



**HAL**  
open science

# Sur la commande des robots manipulateurs industriels en co-manipulation robotique

Abdelkrim Bahloul

► **To cite this version:**

Abdelkrim Bahloul. Sur la commande des robots manipulateurs industriels en co-manipulation robotique. Automatique / Robotique. Université Paris Saclay (COmUE), 2018. Français. NNT : 2018SACLS511 . tel-01963246v2

**HAL Id: tel-01963246**

**<https://theses.hal.science/tel-01963246v2>**

Submitted on 21 Dec 2018

**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.

# Sur la commande des robots manipulateurs industriels en co-manipulation robotique

Thèse de doctorat de l'Université Paris-Saclay  
préparée à l'Université Paris-Sud

École doctorale n° 580 – Sciences et Technologies de  
l'Information et de la Communication (STIC)

Spécialité de doctorat: Robotique

Thèse présentée et soutenue à Gif-sur-Yvette, le 07/12/2018, par

**Abdelkrim BAHLOUL**

Composition du jury:

Catherine BONNET Directrice de recherche INRIA, CentraleSupélec	Présidente du jury
Yacine CHITOUR Professeur Université Paris-Sud	Directeur de thèse
Yassine HADDAB Professeur Université de Montpellier	Examineur
Maria MAKAROV Professeur assistante CentraleSupélec	Examinatrice
Salah LAGHROUCHE Maître de conférences, HDR Université de Technologie Belfort-Montbéliard	Rapporteur
Mario SIGALOTTI Directeur de recherche INRIA, University Pierre and Marie CURIE	Examineur
Sami TLIBA Maître de conférences Université Paris-Sud	Co-Encadrant
Ali ZEMOUCHE Maître de conférences, HDR Université de Lorraine	Rapporteur



# Remerciements

Je tiens à adresser en premier lieu mes plus chaleureux remerciements à mon directeur de thèse Yacine CHITOUR pour m'avoir confié ce travail de recherche, ainsi que pour son aide et ses précieux conseils au cours de ces années.

Je souhaiterais exprimer ma gratitude à mon co-encadrant Sami TLIBA pour la confiance qu'il m'a témoignée du début à la fin du travail, pour sa disponibilité à comprendre et à communiquer, sans oublier ses précieuses intuitions. Je suis ravi d'avoir travaillé en sa compagnie car, outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

Je voudrais remercier vivement Monsieur Ali ZEMOUCHE, Maître de conférences à l'Université de Lorraine, et Monsieur Salah LAGHROUCHE, Maître de conférences à l'Université de Technologie de Belfort-Montbéliard, pour avoir consacré du temps à la lecture de ce document en tant que rapporteurs.

J'associe à ces remerciements Madame Catherine BONNET, Directrice de recherche à l'INRIA d'avoir présidé le jury de la soutenance, ainsi que Monsieur Yassine HADDAB Professeur à l'Université de Montpellier, Monsieur Mario SIGALLOTI Directeur de recherche à l'INRIA, et Madame Maria MAKAROV Professeur assistante à CentraleSupélec, pour avoir accepté d'examiner mon travail.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	La co-manipulation robotique : une nécessité industrielle . . . . .	1
1.2	Contexte & objectif . . . . .	4
1.3	Contributions . . . . .	6
1.4	Structure du mémoire . . . . .	8
<b>2</b>	<b>Généralités sur le concept de co-manipulation</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.1.1	Définition . . . . .	11
2.1.2	Nécessité de la co-manipulation . . . . .	12
2.1.3	Classification des interactions homme-robot . . . . .	12
2.2	Différentes catégories de co-manipulation . . . . .	13
2.2.1	La co-manipulation parallèle . . . . .	13
2.2.2	La co-manipulation exosquelette . . . . .	14
2.2.3	La co-manipulation série . . . . .	14
2.3	État de l’art . . . . .	15
2.3.1	Programmation par démonstration . . . . .	15
2.3.2	Compensation du poids de l’outil/charge . . . . .	16
2.3.3	Augmentation d’effort . . . . .	17
2.3.4	Filtrage des tremblements . . . . .	19
2.3.5	Co-manipulation contrainte . . . . .	20
2.4	Sécurité de la co-manipulation homme-robot . . . . .	22
2.5	Plus d’exigences sur la co-manipulation dans l’industrie . . . . .	24
2.5.1	Assistance du robot pendant la co-manipulation . . . . .	24
2.5.2	Transparence de la co-manipulation . . . . .	25
2.5.3	Atteignabilité de la trajectoire générée . . . . .	25
2.5.4	Stabilité du robot en co-manipulation . . . . .	25
2.6	Conclusion . . . . .	26
<b>3</b>	<b>Modélisation et identification des paramètres dynamiques d’un robot</b>	<b>27</b>
3.1	Introduction . . . . .	27

3.2	Définitions . . . . .	28
3.3	Modélisation des robots séries à articulations rotoïdes . . . . .	30
3.3.1	Modèle géométrique direct . . . . .	30
3.3.2	Modèle cinématique direct . . . . .	35
3.3.3	Modèle dynamique . . . . .	39
3.4	Modélisation du robot Denso VP-6242G . . . . .	45
3.4.1	Description du robot Denso VP-6242G et de son module de contrôle . . . . .	45
3.4.2	Modélisation du robot . . . . .	47
3.5	Identification des paramètres dynamiques du robot . . . . .	50
3.5.1	Modèle dynamique pour l'identification . . . . .	50
3.5.2	Vecteur des paramètres inertiels minimaux . . . . .	51
3.5.3	Signaux d'excitation pour l'identification . . . . .	51
3.5.4	Traitement des signaux mesurés . . . . .	52
3.5.5	Estimation du vecteur des paramètres de base . . . . .	53
3.5.6	Identification des paramètres de la charge utile du robot . . . . .	56
3.5.7	Validation . . . . .	57
3.6	Identification des paramètres dynamiques du robot Denso VP-6242G . . . . .	57
3.6.1	Vecteur des paramètres de base . . . . .	57
3.6.2	Justification du modèle de frottement . . . . .	59
3.6.3	Signaux d'excitation . . . . .	61
3.6.4	Traitement des signaux mesurés . . . . .	63
3.6.5	Estimation des paramètres . . . . .	69
3.6.6	Reconstruction des couples . . . . .	72
3.6.7	Paramètres virtuels associés aux paramètres de base . . . . .	76
3.6.8	Reconstruction des couples de Coriolis, de gravité et de frottement . . . . .	77
3.7	Conclusion . . . . .	82
<b>4</b>	<b>Génération de trajectoires en temps-réel</b> . . . . .	<b>83</b>
4.1	Introduction . . . . .	83
4.2	La génération de trajectoire . . . . .	84
4.2.1	Génération de trajectoire dans l'espace articulaire . . . . .	84
4.2.2	Génération de trajectoire dans l'espace opérationnel . . . . .	85
4.2.3	Types des trajectoires générées . . . . .	85
4.2.4	Génération de trajectoire par interpolation . . . . .	87
4.3	Générateur de trajectoire pour la co-manipulation . . . . .	90
4.3.1	Formulation du problème de la co-manipulation . . . . .	90
4.3.2	Signal mesuré par le capteur de force . . . . .	92
4.3.3	Estimation de la force de l'opérateur . . . . .	93
4.3.4	Pénalisation de la force de l'opérateur . . . . .	95
4.3.5	Générateur de la vitesse désirée . . . . .	103
4.3.6	Pénalisation de la force de l'opérateur (suite) . . . . .	106
4.3.7	Position et accélération désirées . . . . .	108
4.4	Tests de validation . . . . .	108

---

4.5	Conclusion . . . . .	126
<b>5</b>	<b>Commande d'un robot en interaction</b>	<b>127</b>
5.1	Introduction . . . . .	127
5.2	Contrôle d'impédance par une commande basée sur l'inversion du modèle dynamique . . . . .	128
5.2.1	Loi de commande basée sur l'inversion du modèle dynamique .	129
5.2.2	Modulation des paramètres de l'opérateur d'impédance . . . .	130
5.3	Simulations & analyse des résultats . . . . .	131
5.4	Conclusion . . . . .	142
	<b>Conclusion et perspectives</b>	<b>145</b>
	<b>Bibliographie</b>	<b>146</b>

# Table des figures

2.1	Le shopping avec un robot [39]. . . . .	13
2.2	Un ouvrier en co-manipulation avec un robot pour une opération de montage dans l'industrie. . . . .	14
2.3	Co-manipulation exosquelette pour la rééducation du bras [42]. . . . .	14
2.4	Des robots utilisés en mode de co-manipulation série pour la suture chirurgicale [3]. . . . .	15
2.5	Robot en phase de démonstration. . . . .	15
2.6	Technique de programmation par démonstration basée sur la commande en impédance, et la compensation de la dynamique de l'outil [31]. . . . .	16
2.7	Système de levage permettant une compensation du poids de la charge [8]. . . . .	17
2.8	Le projet <i>Hardiman</i> inventé par <i>General Electric</i> . . . . .	18
2.9	Le projet <i>Extender</i> . . . . .	18
2.10	Robot de laparoscopie MC <sup>2</sup> E [25]. . . . .	19
2.11	Le robot développé dans [65]. . . . .	20
2.12	Placement d'objets dans un placard à l'aide de guides virtuels pour éviter les collisions avec les étagères [68]. . . . .	21
2.13	Le système de co-manipulation <i>Acrobot</i> utilisé en chirurgie orthopédique [40]. . . . .	22
2.14	Modèle d'un robot conçu à base d'un tronc visco-élastique [60]. . . . .	23
2.15	Principe de détection de collision à base de deux capteurs de force [62]. . . . .	24
3.1	Chaînes cinématiques ouverte (gauche), composée (milieu), et arborescente (droite). . . . .	29
3.2	Composantes d'un robot série. . . . .	29
3.3	Articulations rotoïde et prismatique. . . . .	29
3.4	Convention de Denavit-Hartenberg Modifiée. . . . .	31
3.5	Rotation et translation selon $x_i$ . . . . .	32
3.6	Rotation et translation selon $z_i$ . . . . .	33
3.7	Le robot <i>Denso VP-6242G</i> équipé d'un capteur de force et d'une outil. . . . .	45
3.8	Le capteur de force (a), l'outil installé (b) et le module de contrôle (c). . . . .	46
3.9	Les blocs <i>Denso Read et Write</i> . . . . .	46

3.10	Repères attachés au robot Denso VP-6242G selon la convention DHM.	47
3.11	Page principale du logiciel OpenSYMORO.	48
3.12	Options de menu disponibles dans OpenSYMORO.	49
3.13	Allure de la position angulaire (gauche) et de la vitesse articulaire (droite) correspondante pour l'axe 1, avec une vitesse $\dot{q}_1$ maximale de $50^\circ/sec$ .	59
3.14	Caractéristique couple de frottement en fonction de la vitesse articulaire pour les axes 1 (gauche) et 2 (droite).	60
3.15	Caractéristique couple de frottement en fonction de la vitesse articulaire pour les axes 3 (gauche) et 5 (droite).	60
3.16	Caractéristique temps de positionnement pour les six axes du robot Denso VP-6242G.	62
3.17	Signaux de données mesurées $q_1$ , $\tau_1$ , et calculées $\dot{q}_1$ , $\ddot{q}_1$ de l'axe 1 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	64
3.18	Signaux de données mesurées $q_2$ , $\tau_2$ , et calculées $\dot{q}_2$ , $\ddot{q}_2$ de l'axe 2 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	65
3.19	Signaux de données mesurées $q_3$ , $\tau_3$ , et calculées $\dot{q}_3$ , $\ddot{q}_3$ de l'axe 3 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	65
3.20	Signaux de données mesurées $q_4$ , $\tau_4$ , et calculées $\dot{q}_4$ , $\ddot{q}_4$ de l'axe 4 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	66
3.21	Signaux de données mesurées $q_5$ , $\tau_5$ , et calculées $\dot{q}_5$ , $\ddot{q}_5$ de l'axe 5 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	66
3.22	Signaux de données mesurées $q_6$ , $\tau_6$ , et calculées $\dot{q}_6$ , $\ddot{q}_6$ de l'axe 6 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).	67
3.23	Signaux de données mesurées et calculées des axes 1 et 2 résultant du traitement décrit ci-dessus.	68
3.24	Signaux de données mesurées et calculées des axes 3 et 4 résultant du traitement décrit ci-dessus.	68
3.25	Signaux de données mesurées et calculées des axes 5 et 6 résultant du traitement décrit ci-dessus.	69
3.26	Comparaison des couples mesurés après filtrage avec les couples calculés dans les deux cas (à gauche); Histogramme de l'erreur relative associée (à droite).	74
3.27	Comparaison des couples mesurés après filtrage avec les couples calculés dans les deux cas (à gauche); Histogramme de l'erreur relative associée (à droite).	75

