T. 3. IEEE, 2004, p. 2913-2917.
- [102] M.A. MARTINEZ-PRADO, J. RODRIGUEZ-RESÉNDIZ, R.A. GÓMEZ-LOENZO, G. HERRERA-RUIZ et L.A. FRANCO-GASCA. “An FPGA-based open architecture industrial robot controller”. In : *IEEE Access* 6 (2018), p. 13407-13417.
- [103] J.A. MARVEL. “Collaborative robots : A gateway into factory automation”. In : *National Institute of Standards and Technology* (2014).
- [104] H. MAYEDA, K. YOSHIDA et K. OSUKA. “Base parameters of manipulator dynamic models”. In : *IEEE Transactions on Robotics and Automation*. T. 6. IEEE, 1985, p. 312-321.
- [105] B. MERCER. *Newnes industrial control wiring guide*. Routledge, 2001.
- [106] B. MITCHELL, J. KOO, I. IORDACHITA, P. KAZANZIDES, A. KAPOOR, J. HANDA, G. HAGER et R. TAYLOR. “Development and application of a new steady-hand manipulator for retinal surgery”. In : *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, p. 623-629.
- [107] C. MITSANTISUK, K. OHISHI, S. URUSHIHARA et S. KATSURA. “Kalman filter-based disturbance observer and its applications to sensorless force control”. In : *Advanced Robotics* 25.3-4 (2011), p. 335-353.
- [108] A. NIKOUBIN et R. HAGHIGHI. “Lyapunov-based nonlinear disturbance observer for serial n-link robot manipulators”. In : *Journal of Intelligent and Robotic Systems* 55.2 (2009), p. 135-153.
- [109] P. OBAL, A. BURGHARDT et Krzysztof KURC. “Monitoring the parameters of industrial robots”. In : *International Workshop on Modeling Social Media*. Springer, 2018, p. 230-238.
- [110] K. OHNISHI. “H Observed Based Force Control without Force Sensor”. In : *IEEE Proc. IECON'91* (1991), p. 1049-1054.
- [111] J. ORTEGA, J.G. GARCIA, L.N. NIETO et A.S. GARCIA. “Open software architecture for advanced control of robotic manipulators”. In : (2014).
- [112] M.R. PERVEZ, P.N. ROY, H. AHAMED, M.M. ISLAM, C.M. ODDO, G. CIUTI et P. DARIO. “Real time position control of industrial robot over ethernet based communication framework”. In : *AIP Conference Proceedings*. T. 2121. 1. AIP Publishing LLC, 2019, p. 150001.
- [113] M. PESHKIN et J.E. COLGATE. “Cobots”. In : *Industrial Robot : An International Journal* 26(5) (1999), p. 335-341.