3.28	Signaux d'excitation en position pour des morceaux de trajectoires articulaires à vitesse constante (accélération nulle). Les lignes rouges indiquent les limites articulaires. Les lignes vertes délimitent le mouvement articulaires à vitesse constante. . . . .	79
3.29	Vitesses et accélérations de référence correspondant aux positions de références de la figure 3.28. . . . .	80
3.30	Comparaison des couples mesurés après filtrage (bleu) avec ceux calculés (rouge) à gauche ; Histogramme de l'erreur relative associée à droite. . . . .	81
4.1	Exemple d'une trajectoire de référence à suivre par un manipulateur. . . . .	83
4.2	Génération de trajectoire en espace articulaire . . . . .	84
4.3	Génération de trajectoire en espace opérationnel . . . . .	85
4.4	Génération de trajectoire en espace opérationnel . . . . .	85
4.5	Exemple d'une trajectoire point-à-point pour une opération de soudage. . . . .	86
4.6	Exemple d'une trajectoire à points intermédiaires pour une opération d'oxycoupage. . . . .	86
4.7	Trajectoires en position, vitesse et accélération obtenues avec une spline cubique. . . . .	88
4.8	Trajectoires de position, vitesse, accélération et jerk obtenus avec une spline quintique. . . . .	90
4.9	Générateur de trajectoire en temps réel. . . . .	91
4.10	Estimation du biais pendant les premières secondes de la mise en marche du robot. . . . .	93
4.11	Récupération du signal de la force mesurée exprimé dans le repère attaché au capteur. . . . .	93
4.12	Capteur de force installé entre un bras et un outil. . . . .	94
4.13	Estimation de la force de l'opérateur. . . . .	95
4.14	Illustration du cas : $P = \emptyset$ . . . . .	97
4.15	Illustration du cas : $P \neq \emptyset$ & $K' = \emptyset$ . . . . .	98
4.16	Illustration du cas : $P \neq \emptyset$ , $K' \neq \emptyset$ & $\forall i, g_i(X_d) \leq 0$ . . . . .	98
4.17	Illustration du cas : $P \neq \emptyset$ , $K' \neq \emptyset$ & $\exists i, g_i(X_d) > 0$ . . . . .	99
4.18	Les différents cas possibles . . . . .	99
4.19	Procédure de pénalisation de $F_h$ . . . . .	100
4.20	Profil de $f(x)$ pour $\eta = 100$ , et $\gamma_1 = 0.75$ . . . . .	102
4.21	Générateur de la vitesse désirée. . . . .	103
4.22	L'automate gérant les états désirés du robot. . . . .	103
4.23	Génération de $\ \dot{X}_d\ $ en fonction de $\ F_h'\ $ . . . . .	105
4.24	Illustration du cas : $X_0 = [0 \ 0 \ 1.2]^T$ et $F_h = [1 \ 1 \ 0]^T$ . . . . .	110
4.25	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 1.2]^T$ et $F_h = [1 \ 1 \ 0]^T$ . . . . .	111
4.26	Illustration du cas : $X_0 = [0 \ 0 \ 1.2]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	112
4.27	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 1.2]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	113
4.28	Illustration du cas : $X_0 = [0 \ 0 \ 1.2]^T$ , $F_h = [0 \ 0 \ -2]^T$ . . . . .	114
4.29	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 1.2]^T$ , $F_h = [0 \ 0 \ -2]^T$ . . . . .	115
4.30	$\ddot{x}_d$ , $\ddot{y}_d$ & $\ddot{z}_d$ pour le cas : $X_0 = [0 \ 0 \ 1.2]^T$ , $F_h = [0 \ 0 \ -2]^T$ . . . . .	116

4.31	Illustration du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	116
4.32	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	117
4.33	$\ddot{x}_d$ , $\ddot{y}_d$ & $\ddot{z}_d$ pour le cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	118
4.34	Illustration du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [2 \ 0 \ 0]^T$ . . . . .	118
4.35	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [2 \ 0 \ 0]^T$ . . . . .	119
4.36	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [2 \ 0 \ 0]^T$ . . . . .	120
4.37	Illustration du cas : $X_0 = [0 \ 0 \ 0.92]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	120
4.38	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.92]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	121
4.39	Illustration du cas : $X_0 = [0 \ 0 \ 0.95]^T$ , $F_h = [0.9487 \ 0 \ -0.9]^T$ . . . . .	122
4.40	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.95]^T$ , $F_h = [0.9487 \ 0 \ -0.9]^T$ . . . . .	123
4.41	Illustration du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	124
4.42	Résultats de simulation du cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	125
4.43	$\ddot{x}_d$ , $\ddot{y}_d$ & $\ddot{z}_d$ pour le cas : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$ . . . . .	126
5.1	La stratégie de commande en impédance étudiée. . . . .	128
5.2	Stratégie du contrôle d'impédance. . . . .	129
5.3	Profil de $f_{K_p}(d)$ pour $\Lambda = 10$ et $\eta = 50$ . . . . .	131
5.4	Force mesurée avant (bleu) et après (rouge) filtrage. . . . .	133
5.5	Force $F_h$ estimée, avant et après pénalisation. . . . .	134
5.6	Position de l'organe terminal. . . . .	135
5.7	Norme de la position de l'organe terminal. . . . .	136
5.8	Composante normale de la force avant et après pénalisation. . . . .	137
5.9	Caractéristiques du vecteur vitesse linéaire du point d'interaction homme-robot. . . . .	138
5.10	. . . . .	140
5.11	Positions et vitesses articulaires, avec leurs limites. . . . .	141
5.12	Couples articulaires. . . . .	142

# Liste des tableaux

3.1	Paramètres de la convention DHM correspondant au robot Denso VP-6242G. . . . .	48
3.2	Paramètres des signaux d'excitation générés. . . . .	63
3.3	Spécifications des moteurs du robot Denso VP-6242G. . . . .	63
3.4	Valeurs numériques de $\theta_B$ avec et sans charge (exprimées en SI), et l'écart type des paramètres $\sigma$ . . . . .	70
3.5	Valeurs numériques de $\theta_p$ exprimées dans SI. . . . .	72
3.6	Vérification des contraintes dans (3.70). . . . .	72
3.7	Valeurs numériques des paramètres virtuels (exprimées en SI). . . . .	76
4.1	Valeurs des paramètres du profil de la vitesse pour chaque état [45]. . . . .	105





## Introduction

### **1.1 La co-manipulation robotique : une nécessité industrielle**

De nos jours, plusieurs secteurs de l'industrie reposent sur l'utilisation des robots manipulateurs (industriels), comme c'est le cas dans l'industrie automobile ou dans la construction navale. Plusieurs tâches ont été confiées à des robots, comme le transport de pièces, la manipulation, l'assemblage, la soudure ou encore la peinture. Ce type d'industrie nécessite fréquemment la manipulation de pièces massives et volumineuses. Par conséquent, les robots utilisés sont de grande dimension et possèdent des capacités importantes, que ce soit en terme de volume de travail, de vitesse de déplacement, de précision des mouvements ou bien en terme de charge utile. Ces capacités se traduisent par une puissance mécanique développée par un robot qui est telle, que ce dernier peut être meurtrier si, au cours de son mouvement, un robot manipulateur venait à entrer en collision avec un être humain se trouvant dans sa zone d'évolution. Ainsi, pour que les robots puissent accomplir leurs tâches dans les meilleures conditions, la maîtrise de leur environnement est indispensable. L'apparition inopportune d'un obstacle au mouvement pré-établi d'un robot augmente le risque de collision avec ce dernier, avec des conséquences pouvant être très néfastes. Jusqu'à présent, la perception limitée des robots vis-à-vis de leur environnement constitue un obstacle important à un plus large déploiement des robots dans les chaînes de production où co-existent des êtres humains. Cela contraint les entreprises à délimiter physiquement une zone de sécurité autour du robot, avec l'interdiction absolue pour un opérateur humain (OH) d'y pénétrer durant son fonctionnement. Il existe différentes normes et réglementations qui régissent l'espace de travail de ce type de machine-outil, au point que l'exigence précédente est la plupart du temps intégralement gérée par des automates complexes, dont le comportement fait état d'une sûreté de fonctionnement réputée être à toute épreuve.

Malgré tout, la robotisation des chaînes de production a permis d'augmenter la productivité grâce à leurs capacités mécaniques remarquables, mais aussi grâce à leur aptitude à fonctionner sans interruption pendant de très longues périodes. Une

ligne de production robotisée peut fonctionner sans arrêt, vingt-quatre heures sur vingt-quatre, car les robots ne sont pas sujets à l'épuisement, contrairement aux êtres humains. Ils ne sont arrêtés qu'occasionnellement pour des raisons de maintenance. De plus, les techniques de commande développées pour le contrôle des robots, ainsi que l'évolution de la technologie, ont permis de réaliser des produits à très haute valeur ajoutée avec un rendement tel, que l'investissement consacré à la robotisation d'une chaîne de production se retrouve vite amorti. Tout cela fait des robots manipulateurs, des outils économiquement incontournables pour plusieurs secteurs industriels.

Avec l'évolution de la société vers un mode de consommation toujours plus exigeant en terme d'innovation, conjugué à la compétition naturellement établie entre différents constructeurs d'un même type de produit, il est très important pour une manufacture moderne de pouvoir proposer de nouveaux produits dans un laps de temps toujours plus court. Les industriels ont alors tendance à miser sur un outil de production aussi polyvalent que possible afin de gagner en temps de développement et en compétitivité. Pour des raisons évidentes de coût et de rentabilité de l'outillage, le changement de produit dans une chaîne de production doit pouvoir être fait avec un investissement aussi réduit que possible dans l'outillage nécessaire et dans la reconfiguration des machines-outils déjà présentes. En milieu industriel, ce type d'attente s'intègre dans une logique dite de *flexibilité de l'outil de production*, érigée en paradigme. S'agissant des robots manipulateurs, cela passe par l'étude et la programmation de nouvelles trajectoires, adaptées à la nouvelle gamme de produit. Cette étape peut nécessiter un temps de développement coûteux au point que toute solution permettant de réduire le temps consacré à cela, s'inscrivant ainsi dans la logique de *flexibilité de l'outil de production*, ne peut que rencontrer un accueil enthousiaste parmi les ingénieurs chargés de concevoir et mettre en œuvre une nouvelle chaîne de production.

Parmi les solutions envisageables en accord avec cette logique, l'une d'elle consiste à permettre à un OH de pénétrer dans la zone de travail du robot, et lui donner la possibilité d'interagir directement et physiquement avec ce dernier. On parle alors d'Interaction Physique Homme-Robot (IPHR), traduction française de PHRI pour *Physical Human-Robot Interaction* en anglais, terme abondamment rencontré ces dernières années dans la littérature scientifique consacrée à la robotique [71, 79, 21, 28, 22]. L'idée sous-jacente est de faire collaborer un OH et un robot dans l'accomplissement d'une même opération, de manière à profiter des capacités du robot d'un côté et de la connaissance par l'OH de la tâche à accomplir de l'autre. Il existe différents modes d'interaction physique entre un robot et un être humain qui sont envisageables. Par exemple, un OH peut être chargé d'enseigner au robot les nouvelles trajectoires à suivre pour l'accomplissement d'une tâche (*p.ex.* la réalisation d'un cordon de soudure). Cette approche est connue sous le nom de « programmation par apprentissage » (*lead-through programming* en anglais). Elle rencontre de plus en plus d'intérêt dans les chaînes de fabrication, de montage ou d'assemblage, notamment dans le secteur automobile [31, 78]. Le principe est de permettre à l'OH de saisir le robot manipulateur à un endroit précis avec sa main (ou ses deux mains), cela peut être par exemple l'organe terminal équipé de l'outil nécessaire, en ayant préalablement établi une relation de type *maître-à-esclave* entre l'OH et le robot, par le biais

d'une instrumentation en capteurs et d'une loi de commande appropriées. L'OH dirigerait alors le déplacement de l'organe terminal avec un effort d'interaction minimal, le robot se soumettant au désir de déplacement imposé par l'OH sans opposer de résistance, et sans présenter de danger pour l'intégrité physique de l'OH. Ces exigences correspondent respectivement à des propriétés dites de « *transparence* » et de « *sécurité* » de fonctionnement. Il est également possible de convenir d'un mode d'interaction où, plutôt que d'être équipé d'un outil au niveau de son organe terminal, le robot manipulerait une pièce mécanique brut ou semi-finie, sur laquelle doivent être réalisées des opérations de fabrication (*p.ex.* du soudage, du découpage...), d'assemblage (*p.ex.* du boulonnage, du clipsage...) ou de finition (*p.ex.* du ponçage, de la peinture...) par l'OH. Cette pièce pouvant être lourde et volumineuse, son positionnement dans une configuration adaptée à l'opération réalisée peut s'avérer déterminant aussi bien pour le confort de l'OH que pour le coût de l'opération, en parti dû au temps passé par l'OH sur cette opération. Un être humain sera d'autant plus performant dans l'accomplissement d'une opération sur une pièce mécanique, que le positionnement dans l'espace de cette dernière soit adapté à sa physiologie, et que les efforts musculaires nécessaires de l'OH soient faibles. Cela réduit les risques de troubles musculosquelettique pouvant affecter l'OH, liés à la répétition au cours du temps d'une opération de fabrication dans une position d'inconfort musculaire, entraînant alors des douleurs chroniques pour l'OH et des « *arrêts-maladies* » coûteux pour la société. Pour ce mode de fonctionnement, le robot doit pouvoir maintenir la pièce dans une position et une orientation immobiles mais librement définie par l'OH, à n'importe quel moment. Là encore, l'IPHR doit pouvoir se faire sans nécessiter des efforts importants de la part de l'OH, le robot se chargeant de fournir les efforts nécessaires de maintien en position tout en étant robuste à des actions mécaniques exogènes pouvant perturber le positionnement choisi (collision fortuite, efforts appliqués à la pièce par l'outil...) durant l'opération. Cette exigence fait référence à la propriété « *d'assistance en effort* » par le robot. Le mode de fonctionnement ainsi défini correspond à la problématique de co-manipulation robotique pour la manutention de charges lourdes.

Ces deux modes d'interaction physique correspondent à une véritable collaboration entre l'OH et le robot dans l'accomplissement d'une opération. Une collaboration qui se veut intuitive avec un partage des tâches idéalement réparties entre le robot et l'OH dans la manipulation et le positionnement de la pièce. On parle alors de « *co-manipulation homme-robot* », qui a donné lieu à la thématique de recherche connue sous le nom de « *co-manipulation robotique* ». Un robot manipulateur industriel fonctionnant en mode de co-manipulation robotique sera désigné par le terme de « *cobot* », contraction des termes de co-manipulation et de robot.

## 1.2 Contexte & objectif

L'équipe du L2S dans laquelle s'est déroulée cette thèse a fait l'acquisition en juin 2013 d'un robot manipulateur DENSO VP-6242G dans le cadre du projet *Roboteo*, financé par le RTRA DIGITEO, et du projet *Comanip* financé par le LIDEX ICODE de la FCS Paris-Saclay. Il s'agit d'un robot manipulateur de type série à six axes qui sera décrit plus en détail dans le chapitre 3 de ce mémoire. Il est équipé d'un capteur de force inséré entre l'organe terminal du robot et une poignée sphérique permettant l'IPHR avec un OH dans le contexte d'une co-manipulation. Ce dispositif expérimental, configuré et opérationnel depuis juin 2017, doit permettre de reproduire une situation industrielle réaliste d'une IPHR pour la manutention d'une charge lourde.

Le travail exposé dans ce mémoire se veut être une contribution concrète à la problématique de commande posée dans le cas d'une manutention de charges lourdes par un robot manipulateur industriel, configuré en mode de co-manipulation. Il a pour point de départ les travaux de thèse de Sarra Jlassi [42], thèse effectuée au L2S et soutenue en novembre 2013. Le travail de thèse de Sarra Jlassi se basait sur une nouvelle formulation mathématique du problème de commande en co-manipulation robotique pour la manutention d'une charge lourde. Dans ce contexte, la formulation mathématique proposée a permis d'aborder la problématique considérée par le biais de critères mathématiques que la commande doit satisfaire afin d'apporter des éléments de réponse aux spécifications attendues d'assistance, de transparence et de sécurité. L'une des principales originalités de sa thèse portait sur la manière d'établir la relation « *maître-à-esclave* » recherchée en co-manipulation entre l'OH et le robot. Concernant cet aspect, le choix a été fait d'imposer une relation de colinéarité que le robot doit satisfaire en boucle fermée entre le vecteur de la vitesse linéaire du point d'interaction homme-robot, et le vecteur de la force d'interaction appliquée en ce point et mesurée par un capteur de force. Pour compléter cette formulation, le produit scalaire de ces deux vecteurs doit être positif afin d'exprimer le fait que le déplacement se produit dans le même sens que celui de la force de poussée exercée par l'OH. Il en résulte alors intuitivement qu'une telle commande du robot permet à l'OH de diriger le déplacement du point d'interaction par la seule connaissance de la force d'interaction mesurée par le capteur, établissant ainsi la relation « *maître-à-esclave* » nécessaire au type d'IPHR recherché pour la problématique considérée.

L'autre aspect original de cette thèse portait sur la manière d'établir cette relation « *maître-à-esclave* » tout en satisfaisant les exigences de transparence de l'interaction, d'assistance en effort et de sécurité durant l'interaction. Pour satisfaire ces différentes exigences, le schéma de commande développé dans la thèse de Sarra Jlassi est réalisé à l'aide d'un générateur de trajectoire (GT) en temps réel, combiné à un bouclage des position et vitesse articulaires sur les couples de commande articulaires. L'exigence de transparence suppose l'application d'un effort de faible amplitude par l'OH pour diriger l'organe terminal du robot. Cette spécification a été prise en compte dans le fonctionnement du GT par le biais de seuils d'effort qu'il faut franchir, dans un sens pour la mise en mouvement ou dans l'autre pour l'arrêt du mouvement. Ces

seuils d'effort sont ajustables pour régler le degré de transparence en fonctionnement durant l'interaction. Il en résulte un fonctionnement du GT à quatre états, suivant l'amplitude de la force mesurée par rapport aux seuils de mise en mouvement ou d'arrêt : *l'état d'immobilité*, *l'état de mise en mouvement*, *l'état du mouvement établi* et *l'état d'arrêt du mouvement*. Un automate à état discret est chargé de gérer le déclenchement ou l'arrêt du mouvement, en vérifiant à chaque instant les conditions de transition entre les différents états du GT. Ce dernier peut être vu comme un système dynamique entrées-sorties à événements discrets, où la force mesurée et le temps chronométré sont les entrées, les trajectoires désirées en terme de position, de vitesse et d'accélération, exprimées dans le repère de référence, constituent les sorties. Les trajectoires générées par le GT sont construites autour de la vitesse linéaire désirée pour le point d'interaction. La direction du vecteur vitesse désiré est donnée par la mesure du vecteur force délivrée par le capteur de force situé au niveau du point d'interaction. Son amplitude est quant à elle contrôlée par l'automate suivant l'état qui est actif. Il est à noter que *l'état du mouvement établi* se fait à une vitesse de norme constante, et de valeur limitée, conformément à la norme en vigueur ISO 10218-1 :2011 [14]. D'une part, cela concourt à limiter l'énergie cinétique emmagasinée par le robot durant le mouvement. D'autre part, cela permet de protéger l'OH de graves blessures en cas de collision. Par ailleurs, l'OH conserve l'autorité sur la position du point d'interaction. La trajectoire désirée étant générée en temps réel, il ne reste alors qu'à commander le mouvement articulaire du robot pour suivre cette trajectoire et assurer l'assistance en effort qui est requise. Toute loi de commande par bouclage entrées-sorties pouvant réaliser un suivi de trajectoire en boucle fermée peut alors être utilisée. Certaines de ces lois nécessitent la mesure des positions, vitesses et accélérations articulaires, d'autres ne requièrent que la mesure des positions et vitesses articulaires. Des travaux récents ont montré qu'il est possible de n'envisager que la mesure des positions articulaires pour permettre le suivi de trajectoire par un robot manipulateur industriel [70].

Toutes ces idées ont été illustrées dans la thèse de Sarra Jlassi par des simulations sur un modèle de robot virtuel, à deux axes, et évoluant dans le plan, le robot DENSO VP-6242G étant loin d'être opérationnel à son époque pour pouvoir effectuer quel qu'essai expérimental que ce soit. Dans ce contexte, beaucoup de limitations physiques qu'un véritable robot manipulateur peut avoir, ont été ignorées. C'est le cas par exemple de la saturation des couples mécaniques disponibles sur chaque axe, ou bien celui des limites physiques de l'espace articulaire (butées mécaniques). C'est également le cas des frottements dans les articulations ou bien les flexibilités articulaires, voire structurales qui peuvent avoir une contribution non-négligeable dans la dynamique d'un robot manipulateur industriel, trop souvent considéré à tort comme étant parfaitement rigide.

Dans le cadre de cette thèse, nous nous sommes attachés à la mise en œuvre pratique de cette stratégie de commande sur le robot manipulateur DENSO VP-6242G présent dans l'équipe, en tenant compte autant que possible des conditions de fonctionnement réalistes et avec le soucis constant de préserver l'intégrité du robot et la sécurité de l'OH durant les essais.

## 1.3 Contributions

Afin de répondre à cet objectif, nous avons d'abord focalisé notre attention sur l'obtention d'un modèle dynamique du robot manipulateur série à six axes DENSO VP-6242G, évoluant dans l'espace tridimensionnel. Chaque axe est considéré comme réalisant une liaison pivot (encore appelée *liaison rotoïde*) entre deux bras du robot, une liaison mécaniquement imparfaite compte tenu de la présence non négligeable de frottement dans toutes les liaisons. Chaque bras est supposé être parfaitement rigide. Pour cela, nous avons commencé par la construction d'un modèle analytique décrivant la position et l'orientation de l'organe terminal à partir des coordonnées articulaires. Ce modèle, dit Modèle Géométrique Direct (MGD), ne dépend que de paramètres géométriques qui sont connus. Il a permis d'en déduire le Modèle Dynamique Direct (MDD) suivant la structure mathématique usuelle des équations d'Euler-Lagrange. Le MDD dépend des paramètres inertiels de chaque bras du robot, c'est-à-dire les masses, les coordonnées de chaque centre de gravité ainsi que les moments et les produits d'inertie. Un modèle de frottement paramétrique a été intégré dans le MDD pour tenir compte des frottements dans les liaisons. La documentation technique disponible pour ce robot ne renseigne principalement que sur certaines caractéristiques géométriques comme les distances entres-axes ou les limites articulaires. En dehors de la masse totale du robot, aucun paramètre inertiel n'est donné. Par ailleurs, à notre connaissance, il n'existe aucun modèle géométrique numérique issu d'un logiciel de CAO<sup>1</sup> du robot DENSO VP-6242G disponible publiquement, ce qui permettrait de récupérer une approximation des informations inertielles manquantes. Par conséquent, nous nous sommes naturellement orientés vers l'identification des paramètres du robot à partir de données expérimentales. Cela nous a amené à une étude approfondie de cette thématique en commençant d'abord par considérer le Modèle Dynamique dit Inverse (MDI), un modèle non-linéaire en les coordonnées articulaires (et leurs deux premières dérivées temporelles) servant d'entrées mais linéaire en un vecteur de paramètres, dits *paramètres de base*. Les couples articulaires sont alors considérés comme des signaux de sortie. L'avantage du MDI, c'est qu'il rend possible la réécriture des équations d'Euler-Lagrange sous la forme d'un régresseur linéaire en le vecteur des paramètres de base. L'étude s'est poursuivie par la définition de signaux d'excitation appropriés, les essais expérimentaux et l'acquisition des données nécessaires pour la méthode standard d'identification conduisant à l'obtention d'un modèle de prédiction. Cette méthode s'appuie sur la formulation du problème d'estimation des paramètres de base comme la minimisation d'un critère mesurant l'écart des sorties prédites par le modèle avec les sorties mesurées. Cette mesure de l'écart se fait habituellement par le biais d'une norme euclidienne (critère des moindres carrés et ses différentes variantes). Cette approche, bien qu'efficace pour l'obtention d'un modèle de prédiction des sorties du MDI, possède l'inconvénient de fournir un modèle inapproprié pour les besoins de la commande, c'est-à-dire pour les méthodes de conception de lois de commande construites avec les différents termes du MDD. On parle de lois de commande à base de modèles (ou *model-based control* en anglais). Par inapproprié, il est question ici

---

1. CAO : Conception Assistée par Ordinateur

d'un modèle ne satisfaisant pas certaines propriétés mathématiques fondamentales du MDD, comme le fait que la matrice d'inertie doit être définie positive quelle que soit la configuration articulaire du robot. C'est dans ce cadre que nous avons envisagé d'intégrer, dans la procédure d'identification, des contraintes sur les paramètres dits *physiques* par lesquels, les paramètres de base peuvent être exprimés. Cela a donné lieu à la publication de l'article [19] dans lequel nous détaillons notre approche et sur laquelle nous reviendrons dans le chapitre 3. Cette approche a également servi à établir une procédure d'identification des paramètres inertiels d'une charge arrimée à l'organe terminal du robot, à partir d'un modèle identifié du robot manipulateur sans cette charge. Ce travail a été publié dans [20]. L'approche d'identification sous contraintes s'est révélée efficace en terme d'obtention d'un modèle de prédiction des sorties du MDI, et de ses différents termes. Nous avons utilisé deux jeux de données pour la validation des paramètres estimés. Le premier possède des caractéristiques spectrales, similaires au jeu de données utilisé pour l'identification. Le second a été produit pour permettre d'avoir des évolutions du robot à des vitesses articulaires constantes sur de longs intervalles de temps, afin que les accélérations articulaires soient (quasi-)nulles sur ces intervalles et que seuls certains termes du MDD, mathématiquement décrits dans le formalisme d'Euler-Lagrange, contribuent à l'équilibre dynamique. Nous avons considéré que le modèle obtenu était adapté aux besoins de la commande basée-modèle envisagée. Tout cela sera présenté et discuté dans le chapitre 3.

Concernant la partie commande, l'objectif de prise en compte des conditions de fonctionnement réalistes nous a d'abord amené à étendre les capacités du GT proposé dans [42], au cas d'un robot manipulateur industriel évoluant dans l'espace tridimensionnel. Par la suite, nous avons cherché à consolider le fonctionnement de l'automate à états discrets qui gère le déclenchement ou l'arrêt du mouvement durant l'IPHR, en rendant son fonctionnement plus sûr et plus robuste. Il s'agissait plus spécifiquement de mieux garantir les conditions de transition entre les quatre états du GT afin d'éviter des dynamiques de transition inadaptées à l'IPHR, par exemple en cas de variation trop rapide de l'intensité de la force traitée par l'automate autour des seuils de mise en mouvement et d'arrêt. Ensuite, nous avons cherché à tenir compte des limites de l'espace articulaire en intégrant d'une part des butées virtuelles sur les mouvements articulaires, et d'autre part des butées virtuelles dans le GT limitant le mouvement du point d'interaction dans un espace de travail restreint, délimité par une surface fermée qui est définie au préalable. L'objectif est de limiter la co-manipulation dans un volume que le point d'interaction homme-robot peut atteindre. Pour cela, lorsque le point d'interaction se situe sur la limite, le vecteur force mesurée est projeté sur le plan tangent de manière à ce que le GT génère une trajectoire sur la surface limitante, tant que l'opérateur exerce un effort dirigé vers l'extérieur de l'espace de travail restreint. Pour finir, nous proposons également d'adapter les paramètres de la loi de commande, de sorte qu'en s'approchant de la surface limitante, l'OH ressente une rigidité accrue de l'interaction pour lui faire ressentir physiquement qu'il a atteint cette limite de l'espace défini pour la co-manipulation.



## 1.4 Structure du mémoire

Dans le **chapitre 2**, nous revenons sur la description de l'IPHR en focalisant la présentation sur la co-manipulation homme-robot, telle qu'elle est apparue à ses débuts et les différents stades de l'évolution de cette thématique. Nous proposons une revue bibliographique des différents travaux ayant trait à la co-manipulation, en nous intéressant particulièrement aux différentes problématiques proposées dans la littérature. Nous finissons par l'introduction des différentes exigences pendant une tâche de co-manipulation homme-robot pour la manutention de charges lourdes. Le **chapitre 3** est consacré à la modélisation et à l'identification des paramètres permettant d'aboutir au modèle dynamique du robot **DENSO VP-6242G**. Nous commençons par expliquer la démarche d'obtention des modèles géométrique, cinématique et dynamique conformément à la version modifiée de la convention de Denavit-Hartenberg, telle qu'elle est proposée dans [48]. Nous poursuivons par la présentation de la méthode permettant d'aboutir à la formulation standard du problème d'identification des paramètres du MDI. Nous décrivons alors le protocole expérimental nécessaire à l'acquisition de données que l'on exploitera pour cette identification. Ensuite, nous présenterons la formulation du problème d'identification des paramètres du MDI qu'il s'agit de résoudre comme un problème d'optimisation d'un critère non-linéaire en les variables à estimer, sous différents types de contraintes. Nous illustrerons cette approche en exploitant les données issues des essais effectués sur le robot **DENSO VP-6242G**. Nous terminerons cette partie par l'examen de la qualité de reconstruction des couples mesurés et des termes dits de *Coriolis*, *gravité*, *frottement*, tout en se basant sur les paramètres estimés et le modèle dynamique du robot.