- [114] L.D. PHONG, J. CHOI et S. KANG. “External force estimation using joint torque sensors for a robot manipulator”. In : *2012 IEEE International Conference on Robotics and Automation*. 2012, p. 4507-4512.
- [115] T.C PHUNG. “Study on Application of Real-time Control Method for Controlling Position of Robots in the Given Time”. In : *Science & Technology Development Journal-Engineering and Technology* 3.3 (2020), p. 461-471.
- [116] P POIGNET et M. GAUTIER. “Comparison of weighted least squares and extended Kalman filtering methods for dynamic identification of robots. IEEE international conference on robotics and automation”. In : *IEEE International Conference on Robotics and Automation ICRA* (2000), p. 3622-3627.
- [117] C PRESSE et M. GAUTIER. “New criteria of exciting trajectories for robot identification”. In : *Proceedings IEEE International Conference on Robotics and Automation*. IEEE. 1993, p. 907-912.
- [118] W. RACKL, R. LAMPARIELLO et G. HIRZINGER. “Robot excitation trajectories for dynamic parameter estimation using optimized B-splines”. In : *2012 IEEE International Conference on Robotics and Automation*. 2012, p. 2042-2047.
- [119] M.H RAIBERT et J.J CRAIG. “Hybrid position/force control of manipulators”. In : (1981).
- [120] ABB ROBOTICS. “Application manual PC SDK”. In : *ABB AB Robotic products* (2015), p. 204-217.
- [121] J.K. SALISBURY. “Active stiffness control of a manipulator in cartesian coordinates”. In : *1980 19th IEEE conference on decision and control including the symposium on adaptive processes*. IEEE. 1980, p. 95-100.
- [122] B. SEGRESTIN. “Le tournant fayolien : des révolutions industrielles à la naissance de l’entreprise moderne”. In : *Entreprises et histoire* 2 (2016), p. 5-12.
- [123] S. V. SHAH, S. K. SAHA et J. K. DUTT. “Denavit-Hartenberg Parameterization of Euler Angles”. In : *Journal of Computational and Nonlinear Dynamics* 7 (avr. 2012).
- [124] F. SHAHMIRI et J. FICCA. “A model for real-time control of industrial robots”. In : *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. T. 33. IAARC Publications. 2016, p. 1.
- [125] B. SICILIANO et O. KHATIB. *Springer handbook of robotics*. springer, 2016.
- [126] A.C SMITH, F. MOBASSER et K. HASHTRUDI-ZAAD. “Neural-network-based contact force observers for haptic applications”. In : *IEEE Transactions on Robotics* 22.6 (2006), p. 1163-1175.
- [127] M.K. STEÏN et J. KAIVO-OJA. “Collaborative Robots : Frontiers of Current Literature”. In : *Zeki Sistemler Teori ve Uygulamaları Dergisi* 3.2 (2020), p. 13-20.
- [128] A. STOLT, F.B CARLSON, M. G ARDAKANI, I. LUNDBERG, A. ROBERTSSON et R. JOHANSSON. “Sensorless friction-compensated passive lead-through programming for industrial robots”. In : *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, p. 3530-3537.
- [129] A. STOLT, M. LINDEROTH et A. ROBERTSSON. “Force controlled robotic assembly without a force sensor”. In : *IEEE International Conference on Robotics and Automation*. IEEE. 2012, p. 1538-1543.
- [130] K. STOUFFER, J. FALCO et K. SCARFONE. “Guide to industrial control systems (ICS) security”. In : *NIST special publication* 800.82 (2011), p. 16-16.