Le **chapitre 4** est consacré à la présentation, à l'étude et l'amélioration du générateur de trajectoire introduit dans [42]. D'abord, nous passerons en revue les différentes façons d'aborder la problématique de génération de trajectoire, dans le cas général. Nous reviendrons ensuite sur la spécificité d'un tel générateur dans le cas d'une problématique de co-manipulation homme-robot destinée à la manutention de charges lourdes. Nous exposerons les raisons pour lesquelles il est nécessaire d'adapter le GT à la force appliquée par l'OH et non pas simplement à la force mesurée par le capteur. En effet, cette dernière correspond à la résultante de l'action du robot et de l'OH sur la poignée du robot, c'est-à-dire à la force d'interaction. Nous poursuivrons alors la présentation en décrivant notre approche pour l'estimation de la force appliquée par l'OH à partir des mesures du capteur de force et des grandeurs cinématiques articulaires. La partie suivante sera consacrée à la prise en compte des limites de l'espace de travail réservé à la co-manipulation pour l'insertion d'une butée virtuelle dans l'espace de travail. Nous expliquerons comment nous tenons compte de la force appliquée par l'OH pour générer une trajectoire compatible avec les limites du volume défini pour l'interaction homme-robot. Nous présenterons ensuite la génération de la vitesse désirée pour l'organe terminal, et l'automate chargé de gérer les différents états de l'interaction. Nous y exposerons les modifications que nous y avons apportées pour consolider son fonctionnement, afin d'éviter une interaction pouvant être surprenante, voire dangereuse pour l'OH. La dernière partie sera dédiée aux tests de validation

de l'algorithme de génération de trajectoire, à travers la simulation de différents cas possibles.

Le **cinquième** et dernier **chapitre** est destiné à la présentation d'un schéma de commande modifié, comprenant le générateur de trajectoire présenté dans le chapitre 4, lorsqu'il est appliqué au modèle le plus abouti du robot DENS0 VP-6242G issu de l'identification et présenté au chapitre 3. Des simulations d'une interaction homme-robot seront présentées et discutées dans ce chapitre, afin d'illustrer la méthode développée pour répondre à la problématique de co-manipulation pour la manutention de charges lourdes. Nous concluons ce chapitre par quelques perspectives portant sur des développements qu'il serait intéressant d'apporter au schéma de commande.

Ce mémoire s'achèvera par un bilan des travaux effectués durant la thèse. Nous proposerons également quelques perspectives pour les travaux ultérieurs.



# Quelques généralités sur le concept de co-manipulation Homme-Robot

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>11</b>
<b>2.2</b>	<b>Différentes catégories de co-manipulation</b>	<b>13</b>
<b>2.3</b>	<b>État de l'art</b>	<b>15</b>
<b>2.4</b>	<b>Sécurité de la co-manipulation homme-robot</b>	<b>22</b>
<b>2.5</b>	<b>Plus d'exigences sur la co-manipulation dans l'industrie</b>	<b>24</b>
<b>2.6</b>	<b>Conclusion</b>	<b>26</b>

---

## 2.1 Introduction

La collaboration entre les humains et les robots est de plus en plus recommandée dans plusieurs domaines, y compris le domaine de l'industrie. Dans ce chapitre, nous allons aborder le contexte de la co-manipulation homme-robot. Nous expliquerons d'abord la nécessité d'une coopération entre un robot et un être humain. Comme il y a plusieurs façons de faire coopérer un homme avec un robot, nous exposerons les différents types de coopération pouvant exister. Les différentes solutions existantes et les problèmes d'une co-manipulation homme-robot seront aussi discutés. Enfin, nous présenterons le problème de co-manipulation tel que nous l'abordons, et sa mise en perspective avec les autres approches similaires.

### 2.1.1 Définition

En robotique, la co-manipulation consiste à réaliser une tâche précise, d'une façon conjointe par un robot et un être humain (co-manipulation homme-robot), ou bien par deux robots (co-manipulation robot-robot). Nous pouvons donc avancer qu'une co-manipulation nécessite le partage de l'espace de travail, mais aussi le moment d'intervention. La co-manipulation homme-robot a pour objectif de profiter des

capacités des robots (force, précision, ...) et de la capacité décisionnelle des humains pour réaliser des tâches.

Dans cette étude, nous nous intéressons à la co-manipulation homme-robot destinée au secteur industriel. En industrie automobile par exemple, elle peut intervenir pour des tâches de montage, d'assemblage ou de déplacement de charges lourdes.

### 2.1.2 Nécessité de la co-manipulation

Depuis leur apparition, les robots sont de plus en plus présents dans différents secteurs économiques, notamment dans le domaine de l'industrie, mais aussi en chirurgie, en médecine de la rééducation, en agriculture, en construction, dans le domaine des services, de transport, *etc.* Pour des raisons de rapidité, de précision, et parfois de sécurité, les robots ont permis de remplacer l'être humain pour réaliser beaucoup de tâches comme dans les chaînes de production ou bien pour l'intervention dans des milieux dangereux ou même difficilement accessibles par l'être humain.

Dans le secteur industriel, la robotisation des chaînes de production a permis d'augmenter la productivité grâce aux capacités des robots manipulateurs modernes, aussi bien en terme de vitesse qu'en terme de précision ou d'endurance. De plus, la manipulation de charges lourdes nécessite une force importante qui ne peut, la plupart du temps, qu'être assurée par des robots. Ces avantages ont permis d'optimiser le temps de production, d'avoir des produits de qualité, et d'exempter l'opérateur humain de la nécessité de produire des efforts importants.

Cependant, les tâches compliquées sont difficilement achevées par les robots, notamment la manipulation des pièces de dimensions différentes, ou dans un environnement sous contraintes imprévues et non modélisées. La décision limitée des robots dans ces cas peut conduire non seulement à ne pas accomplir la tâche, mais peut aussi entraîner des dégâts matériels. Par ailleurs, un opérateur humain peut générer des décisions beaucoup plus appropriées qu'un robot, lors de la manipulation des pièces de caractéristiques géométriques dissemblables, surtout si elle se fait dans un environnement sous contraintes dynamiques.

De ce qui précède, nous pouvons considérer qu'une co-manipulation homme-robot est indispensable dans le secteur industriel. Néanmoins, pour une collaboration entre un robot et un être humain, le contrôle du robot doit être maîtrisé, afin de garantir la sécurité de l'opérateur, qui est en contact direct avec le robot, tout en s'assurant que les deux accomplissent la tâche correctement.

### 2.1.3 Classification des interactions homme-robot

Selon la fréquence de l'interaction entre un humain et un robot, nous pouvons classer les différents types d'IPHR en interaction de soutien, de collaboration et de coopération [73].

Les *interactions de soutien* correspondent aux interactions où le robot ne fait pas partie de l'exécution de la tâche, mais il fournit à l'humain les outils, les matériaux et les informations nécessaires pour optimiser les performances de la tâche à réaliser

par l'humain. Par exemple, les robots d'aide au shopping pour aider les personnes âgées [39] (figure 2.1). Dans cette catégorie, l'interaction physique est peu fréquente



FIGURE 2.1 – Le shopping avec un robot [39].

et transitoire. Pour soutenir la sécurité ainsi que ces interactions physiques limitées, une communication homme-robot bien structurée est essentielle.

S'agissant des *interactions collaboratives*, le robot et l'humain travaillent tous les deux sur la tâche. Le travail étant divisé entre le robot et l'humain, chacun complétant les parties de la tâche les mieux adaptées à leurs capacités, mais interagissant plus fréquemment. Dans ces scénarios, l'humain réalise les tâches nécessitant de la dextérité humaine ou de la prise de décision, alors que le robot réalise des tâches peu adaptées à une intervention humaine directe. Par exemple, lorsqu'il y a des mouvements répétitifs ou nécessitant une force importante, des dépôts chimiques ou des placements de précision.

Les *interactions coopératives*, connues sous le nom co-manipulation, incluent les interactions en effort avec les êtres humains. Dans ce cas, l'humain et le robot travaillent en contact physique direct ou en contact indirect à travers un objet commun. Il y a un contrôle partagé et continu de la tâche. Les interactions coopératives sont présentes en réhabilitation médicale (robotique de réhabilitation [64]), en chirurgie [82], et en industrie pour le transport, la manipulation et l'assemblage des pièces [43].

## 2.2 Différentes catégories de co-manipulation

Nous pouvons définir trois catégories d'une co-manipulation homme-robot : série, parallèle et exosquelette.

### 2.2.1 La co-manipulation parallèle

Ce type de co-manipulation présente le cadre de notre étude, pour lequel le robot et l'opérateur sont les deux en contact avec l'outil. Ils forment deux chaînes parallèles au niveau des liaisons. L'effort appliqué sur l'outil est la somme des deux efforts générés par le robot et l'opérateur. Le transport et la manutention d'une charge

lourde est un exemple typique de cette catégorie [59, 55], où le robot et l'opérateur collaborent pour transporter/manipuler la pièce (figure 2.2).

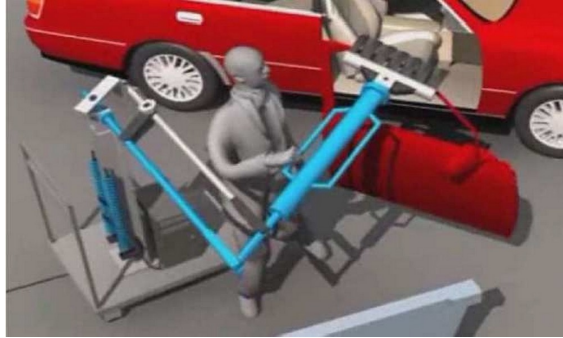


FIGURE 2.2 – Un ouvrier en co-manipulation avec un robot pour une opération de montage dans l'industrie.

### 2.2.2 La co-manipulation exosquelette

La co-manipulation par exosquelette correspond à des robots ayant des mécanismes parallèles. De tels robots possèdent des points de contact multiples avec l'être humain, mais ils ont un fonctionnement très différent par rapport aux co-manipulateurs parallèles présentés ci-dessus. Ce type de robot est notamment utilisé en réhabilitation [26] (figure 2.3).



FIGURE 2.3 – Co-manipulation exosquelette pour la rééducation du bras [42].

### 2.2.3 La co-manipulation série

La co-manipulation série est caractérisée par une série formée par l'opérateur qui est en contact avec le robot (l'opérateur tient le robot), et le robot qui est à son tour en contact avec l'outil. Les trois forment une chaîne cinématique série. La position de l'outil est le résultat de la composition du mouvement du robot avec celui de l'opérateur. Ce type de co-manipulation peut être utilisé en chirurgie (figure 2.4).



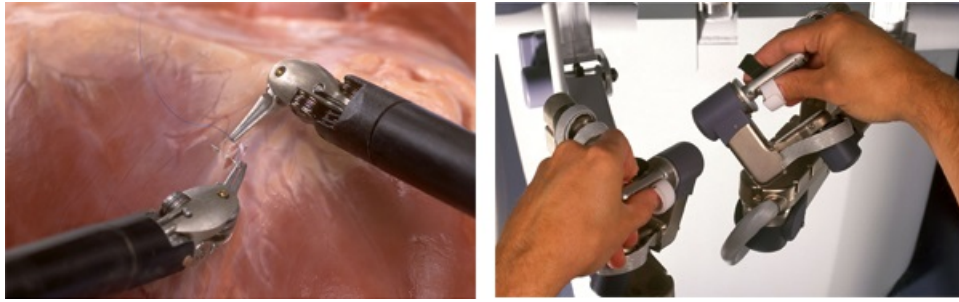


FIGURE 2.4 – Des robots utilisés en mode de co-manipulation série pour la suture chirurgicale [3].

En général, les robots utilisés dans ce mode de fonctionnement sont des robots qui possèdent peu d'articulations, typiquement une ou deux.

## 2.3 État de l'art

Une étude approfondie de la littérature de ce sujet nous a aidé à découvrir les différentes solutions de co-manipulation parallèle qui ont été conçues. Elles sont décrites dans les sous-sections qui suivent.

### 2.3.1 Programmation par démonstration

Connue sous le nom anglophone "*Walk-through programming*" ou parfois "*Lead-through programming*", cette solution a été introduite pour la première fois en 1999 dans [18] pour automatiser les processus de soudage dans l'industrie de la construction navale. Cette technique est caractérisée par deux phases. Une première phase de démonstration (ou programmation) où l'opérateur déplace manuellement le robot le long de la trajectoire à exécuter, de manière autonome, afin qu'il l'enregistre. Dans une phase suivante (phase de répétition), le robot reproduit, sans aide de l'opérateur, la trajectoire plusieurs fois.



FIGURE 2.5 – Robot en phase de démonstration.



En général, cette technique nécessite l'installation d'un capteur d'effort (mesures de force et de couple) et la conception d'une architecture de contrôle, souvent basée sur la maîtrise de l'impédance [31], [18].

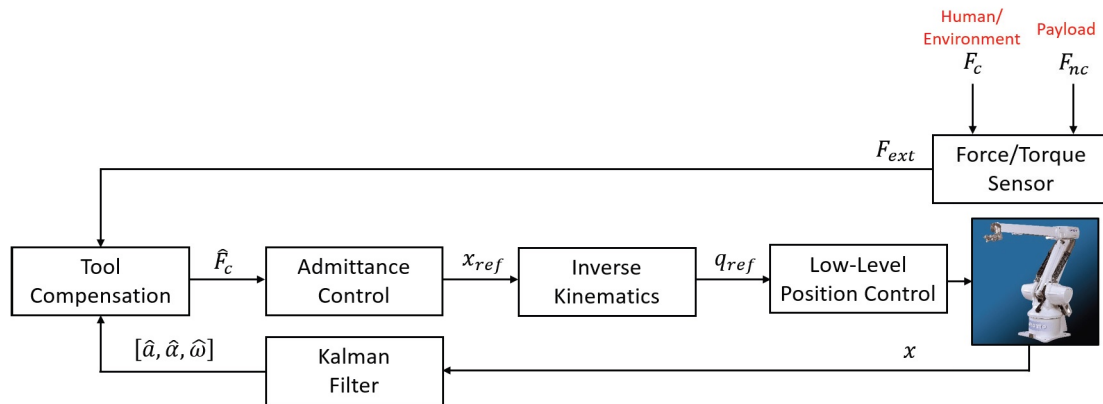


FIGURE 2.6 – Technique de programmation par démonstration basée sur la commande en impédance, et la compensation de la dynamique de l'outil [31].

L'inconvénient majeur de cette méthode reste la non-prise en compte des contraintes et obstacles imprévus dans l'espace de travail du robot, lorsqu'il est en phase de répétition.

### 2.3.2 Compensation du poids de l'outil/charge

Cette méthode est apparue pour la première fois avec l'invention de l'*outil d'équilibrage de Powell* [9]. Cet outil est constitué essentiellement d'un enrouleur pneumatique suspendu et d'un câble faisant la liaison entre ce dernier et la charge. L'équilibreur de Powell permet de compenser le poids d'une charge portée à travers l'application d'un effort constant vers le haut, quand un opérateur peut manipuler la charge portée. L'invention de la machine de levage électrique à équilibrage automatique [10] est considéré comme une amélioration de l'outil de Powell. Pour cette machine, Kornely a utilisé un moteur électrique et un capteur d'effort afin d'assurer une compensation adaptée au poids de la charge. Plus récemment, l'amplificateur de puissance humaine pour les manœuvres verticales (connu sous le nom de *machine de Kazerooni*), introduit dans [12], permet de manipuler plus facilement une charge portée. Son fonctionnement s'appuie sur l'utilisation d'un capteur d'effort installé au niveau du poignet de l'opérateur. Le capteur de force mesure la force verticale appliquée par l'opérateur et délivre un signal qui permet à un contrôleur avec un actionneur de générer un pourcentage prédéterminé de la force nécessaire pour soulever, manipuler ou abaisser la charge. Le pourcentage restant de la force nécessaire est fourni par l'opérateur. La charge semble donc plus légère pour l'opérateur, mais l'opérateur ne perd pas le sens de la levée contre les forces de gravitation et d'inertie provenant de la charge.

La figure 2.7 illustre un exemple des systèmes de compensation de poids de la charge conçu sur le principe de la machine de Kazerooni. Le système applique une

force pour compenser le poids, et l'opérateur applique une force pour manipuler la pièce.



FIGURE 2.7 – Système de levage permettant une compensation du poids de la charge [8].

En revanche, les machines de levage conçues sur ce principe sont des systèmes à câbles, ce qui nécessite de fixer la charge de sorte que son centre de gravité soit aligné avec les câbles de levage. De plus, le point de contact entre le câble et la charge est, dans le meilleur des cas, situé au-dessus du centre de gravité. Cela limite la flexibilité de la manipulation dans le plan horizontal seulement. Les robots manipulateurs à articulations rigides sont considérés comme la solution la plus adéquate pour manipuler des pièces dans l'espace tridimensionnel, à travers les six mouvements de base, c'est-à-dire les trois translations et les trois rotations.

### 2.3.3 Augmentation d'effort

La notion d'augmentation d'effort regroupe deux problèmes de co-manipulation. Le premier concerne l'amplification de l'effort appliqué par l'opérateur. Le second concerne l'amplification de l'effort ressenti par ce dernier.

#### Augmentation de l'effort appliqué par l'opérateur

L'idée d'augmenter l'effort appliqué par un opérateur vient, à l'origine, du besoin de transporter et manipuler des charges lourdes. Dans ce cadre, la société *General Electric* a inventé le premier robot co-manipulateur appelé *Hardiman* permettant l'amplification de la force de l'humain [30]. Le robot était destiné à la télé-manutention de charges pesant jusqu'à 680 kg. Cependant, vu le poids important de cet engin,

plus de deux fois le poids de sa charge utile, et compte tenu de sa complexité, un tel robot n'a jamais fonctionné de manière suffisamment stable. Par conséquent, ce type de robot co-manipulateur n'a jamais pu être mis sur le marché et le projet a été abandonné.

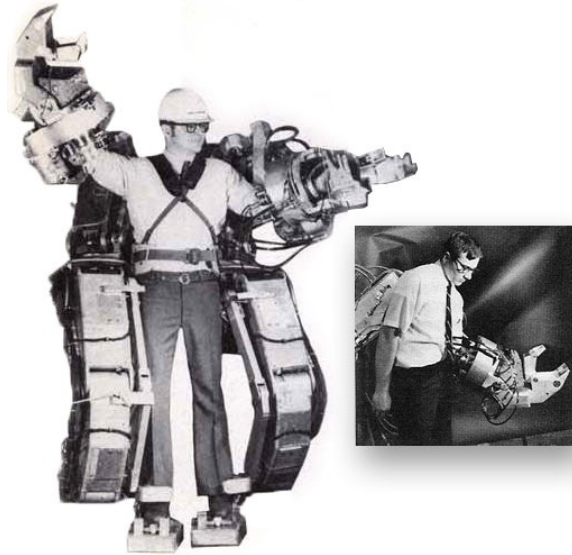


FIGURE 2.8 – Le projet Hardiman inventé par *General Electric*.

Cette tentative de *General Electric* d'introduire le concept de robots amplificateurs d'effort a permis plus tard d'ouvrir la voie à des recherches afin de mettre en pratique ce type de robots. Dans [46], les auteurs se sont intéressés à la co-manipulation directe au lieu la télé-manipulation, où l'opérateur est en contact direct avec un robot faisant partie de la catégorie des « *Extenders* », qui présente une classe de robots manipulateurs portés par les êtres humains pour accroître leur force. Cependant, la sécurité de l'opérateur en contact avec un robot de ce type était loin d'être garantie vu le mécanisme conçu.



FIGURE 2.9 – Le projet *Extender*.

D'autres chercheurs se sont intéressés à la stabilité et la robustesse du système à travers le critère de passivité [32] et [53]. Ce principe a aussi été récemment repris dans [57] pour des applications dans l'industrie agroalimentaire. Dans [57] et [58], il était proposé une amplification de la force de l'opérateur avec un facteur d'amplification fixe (10 fois la force de l'opérateur par exemple). Cependant, une augmentation soudaine de la force de l'opérateur est aussi amplifiée. Par conséquent, le mouvement du robot dans ce cas peut être dangereux pour l'opérateur.

### Augmentation de l'effort ressenti par l'opérateur

L'augmentation de l'effort ressenti par l'opérateur présente presque l'inverse du cas de figure précédent. Dans ce cas, les rôles sont inversés. L'effort appliqué par l'organe terminal est amplifié avant d'être ressenti par l'opérateur. Aussi, il faut disposer d'un robot beaucoup plus petit et d'un capteur d'efforts plus sensible aux efforts de faible amplitude. L'application la plus pertinente de ce type de co-manipulation est le domaine chirurgical. Cette fonctionnalité a été implémentée sur le robot MC<sup>2</sup>E en chirurgie laparoscopique afin de permettre au chirurgien de mieux ressentir les efforts d'interaction entre l'instrument et les tissus biologiques [25].

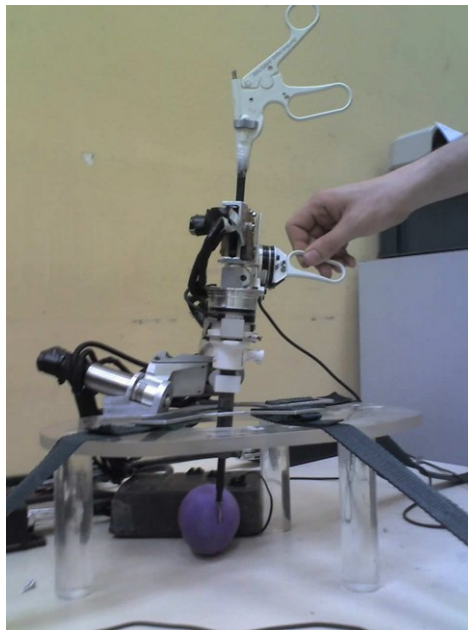


FIGURE 2.10 – Robot de laparoscopie MC<sup>2</sup>E [25].

### 2.3.4 Filtrage des tremblements

L'étude menée dans [76], [56] et [65] concerne la co-manipulation homme-robot en microchirurgie, notamment la chirurgie de la rétine où l'intervention chirurgicale correspondante nécessite des mouvements très précis et de faible amplitude. Les tremblements générés par l'opérateur sont inadmissibles dans ce cas. Un faible excès de force générée peut être dangereux pour l'intégrité de la rétine du patient. Une co-manipulation entre le chirurgien et un robot rigide sur lequel est installé un capteur de force de haute sensibilité, permet d'éliminer les tremblements du chirurgien et les

éventuels excès d'efforts de ce dernier. La figure 2.11 illustre un robot destiné à une telle application.

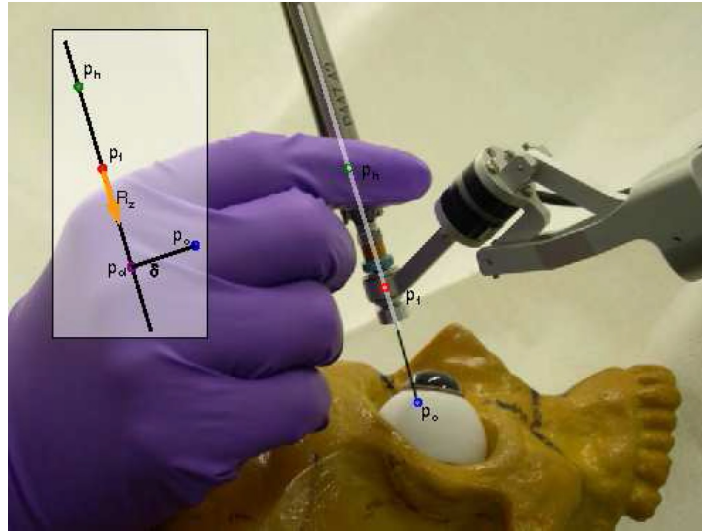


FIGURE 2.11 – Le robot développé dans [65].

### 2.3.5 Co-manipulation contrainte

Une co-manipulation contrainte correspond à une manipulation de l'opérateur, avec des contraintes imposées par le robot. Les contraintes imposées par le robot peuvent être des limites virtuelles de l'espace de travail, ou des trajectoires virtuelles dont il ne faut pas s'écarter. Nous pouvons ainsi distinguer deux catégories de contraintes : les contraintes de type *guide virtuel* et celles de type *butée virtuelle*.

#### Le guide virtuel

Le guide virtuel est un ensemble de contraintes appliquées sur l'organe terminal du robot, qui est en contact avec l'opérateur. Ces contraintes permettent de guider le robot selon une trajectoire prédéfinie. Par exemple, pour une opération de perçage, un opérateur impose à l'outil la position et l'effort de perçage, tandis que le robot aide l'opérateur à garder le forêt bien vertical par rapport à la surface à percer. Un autre exemple est illustré par la figure 2.12, qui consiste au placement d'objets dans un placard à l'aide de guides virtuels pour éviter les collisions avec les étagères.





FIGURE 2.12 – Placement d’objets dans un placard à l’aide de guides virtuels pour éviter les collisions avec les étagères [68].

La trajectoire contrainte peut être définie par démonstration (apprentissage), comme expliqué dans [69, 68]. L’opérateur humain crée des guides virtuels par démonstration à travers une méthode itérative basée sur l’apprentissage kinesthésique. Ensuite, les trajectoires sont regroupées dans une bibliothèque de guides virtuels. Le robot peut dès le début de la co-manipulation détecter la trajectoire à suivre grâce à la bibliothèque des trajectoires. Par conséquent, le robot peut continuer le mouvement en autonomie.

### La butée virtuelle

La butée virtuelle sert de limite surfacique à l’espace de travail. Par exemple, dans une unité de production composée de deux robots manipulateurs où l’un d’eux fonctionne de manière complètement indépendante, et l’autre est en interaction physique avec un opérateur humain pour une co-manipulation. Les deux robots effectuent leurs tâches dans des espaces de travail voisins. Une butée virtuelle sur l’espace de travail du robot co-manipulant avec l’opérateur permet de ne pas pénétrer dans la zone de travail du robot indépendant. Une telle pénétration peut conduire à des collisions et présenter un danger pour la vie de l’opérateur. Le principe de la butée virtuelle a aussi été appliqué en chirurgie. Dans [40], les auteurs se sont intéressés aux opérations de pose de prothèse de genou en orthopédie.



FIGURE 2.13 – Le système de co-manipulation *Acrobot* utilisé en chirurgie orthopédique [40].

Le chirurgien co-manipule avec un robot de type *Acrobot* (figure 2.13) pour usiner une zone de l'os du patient, où sera installée la prothèse. La butée virtuelle permet d'effectuer la coupe de l'os du genou dans une région bien définie, sans qu'il y ait de risque pour le patient ou qu'il y ait des dommages collatéraux.

## 2.4 Sécurité de la co-manipulation homme-robot

La sécurité de l'opérateur est l'exigence la plus importante lors d'une interaction physique avec un robot. Dans l'industrie, le robot et la pièce co-manipulée ne doivent présenter aucun danger pour la vie de l'opérateur ou pour son intégrité physique, pendant toutes les étapes de la co-manipulation. Le danger d'un robot industriel sur la sécurité de l'opérateur peut être dû à sa dimension ou bien son poids important. Il peut également provenir de ses déplacements à vitesse élevée. Dans ce cas, les conséquences d'une collision avec un opérateur peuvent être très graves, voire mortelles. Les premières solutions de sécurité de l'opérateur consistaient à assurer une distance de sécurité permanente entre le robot et l'opérateur, à travers l'installation de capteurs de proximité, des caméras, *etc.* Ces derniers permettent de détecter si la distance entre les deux est respectée ou non [72, 67, 52, 77]. En cas de non respect de cette distance de sécurité, le robot est immédiatement mis à l'arrêt pour éviter tout risque de collision. Cette solution a été incluse dans la norme ISO-10218 :1992 [11], qui expose également des recommandations sur les considérations de sécurité lors de l'utilisation des robots manipulateurs industriels. Cependant, cette solution ne prend pas en compte le cas d'un contact physique entre le robot et l'opérateur, devenu au fil du temps nécessaire dans l'industrie. La norme ISO 10218-1 :2006, qui est une actualisation de la norme précédente, a défini pour la première fois des règles de sécurité claires pour les interactions physiques homme-robot. En particulier, la vitesse maximale de l'organe terminal d'un robot en interaction avec un être humain ne doit pas dépasser les  $0.25\text{ m/s}$ . La force et la puissance dynamique maximales sont limitées

à 150  $N$  et 80  $W$ , respectivement [13].