- [131] D. SUN et J.K MILLS. “Adaptive synchronized control for coordination of multirobot assembly tasks”. In : *IEEE Transactions on Robotics and Automation* 18.4 (2002), p. 498-510.
- [132] D. SURDILOVIC, Y. YAKUT, T.M. NGUYEN, X.B. PHAM, A. VICK et R. MARTIN. “Compliance control with dual-arm humanoid robots : Design, planning and programming”. In : *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2010, p. 275-281.
- [133] J. SWEVERS, C. GANSEMAN, D. B. TUKEL, J. DE SCHUTTER et H. VAN BRUSSEL. “Optimal robot excitation and identification”. In : *IEEE Trans. Robot. Autom.* 13.5 (1997), p. 730-740.
- [134] C.L SZABOLCS. “Comparison between two methods of robot control in the production of prismatic parts”. In : *Nonconventional Technologies Review* 16.2 (2012).
- [135] M. TAGHBALOUT, JF. ANTOINE et G. ABBA. “Dynamic Identification of YuMi ABB Collaborative Robot”. In : *Mechanical French Congress CFM2019, Brest, France* (2019).
- [136] M. TAGHBALOUT, JF. ANTOINE et G. ABBA. “Experimental Dynamic Identification of a YuMi Collaborative Robot”. In : *9th IFAC Conference on Manufacturing Modelling, Management and Control MIM19* 52.13 (2019), p. 1168-1173.
- [137] M. TAGHBALOUT, JA. MAMA TSANGA, JF. ANTOINE et G. ABBA. “Robotic stripping with external position/force control applied to the YuMi collaborative robot”. In : *Robotix-Academy Conference for Industrial Robotics RACIR’21, Germany* (2021).
- [138] J. TAKAMATSU, K. OGAWARA, H. KIMURA et K. IKEUCHI. “Recognizing assembly tasks through human demonstration”. In : *The International Journal of Robotics Research* 26.7 (2007), p. 641-659.
- [139] T. J. TARN, G. A SHOULTS et S.P YANG. “A dynamic model of an underwater vehicle with a robotic manipulator using Kane’s method”. In : *Auton Robot* 3 (1996), p. 269-283.
- [140] J.S TOQUICA, D. BENAVIDES et J.M MOTTA. “Web compliant open architecture for teleoperation of industrial robots”. In : *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2019, p. 1408-1414.
- [141] M. UCHIYAMA et P. DAUCHEZ. “A symmetric hybrid position/force control scheme for the coordination of two robots”. In : *Proceedings. 1988 IEEE international conference on robotics and automation*. IEEE. 1988, p. 350-356.
- [142] A VALERA, M VALLÉS et J TORNERO. “Real-Time Robot Control implementation with Matlab/Simulink”. In : *IFAC Proceedings Volumes* 34.9 (2001), p. 527-532.
- [143] G. VAN ROSSUM et F.L DRAKE. “Python language reference manual”. In : *Network Theory United Kingdom* (2003).
- [144] M. VUKOBRATOVIC et A. TUNESKI. “Contact control concepts in manipulation robotics/spl minus/an overview”. In : *IEEE Transactions on industrial electronics* 41.1 (1994), p. 12-24.
- [145] Y. WANG, M. TAGHBALOUT, JF. ANTOINE et G. ABBA. “External communication of the robot ABB YuMi via virtual machine with analysis of hybrid position-force control”. In : *Robotix-Academy Conference for Industrial Robotics RACIR’20, Luxembourg* (2020).
- [146] W. WANNASUPHOPRASIT, P. AKELLA, M. PESHKIN et J.E. COLGATE. “Cobots : A novel material handling technology”. In : *in proceedings of International Mechanical Engineering Congress and Exposition, Anaheim, ASME 98-WA/MH-2*. Citeseer. 1998.

- [147] C.Y. WENG, W.C. TAN et I.M. CHEN. “A survey of dual-arm robotic issues on assembly tasks”. In : *ROMANSY 22–Robot Design, Dynamics and Control*. Springer, 2019, p. 474-480.
- [148] W. WU, S. ZHU, X. WANG et H. LIU. “Closed-loop dynamic parameter identification of robot manipulators using modified Fourier series”. In : *Int. J. Adv. Robot. Syst.* 9 (2012), p. 29.
- [149] A. S AL-YAHMADI, J. ABDO et T. HSIA. “Modeling and control of two manipulators handling a flexible object”. In : *Journal of the Franklin Institute* 344.5 (2007), p. 349-361.
- [150] Y. YAMADA, S. NAGAMATSU et Y. SATO. “Development of multi-arm robots for automobile assembly”. In : *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. T. 3. IEEE. 1995, p. 2224-2229.
- [151] K. YOSHIDA et W. KHALIL. “Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters”. In : *The International Journal of Robotics Research* 19.5 (2000), p. 498-510.
- [152] Y.F ZHENG et M.Z CHEN. “Trajectory planning for two manipulators to deform flexible beams”. In : *Robotics and autonomous systems* 12.1-2 (1994), p. 55-67.
- [153] R ZOLLNER, T. ASFOUR et R. DILLMANN. “Programming by demonstration : Dual-arm manipulation tasks for humanoid robots”. In : *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. T. 1. IEEE. 2004, p. 479-484.
- [154] L. ZOLLO, E. LOPEZ, L. SPEDALIERE, N. GARCIA ARACIL et E. GUGLIELMELLI. “Identification of Dynamic Parameters for Robots with Elastic Joints”. In : *Advances in Mechanical Engineering* 7.2 (2015), p. 843186.