Pour la sécurité des humains en contact direct avec des robots, plusieurs travaux ont été menés pour implémenter une interaction homme-robot sûre. Parmi les premières solutions qui étaient envisagées, il y avait la conception d'un robot à base d'un tronc visco-élastique constitué d'éléments mécaniques tels que des ressorts et des amortisseurs [60, 80]. L'objectif était d'absorber les collisions et d'en atténuer les effets pour réduire le traumatisme éventuel.

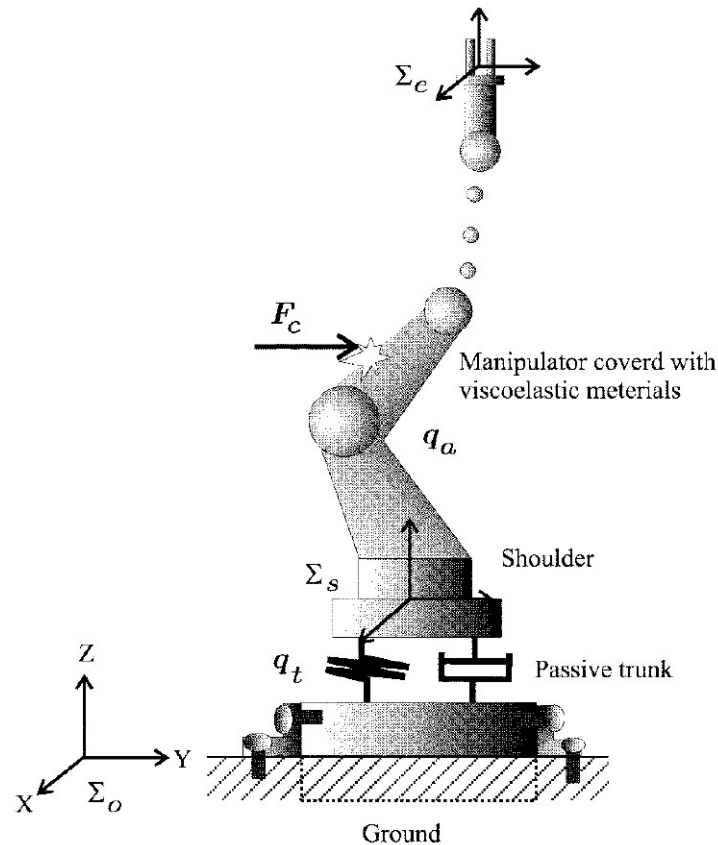


FIGURE 2.14 – Modèle d'un robot conçu à base d'un tronc visco-élastique [60].

Cependant, cette conception était rarement appliquée dans l'industrie à cause des oscillations du robot, cela suppose de la part de l'opérateur en contact avec le robot d'appliquer un effort important pour le rendre précis. De plus, les éléments visco-élastiques sont inutiles pour les robots industriels de forte puissance car les collisions avec les êtres humains demeurent malgré tout trop risquées.

Une autre solution proposée pour protéger un opérateur en contact direct avec un robot consistait à distinguer la force d'interaction et la force de collision. Dans [62], les auteurs ont proposé d'équiper le robot avec deux capteurs de force/couple, l'un situé au niveau de l'organe terminal, et l'autre à la base du robot. Une éventuelle collision serait traduite par une variation du signal mesuré par l'un des deux capteurs de force installés. Selon la force de contact tolérée, un arrêt d'urgence peut être activé en cas de collision.



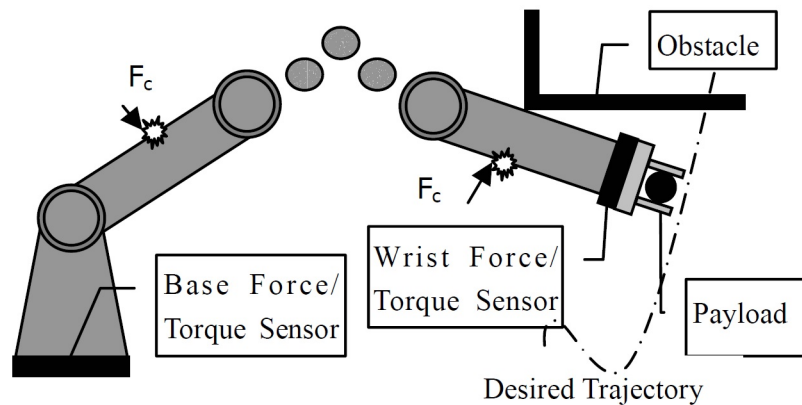


FIGURE 2.15 – Principe de détection de collision à base de deux capteurs de force [62].

Une méthode fréquentielle a été proposée dans [54], pour la distinction entre les forces extérieures appliquées dans le cadre d'une co-manipulation, et les forces de collision. Le principe mathématique sous-jacent repose sur la transformée de Fourier. En effet, les caractéristiques fréquentielles de ces deux types de forces ne sont pas les mêmes. Les forces générées par les humains sont lisses et la réaction pour un contact inattendu est plus rapide. La force mesurée est présentée dans l'espace fréquentiel car le module de la force ne peut pas être un bon indicateur. Dans les deux cas, la force peut être du même ordre de grandeur. Les forces appliquées pour la co-manipulation ont un spectre plutôt localisé aux basses fréquences, tandis que les forces de réaction lors d'un contact inattendu sont caractérisées par des fréquences plus élevées. La définition du seuil de la force dépend essentiellement des paramètres de la commande en impédance implémentée. Cependant, même si ces méthodes peuvent permettre de distinguer la force appliquée pour la co-manipulation et celle de collision, le robot ne se met à l'arrêt qu'après la détection de la collision. Cela signifie qu'il est déjà trop tard et cela peut être dangereux pour la vie de l'opérateur.

## 2.5 Plus d'exigences sur la co-manipulation dans l'industrie

Le problème de la sécurité en co-manipulation homme-robot a été largement abordé dans la littérature. Ainsi, d'autres problèmes peuvent exister lors d'une telle interaction, notamment la minimisation de la force appliquée par l'opérateur, la qualité de suivi des consignes de l'opérateur par le robot, le non dépassement des limites de l'espace de travail et la stabilité de la commande implémentée.

### 2.5.1 Assistance du robot pendant la co-manipulation

L'assistance en co-manipulation signifie que l'opérateur dirige et le robot s'occupe de générer la force nécessaire pour déplacer ou manipuler l'objet. Il s'établit alors

entre les deux une relation de type *maître-à-esclave* au bénéfice de l'opérateur humain. Par exemple dans l'industrie lourde (comme la métallurgie), la co-manipulation d'une pièce avec un robot qui a une grande inertie est difficile pour l'opérateur, s'il devait appliquer lui-même un effort pour manipuler la pièce. Il est donc indispensable de garantir que l'opérateur n'exerce qu'un effort indicatif pour guider le robot, qui à son tour génère la force nécessaire à la manutention de la pièce.

### 2.5.2 Transparence de la co-manipulation

La transparence qualifie la capacité d'un robot à suivre les mouvements imposés par l'opérateur sans aucune force de résistance ressentie de sa part. Pendant une co-manipulation homme-robot, les intentions de l'opérateur doivent être transformées avec souplesse, en mouvement conforme à ce qu'il désire. De plus, le robot ne doit pas restreindre le mouvement voulu par l'opérateur, sauf en cas de dépassement de l'espace de travail prédéfini. Ainsi, l'opérateur ne doit pas percevoir d'effet perturbateur dû aux forces de résistance (inertie, frottements,..) du robot.

### 2.5.3 Atteignabilité de la trajectoire générée

Certains algorithmes de co-manipulation sont conçus autour d'un générateur de trajectoire en temps-réel [44, 45]. C'est aussi le cas de notre étude. Si la trajectoire générée conduit le robot à la limite de son espace de travail, cela va l'entraîner dans une configuration de singularité. Le contrôle du robot dans une telle configuration ne peut être assuré correctement. Il est donc indispensable de corriger la trajectoire dans ce cas afin d'assurer que la co-manipulation homme-robot évolue à l'intérieur de l'espace atteignable par le robot. La correction par asservissement peut être envisagée soit pour garantir que le robot n'atteint pas les limites de son espace de travail, évitant ainsi les singularités potentielles, soit pour ne pas dépasser une butée virtuelle délimitant l'espace de travail.

### 2.5.4 Stabilité du robot en co-manipulation

Comme elle a été définie dans [29], la stabilité de l'interaction homme-robot signifie les conditions de stabilité du robot en boucle fermée, soit « *le robot et la technique de commande appliquée pour l'interaction* » (commande en impédance, par passivité, etc). Pour une trajectoire de référence incluse dans l'espace de travail d'un robot co-manipulant avec un opérateur, sa réponse doit être aussi à l'intérieur de ce dernier. Une éventuelle réponse qui tend à sortir le robot de son espace de travail, va le mettre dans une configuration de singularité (extension complète du bras), qui va déstabiliser le système bouclé. La stabilité de l'interaction homme-robot est donc essentielle pour la sécurité de l'opérateur.

## 2.6 Conclusion

Dans ce chapitre, nous avons brièvement présenté l'ensemble des études qui ont été menées dans le cadre de la co-manipulation homme-robot. Les contraintes définies pour une co-manipulation dépendent beaucoup du domaine d'application (l'industrie, la chirurgie, *etc*). Suivant ces contraintes, plusieurs solutions ont été proposées. Dans cette étude, nous nous intéressons à la co-manipulation avec un robot manipulateur industriel pour la manutention de charges lourdes (compensation du poids), tout en assurant l'assistance, la sécurité et la transparence de la co-manipulation. Dans ce qui suit, nous allons nous intéresser d'abord à la modélisation et à l'identification des paramètres des robots manipulateurs de type série, dans l'objectif d'avoir un modèle sur lequel nous pouvons nous baser pour implémenter notre algorithme de co-manipulation homme-robot développé, qui sera présenté plus tard.

# Modélisation et identification des paramètres dynamiques d'un robot

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>27</b>
<b>3.2</b>	<b>Définitions</b>	<b>28</b>
<b>3.3</b>	<b>Modélisation des robots séries à articulations rotoïdes</b>	<b>30</b>
<b>3.4</b>	<b>Modélisation du robot Denso VP-6242G</b>	<b>45</b>
<b>3.5</b>	<b>Identification des paramètres dynamiques du robot</b>	<b>50</b>
<b>3.6</b>	<b>Identification des paramètres dynamiques du robot Denso VP-6242G</b>	<b>57</b>
<b>3.7</b>	<b>Conclusion</b>	<b>82</b>

---

## 3.1 Introduction

Les robots manipulateurs industriels de type série sont équipés d'actionneurs permettant de mettre en mouvement les différents organes du robot. La plupart de ces actionneurs sont électriques et correspondent, par exemple, à des moteurs à courant continu. Il existe également des robots avec des actionneurs hydrauliques lorsque des efforts importants sont requis pour exécuter des mouvements à vitesse variable dans un contexte où l'encombrement doit être réduit [15, 16].

Les robots manipulateurs industriels sont également équipés de capteurs disposés à différents endroits du robot pour mesurer différentes grandeurs pertinentes, permettant à chaque instant de renseigner l'utilisateur sur l'état du robot, comme sa configuration géométrique par le biais de capteurs de position, ou bien les efforts qu'il développe par le biais de capteurs mesurant le courant consommé par chaque moteur électrique, ou la pression du fluide dans le cas de moteurs hydrauliques. Les capteurs de position permettent de fournir une mesure des coordonnées de position articulaire. Certains robots peuvent également être équipés de capteurs de vitesse fournissant une mesure des coordonnées de vitesse articulaire. Les capteurs de force peuvent fournir une

mesure des coordonnées d'un vecteur force, voire d'un vecteur moment, qui s'applique au robot à l'endroit où se situe un tel capteur.

L'obtention d'un modèle du robot décrivant son comportement dynamique entrées-sorties est essentiel pour pouvoir appliquer la plupart des techniques de commande récentes qui reposent sur la connaissance d'un tel modèle, et plus particulièrement sur la connaissance de sa structure mathématique et de ses paramètres. Si par le terme « entrées » nous faisons référence aux signaux de commande des actionneurs et par le terme « sorties » nous faisons référence aux signaux récupérés par les capteurs de position articulaire du robot, alors nous désignerons par *Modèle Dynamique Direct* (MDD) un modèle qui permette d'étudier l'évolution temporelle des signaux de sortie en réponse à une sollicitation imposée par les signaux d'entrée du robot. Nous désignerons par *Modèle Dynamique Inverse* (MDI) un modèle mathématique issu d'un MDD où les rôles des signaux d'entrée et de sortie sont permutés. C'est-à-dire que les signaux d'entrée du MDD deviennent les signaux de sortie du MDI et la totalité, ou bien seulement une partie des signaux de sortie du MDD deviennent les signaux d'entrée du MDI.

Dans ce chapitre, nous allons commencer par présenter la manière d'obtenir la structure mathématique paramétrée d'un MDI, puis nous présenterons la manière d'obtenir une estimation de ses paramètres par le biais de l'identification de ce modèle, à partir de données expérimentales. L'obtention d'un MDI nécessite au préalable de construire les modèles géométrique et cinématique directs (respectivement MGD et MCD). La construction du MGD consiste à déterminer les équations définissant la position et l'orientation de l'organe terminal, exprimées dans un repère fixe, à partir des coordonnées articulaires de position et des paramètres géométriques du robot. La construction du MCD consiste quant à elle à déterminer les équations définissant la vitesse linéaire et la vitesse angulaire de l'organe terminal, exprimées dans un repère fixe, à partir des coordonnées articulaires de position et de vitesse ainsi que des paramètres géométriques du robot. Ces derniers correspondent aux différents angles et distances entre deux repères consécutifs attachés aux corps du robot.

Ce chapitre est organisé de la manière suivante. La section 3.3 présente les outils mathématiques permettant l'obtention du MDI. La section 3.4 correspond à l'application de ces outils au cas du robot DENSO VP-6242G. Dans la section 3.5, nous focaliserons la présentation sur la démarche d'identification d'un MDI à partir de données expérimentales. Pour finir, la section 3.6 permettra d'appliquer cette démarche au cas du robot DENSO VP-6242G.

## 3.2 Définitions

1. **Robot manipulateur série** : Ce terme désigne une succession de corps rigides en chaîne cinématique ouverte, composée ou arborescente. Les corps de la chaîne sont liés entre eux par des liaisons appelées *articulations*.

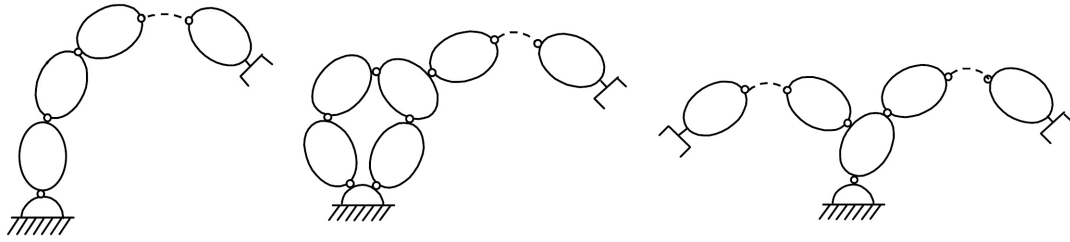


FIGURE 3.1 – Chaînes cinématiques ouverte (gauche), composée (milieu), et arborescente (droite).

Le robot est fixé à travers sa **base** qui est généralement considérée comme le corps 0 du robot. Au dernier corps du robot, il est fixé un outil appelé **organe terminal**.

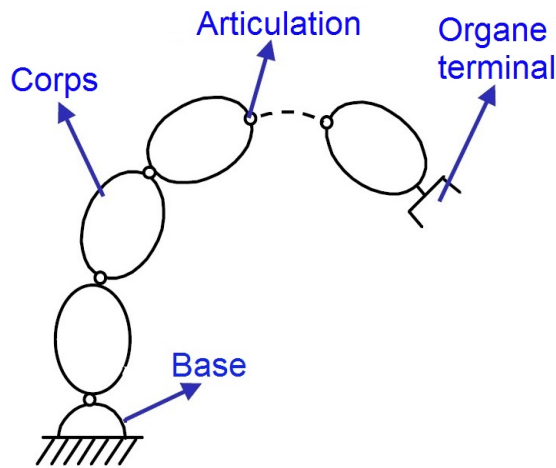


FIGURE 3.2 – Composantes d'un robot série.

Les principales articulations utilisées pour concevoir des robots série sont les articulations rotoïdes et prismatiques.

2. **Articulation rotoïde** : signifie que le mouvement relatif entre les deux corps est une rotation autour de l'axe de l'articulation.
3. **Articulation prismatique** : signifie que le mouvement relatif entre les deux corps est une translation de direction donnée par l'axe de l'articulation.

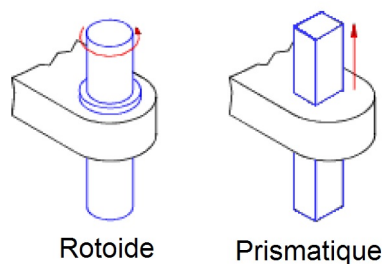


FIGURE 3.3 – Articulations rotoïde et prismatique.

4. **Organe terminal** : cela désigne tout dispositif fixé sur le dernier corps du robot. Il est destiné à manipuler des objets ou à les transformer. Cela peut être une pince, un foret, un couteau, *etc.* L'organe terminal présente l'interface du robot avec son environnement. A noter qu'il n'y a pas d'articulation entre le dernier corps du robot et l'organe terminal.
5. **Degré de liberté (ddl)** : le nombre de ddl d'un robot est défini par le nombre d'articulations munies d'actionneurs appropriés.
6. **Variable articulaire** : pour chaque articulation, elle désigne la position (pour articulation prismatique), ou la position angulaire (pour articulation rotoïde) du corps correspondant dans un repère attaché à ce dernier.
7. **Espace articulaire** : une position et une orientation de l'organe terminal du robot est exprimée dans l'espace articulaire si elle est donnée en fonction des différentes positions articulaires. La dimension de l'espace articulaire est égal au nombre de degré de liberté du robot.
8. **Espace opérationnel/cartésien** : la position et l'orientation de l'organe terminal sont exprimées dans un repère fixe (repère galiléen). La dimension de l'espace opérationnel vaut six, dont trois translations et trois rotations.
9. **Actionneur** : nécessaire au niveau de chaque articulation pour permettre au corps correspondant d'être en mouvement. Les principaux types d'actionneurs utilisés pour les robots manipulateurs série sont les moteurs électriques et les vérins hydrauliques.

## 3.3 Modélisation des robots séries à articulations rotoïdes

### 3.3.1 Modèle géométrique direct

Considérons le cas d'un robot à  $n$  articulations rotoïdes. A chaque corps  $i$ , est attaché un repère  $R_i$ , dont l'axe  $z_i$  est porté par l'axe de rotation de l'articulation. On définit aussi un repère de base  $R_0$  qui est immobile. Il correspond à un repère galiléen. L'un des choix de ce repère est qu'il soit confondu avec le repère  $R_1$  quand la variable articulaire  $q_1$  est nulle. Ce choix nous permet de minimiser le nombre des paramètres géométriques non nuls. Aussi, comme il n'y a pas d'articulation entre le dernier corps du robot et l'organe terminal, le repère  $R_n$  s'attache à ces deux corps.

#### Convention et paramètres de Denavit-Hartenberg Modifiée

Le calcul du modèle dynamique d'un robot se base sur les paramètres géométriques, connus sous le nom de « *paramètres de la convention de Denavit-Hartenberg Modifiée (DHM)* ». Ces paramètres permettent de définir la position et l'orientation d'un repère  $R_i$  dans  $R_{i-1}$ . La figure 3.4 illustre les paramètres de la convention DHM [49] :

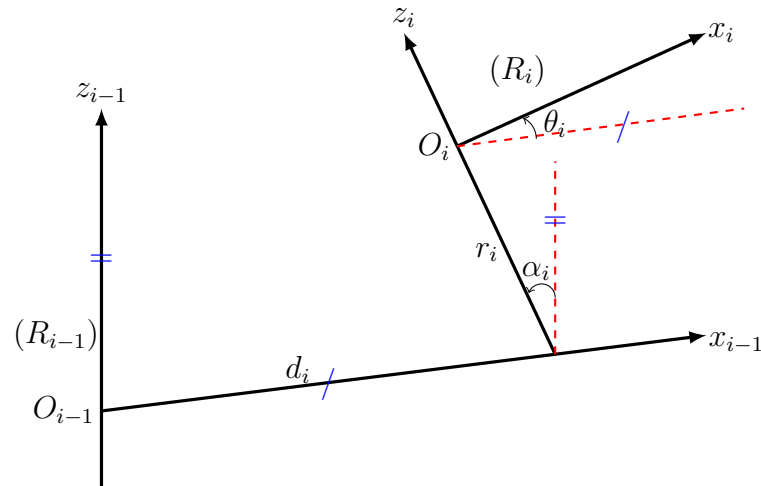


FIGURE 3.4 – Convention de Denavit-Hartenberg Modifiée.

En se basant sur la figure 3.4, nous définissons les paramètres suivants [49] :

$$\begin{cases} \alpha_i = \widehat{z_{i-1}, z_i} / x_{i-1} \\ d_i = \overline{O_{i-1}, z_i} / x_{i-1} \\ r_i = \overline{x_{i-1}, O_i} / z_i \\ \theta_i = \widehat{x_{i-1}, x_i} / z_i \end{cases} \quad (3.1)$$

Tel que :

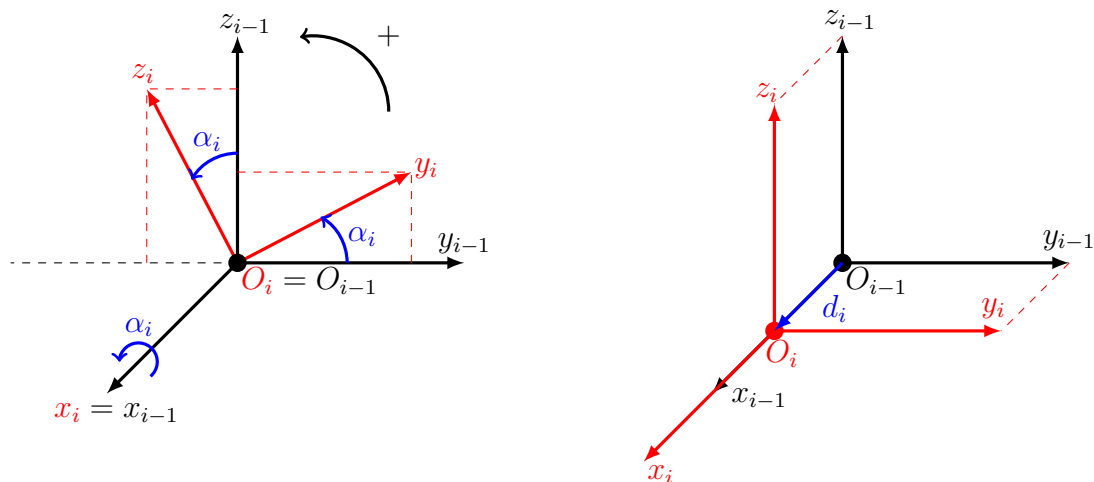
- $z_{i-1}$  : est porté par l'axe de l'articulation  $i - 1$  ;
- $x_{i-1}$  : est porté par la perpendiculaire entre les axes  $z_{i-1}$  et  $z_i$ . Si  $z_{i-1}$  et  $z_i$  se coupent,  $x_{i-1}$  est alors dirigé perpendiculairement au plan formé par les deux axes ;
- $O_{i-1}$  : résulte de l'intersection de la perpendiculaire entre  $z_i$  et  $z_{i-1}$  avec ce dernier.

## Transformations élémentaires

### 1. Rotation et translation selon $x_i$

La figure 3.5 illustre le repère  $R_i$  résultant d'une rotation du repère  $R_{i-1}$  d'un angle  $\alpha_i$  (gauche) autour de l'axe  $x_{i-1}$ , et d'une translation d'une distance  $d_i$  selon  $x_{i-1}$  (droite) :




 FIGURE 3.5 – Rotation et translation selon  $x_i$ .

Les vecteurs unitaires  $x_i$ ,  $y_i$ ,  $z_i$  et l'origine  $O_i$  de  $R_i$  peuvent être exprimés dans le repère  $R_{i-1}$  comme suit [74], [63] :

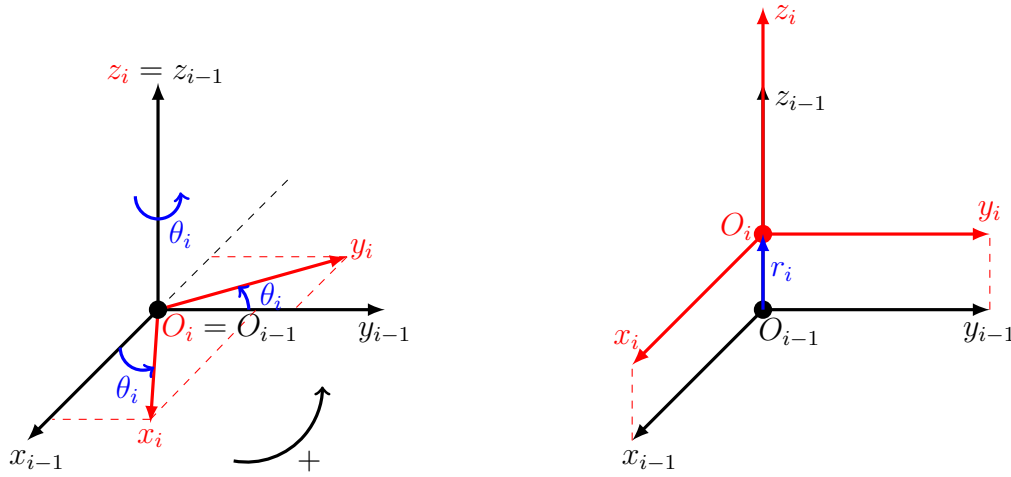
$$x_i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, y_i = \begin{bmatrix} 0 \\ \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix}, z_i = \begin{bmatrix} 0 \\ -\sin(\alpha_i) \\ \cos(\alpha_i) \end{bmatrix}, O_i = \begin{bmatrix} d_i \\ 0 \\ 0 \end{bmatrix}. \quad (3.2)$$

Les matrices de rotation  $Rot(x, \alpha_i)$  et de translation  $Trans(x, d_i)$ , exprimées comme des matrices de transformation homogène en fonction des vecteurs (3.2), sont les suivantes :

$$Rot(x, \alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Trans(x, d_i) = \begin{bmatrix} 1 & 0 & 0 & d_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

## 2. Rotation et translation selon $z_i$

La figure 3.6 illustre le repère  $R_i$  résultant d'une rotation du repère  $R_{i-1}$  d'un angle  $\theta_i$  (gauche), et d'une translation d'une distance  $r_i$  selon l'axe  $x_{i-1}$  (droite) :

FIGURE 3.6 – Rotation et translation selon  $z_i$ .