# Résumé

Cette thèse a pour objectif principal de développer une loi de commande pour contrôler un robot à deux bras afin de remplir les tâches de manipulation et de garantir les aspects de sécurité pour l'homme et le robot. Pour y parvenir, nous avons présenté un aperçu des études qui ont été menées dans ce cadre. Ensuite, nous avons procédé à la modélisation cinématique géométrique et dynamique du robot. Nous avons utilisé le logiciel SYMORO+ pour calculer son modèle dynamique. Au terme d'une présentation détaillée de la méthode d'identification des paramètres des robots manipulateurs, nous avons appliqué celle-ci de manière concrète sur notre propre robot. Ceci nous a permis d'obtenir un vecteur de paramètres capable de garantir une matrice d'inertie définie positive pour toute configuration d'articulation du robot, tout en assurant une bonne qualité de reconstruction du couple pour des vitesses articulaires à la fois constantes et variables. Grâce à cette identification dynamique, nous avons pu réaliser un simulateur précis du robot sur Matlab/Simulink. Les forces externes du robot ont été estimées à partir du module dynamique identifié et une validation expérimentale a été décrite. Pour faire appliquer des lois de commande sur un robot industriel, il a été nécessaire de réfléchir à une communication externe sur le robot. L'approche adoptée a été présentée ainsi que sa réalisation sur le robot. Cette dernière, nous a permis de programmer en python toutes nos problématiques à mettre en œuvre sur le robot. Après avoir validé toutes les étapes préalablement citées nous sommes partis vers l'implémentation de la commande sur le robot. La tâche principale que nous avons adoptée est le dénudage des câbles. A cet effet, nous avons choisi de réaliser une commande hybride force/position dans laquelle les forces externes sont contrôlées afin de garantir l'absence de dommages sur les câbles à dénuder. Le robot qui a servi à réaliser toutes les expériences est le robot IRB14000 YuMi de ABB. Toutes les expériences réalisées dans le cadre de ce projet ont été décrites et présentées. Ce travail de thèse a été réalisé dans le cadre du projet Robotix Academy financé par le fonds européen de développement régional INTERREG V-A Grande Région.

## Mots-clefs

Robotique, Modélisation, Identification dynamique, Commande hybride, Commande position-effort, Robot collaboratif.

# Abstract

The main objective of this thesis is to develop a control law to control a two arm robot in order to achieve the manipulation tasks and to ensure the safety aspects for the human and the robot. To accomplish this, we presented an overview of the studies that have been conducted in this context. Thereafter, we proceeded to the kinematic and dynamic robot modeling. In order to calculate its dynamic model, we used the SYMORO+ software. After giving a detailed presentation of the method for identifying the parameters of manipulator robots, we applied it in a concrete way on our own robot. This allowed us to obtain a vector of parameters able to ensure a positive defined inertia matrix for any robot joint configuration, as well as a good quality of torque reconstruction for both constant and variable joint speeds. Thanks to this dynamic identification, we were able to realize an accurate simulator of the robot on Matlab/Simulink. The external forces of the robot have been estimated from the identified dynamic modulus and an experimental validation has been described. To apply control laws on an industrial robot, it was necessary to think about an external communication on the robot. The adopted approach was presented as well as its realization on the robot. The latter allowed us to program in python all our problems to be implemented on the robot. After having validated all the steps previously mentioned we moved on to the implementation of the control on the robot. The main task we adopted is the stripping of the cables. For this purpose, we chose to implement a hybrid force/position control in which the external forces are controlled in order to guarantee the absence of damage on the cables to be stripped. The robot used to perform all the experiments is the IRB14000 YuMi robot from ABB. All the experiments performed in this project have been described and presented. This thesis work was carried out within the framework of the Robotix Academy project financed by the European regional development fund INTERREG V-A Grande Région.

## Keywords

Robotics, Modeling, Dynamic identification, Hybrid control, Position-effort control, Collaborative robot.