Les vecteurs unitaires  $x_i$ ,  $y_i$ ,  $z_i$  et l'origine  $O_i$  du repère résultant  $R_i$  peuvent être exprimés dans le repère  $R_{i-1}$  comme suit [74], [63] :

$$x_i = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \\ 0 \end{bmatrix}, y_i = \begin{bmatrix} -\sin(\theta_i) \\ \cos(\theta_i) \\ 0 \end{bmatrix}, z_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, O_i = \begin{bmatrix} 0 \\ 0 \\ r_i \end{bmatrix}. \quad (3.4)$$

Les matrices de rotation  $Rot(z, \theta_i)$  et de translation  $Trans(z, r_i)$  exprimées comme des matrices de transformation homogènes en fonction des vecteurs (3.4), sont les suivantes :

$$Rot(z, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Trans(z, r_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.5)$$

### Matrice de transformation

La matrice de transformation qui permet d'exprimer la position et l'orientation d'un repère  $R_i$  dans le repère  $R_{i-1}$  selon la convention de Denavit-Hartenberg Modifiée est donnée par l'équation suivante [49] :

$${}^{i-1}T_i = Rot(x, \alpha_i)Trans(x, d_i)Rot(z, \theta_i)Trans(z, r_i). \quad (3.6)$$

En remplaçant les matrices des équations (3.3) et (3.5) dans l'équation (3.6), la matrice de transformation, permettant d'exprimer l'orientation du repère  $R_i$  et la position de son origine  $O_i$  dans  $R_{i-1}$ , est donnée par :

$$\begin{aligned}
 {}^{i-1}T_i &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(\alpha_i) \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) & -r_i \sin(\alpha_i) \\ \sin(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \cos(\theta_i) & \cos(\alpha_i) & r_i \cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} {}^{i-1}x_i & {}^{i-1}y_i & {}^{i-1}z_i & {}^{i-1}t_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}t_i \\ 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{3.7}$$

### Modèle géométrique direct (MGD)

L'orientation de  $R_i$  et la position de  $O_i$  dans le repère de base  $R_0$  sont obtenues par la multiplication des matrices de transformation selon l'équation (3.8) :

$${}^0T_i = {}^0T_1 {}^1T_2 \dots {}^{i-1}T_i. \tag{3.8}$$

Le MGD permet d'exprimer la position et l'orientation du repère attaché à l'organe terminal  $R_n$  dans le repère  $R_0$  :

$${}^0T_n = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n. \tag{3.9}$$

Le vecteur  ${}^0t_n$  permet d'exprimer la position de l'organe terminal, tandis que la matrice de rotation  ${}^0R_n$  permet d'exprimer sa rotation dans  $R_0$ .

### Expression de $\frac{\partial {}^0T_i}{\partial q_j}$

Pour calculer l'expression de la dérivée de  ${}^0T_i$  par rapport à la variable articulaire  $q_j$ , on distingue deux cas possibles :

1. Si  $j > i$  : dans ce cas,  ${}^0T_i$  n'est pas fonction de  $q_j$ . D'où :

$$\frac{\partial {}^0T_i}{\partial q_j} = 0_{4 \times 4}.$$

$0_{4 \times 4}$  est une matrice carrée nulle de dimension 4.

2. Si  $j \leq i$  : pour ce cas, comme la matrice  ${}^0T_i$  résulte du produit matriciel des  $i$  matrices de  ${}^0T_1$  jusqu'au  ${}^{i-1}T_i$ , il est clair que seule la matrice  ${}^{j-1}T_j$  contient  $q_j$  :

$$\begin{aligned}
 \frac{\partial {}^0T_i}{\partial q_j} &= \frac{\partial ({}^0T_1 \dots {}^{j-1}T_j \dots {}^{i-1}T_i)}{\partial q_j} \\
 &= \underbrace{\frac{\partial {}^0T_1}{\partial q_j} {}^1T_2 \dots {}^{i-1}T_i + \dots + {}^0T_1 \dots \frac{\partial {}^{j-1}T_j}{\partial q_j} \dots {}^{i-1}T_i}_{=0} \\
 &\quad + \dots + \underbrace{{}^0T_1 \dots {}^{i-2}T_{i-1} \frac{\partial {}^{i-1}T_i}{\partial q_j}}_{=0}.
 \end{aligned} \tag{3.10}$$

Il en résulte que le terme  $\frac{\partial {}^0T_i}{\partial q_j}$  se simplifie pour donner la relation suivante :

$$\begin{aligned}\frac{\partial {}^0T_i}{\partial q_j} &= {}^0T_1 \dots {}^{j-2}T_{j-1} \frac{\partial {}^{j-1}T_j}{\partial q_j} {}^jT_{j+1} \dots {}^{i-1}T_i \\ &= {}^0T_{j-1} \frac{\partial {}^{j-1}T_j}{\partial q_j} {}^jT_i.\end{aligned}\quad (3.11)$$

**Expression de  $\frac{\partial {}^{j-1}T_j}{\partial q_j}$**

La relation entre la variable articulaire  $q_j$  et le paramètre géométrique de la convention MDH  $\theta_j$  est la suivante :

$$\theta_j = q_j + q_{j_0}, \quad (3.12)$$

où  $q_{j_0}$  est l'angle (constant) entre les axes  $x_{j-1}$  et  $x_j$  par rapport à  $z_j$  quand l'angle  $q_j$  est nul. A partir de la relation (3.7), la matrice  ${}^{j-1}T_j$  peut être ré-écrite en fonction de  $q_j$  comme suit :

$${}^{j-1}T_j = \begin{bmatrix} \cos(q_j + q_{j_0}) & -\sin(q_j + q_{j_0}) & 0 & d_j \\ \cos(\alpha_j) \sin(q_j + q_{j_0}) & \cos(\alpha_j) \cos(q_j + q_{j_0}) & -\sin(\alpha_j) & -r_j \sin(\alpha_j) \\ \sin(\alpha_j) \sin(q_j + q_{j_0}) & \sin(\alpha_j) \cos(q_j + q_{j_0}) & \cos(\alpha_j) & r_j \cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.13)$$

La dérivée de ce terme par rapport à la coordonnée articulaire de position  $q_j$ ,  $\frac{\partial {}^{j-1}T_j}{\partial q_j}$ , est donnée par :

$$\frac{\partial {}^{j-1}T_j}{\partial q_j} = \begin{bmatrix} -\sin(q_j + q_{j_0}) & -\cos(q_j + q_{j_0}) & 0 & 0 \\ \cos(\alpha_j) \cos(q_j + q_{j_0}) & -\cos(\alpha_j) \sin(q_j + q_{j_0}) & 0 & 0 \\ \sin(\alpha_j) \cos(q_j + q_{j_0}) & -\sin(\alpha_j) \sin(q_j + q_{j_0}) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.14)$$

Les expressions de  $\frac{\partial {}^0T_i}{\partial q_j}$  et  $\frac{\partial {}^{j-1}T_j}{\partial q_j}$  seront utiles pour exprimer les équations du modèle dynamique.

### 3.3.2 Modèle cinématique direct

Le modèle cinématique direct d'un robot permet d'exprimer les vitesses linéaire et angulaire du repère rattaché à l'organe terminal  $R_n$  dans le repère de base  $R_0$  :

#### Vitesses linéaire et angulaire de l'origine $O_n$

Les vitesses linéaire  ${}^0V_{O_n}$  et angulaire  ${}^0W_{O_n}$  de l'organe terminal, exprimées dans le repère absolu  $R_0$ , sont données en fonction des vitesses articulaires et les composantes

des matrices de transformation [27] :

$$\begin{aligned} \begin{bmatrix} {}^0V_{O_n} \\ {}^0W_{O_n} \end{bmatrix} &= \begin{bmatrix} {}^0z_1 \times {}^0l_1 & {}^0z_2 \times {}^0l_2 & \dots & {}^0z_n \times {}^0l_n \\ {}^0z_1 & {}^0z_2 & \dots & {}^0z_n \end{bmatrix} \dot{q} \\ &= \begin{bmatrix} J_v(q) \\ J_w(q) \end{bmatrix} \dot{q} \\ &= J(q)\dot{q}, \end{aligned} \quad (3.15)$$

où :

- $q$  et  $\dot{q}$  sont les vecteurs des positions et vitesses articulaires, respectivement ;
- ${}^0l_k = {}^0t_n - {}^0t_k$ ,  $k = 1, \dots, n$  ;
- $\times$  désigne le produit vectoriel ;
- $J(q)$  : s'appelle la matrice jacobienne, et  $J_v(q)$  et  $J_w(q)$  s'appellent les matrices jacobiennes de la vitesse linéaire et angulaire, respectivement.

### Vitesses linéaire et angulaire d'une origine $O_i$

Pour une origine  $O_i$  ( $i = 1, \dots, n$ ), l'expression des vitesses est donnée comme suit :

$$\begin{aligned} \begin{bmatrix} {}^0V_{O_i} \\ {}^0W_{O_i} \end{bmatrix} &= \begin{bmatrix} {}^0z_1 \times {}^0l_1 & {}^0z_2 \times {}^0l_2 & \dots & {}^0z_i \times {}^0l_i & 0 & \dots & 0 \\ {}^0z_1 & {}^0z_2 & \dots & {}^0z_i & 0 & \dots & 0 \end{bmatrix} \dot{q} \\ &= \begin{bmatrix} J_{v_i}(q) \\ J_{w_i}(q) \end{bmatrix} \dot{q} \\ &= J_i(q)\dot{q}. \end{aligned} \quad (3.16)$$

### Expression de $\frac{\partial J_{v_i}}{\partial q_j}$

Le calcul de la dérivée  $\frac{\partial J_{v_i}}{\partial q_j}$  nécessite de dériver chaque composante de l'expression (3.16) :

$$\frac{\partial J_{v_i}}{\partial q_j} = \frac{\partial \left[ {}^0z_1 \times {}^0l_1 \quad {}^0z_2 \times {}^0l_2 \quad \dots \quad {}^0z_i \times {}^0l_i \quad 0 \quad \dots \quad 0 \right]}{\partial q_j}, \quad (3.17)$$

d'où :

$$\frac{\partial J_{v_i}}{\partial q_j} = \left[ \frac{\partial ({}^0z_1 \times {}^0l_1)}{\partial q_j} \quad \frac{\partial ({}^0z_2 \times {}^0l_2)}{\partial q_j} \quad \dots \quad \frac{\partial ({}^0z_i \times {}^0l_i)}{\partial q_j} \quad 0 \quad \dots \quad 0 \right]. \quad (3.18)$$

Pour chaque terme de (3.18) :

$$\frac{\partial ({}^0z_k \times {}^0l_k)}{\partial q_j} = \frac{\partial {}^0z_k}{\partial q_j} \times {}^0l_k + {}^0z_k \times \frac{\partial {}^0l_k}{\partial q_j}, \quad k = 1, \dots, i. \quad (3.19)$$

La dérivée de  $\frac{\partial {}^0l_k}{\partial q_j}$  est donnée par :

$$\frac{\partial {}^0l_k}{\partial q_j} = \frac{\partial {}^0t_i}{\partial q_j} - \frac{\partial {}^0t_k}{\partial q_j} = \frac{\partial {}^0T_i(1:3,4)}{\partial q_j} - \frac{\partial {}^0T_k(1:3,4)}{\partial q_j}. \quad (3.20)$$

Celle de  $\frac{\partial {}^0z_k}{\partial q_j}$  est donnée par :

$$\frac{\partial {}^0z_k}{\partial q_j} = \frac{\partial {}^0T_k(1:3,3)}{\partial q_j}. \quad (3.21)$$

**Expression de  $\frac{\partial J_{w_i}}{\partial q_j}$**

Comme pour le cas de  $\frac{\partial J_{v_i}}{\partial q_j}$ , la dérivée de  $\frac{\partial J_{w_i}}{\partial q_j}$  nécessite le calcul de la dérivée de chaque composante de  $J_{w_i}$  :

$$\frac{\partial J_{w_i}}{\partial q_j} = \frac{\partial \left[ \begin{array}{cccccc} {}^0z_1 & {}^0z_2 & \dots & {}^0z_i & 0 & \dots & 0 \end{array} \right]}{\partial q_j}, \quad (3.22)$$

d'où

$$\frac{\partial J_{w_i}}{\partial q_j} = \left[ \begin{array}{cccccc} \frac{\partial {}^0z_1}{\partial q_j} & \frac{\partial {}^0z_2}{\partial q_j} & \dots & \frac{\partial {}^0z_i}{\partial q_j} & 0 & \dots & 0 \end{array} \right]. \quad (3.23)$$

Les dérivées  $\frac{\partial {}^0z_k}{\partial q_j}$  pour  $k = 1, \dots, i$  sont données par l'équation (3.21).

Les expressions de  $\frac{\partial J_{v_i}}{\partial q_j}$  et  $\frac{\partial J_{w_i}}{\partial q_j}$  seront utiles pour exprimer les équations du modèle dynamique.

### Vitesses linéaire et angulaire du centre de gravité $g_i$

Le vecteur de position  ${}^0O_0g_i$  du centre de masse  $g_i$  du corps  $i$  dans le repère  $R_0$  peut être décomposé en la somme du vecteur de position  ${}^0O_0O_i$  de l'origine  $O_i$  du repère  $R_i$  dans  $R_0$ , avec le vecteur de position  ${}^0O_i g_i$  de  $g_i$  dans  $R_i$  exprimé dans  $R_0$ , comme suit :

$$\begin{aligned} {}^0O_0g_i &= {}^0O_0O_i + {}^0O_i g_i \\ &= {}^0O_0O_i + {}^0R_i {}^iO_i g_i, \\ &= {}^0t_i + {}^0R_i {}^i r_i, \end{aligned} \quad (3.24)$$

tel que  ${}^0R_i$  est la matrice de rotation du repère  $R_i$  dans le repère  $R_0$ . La dérivation de  ${}^0O_0g_i$  par rapport au temps  $t$  donne :

$$\frac{d {}^0O_0g_i}{dt} = \frac{d {}^0t_i}{dt} + \frac{d ({}^0R_i)}{dt} {}^i r_i + {}^0R_i \underbrace{\frac{d ({}^i r_i)}{dt}}_{=0}. \quad (3.25)$$

Comme les coordonnées de  $g_i$  sont fixes dans  $R_i$ ,  $\frac{d({}^i r_i)}{dt}$  est égal à zéro. Notons par  ${}^0V_{g_i}$  la dérivée  $\frac{d{}^0O_0g_i}{dt}$ , et par  ${}^0V_{O_i}$  la dérivée  $\frac{d{}^0t_i}{dt}$ . L'équation (3.25) devient :

$$\begin{aligned} {}^0V_{g_i} &= {}^0V_{O_i} + {}^0\dot{R}_i {}^iR_0 {}^0O_i g_i \\ &= {}^0V_{O_i} + \underbrace{{}^0\dot{R}_i ({}^0R_i)^T}_{{}^0\hat{W}_i} {}^0r_i, \end{aligned} \quad (3.26)$$

avec  $\frac{d({}^0R_i)}{dt} = {}^0\dot{R}_i$ . Dans [63], il a été démontré que  ${}^0\hat{W}_i = {}^0\dot{R}_i ({}^0R_i)^T$ , tel que  ${}^0\hat{W}_i$  est la matrice antisymétrique associée au vecteur de la vitesse angulaire  ${}^0W_i = [{}^0w_{x_i} \quad {}^0w_{y_i} \quad {}^0w_{z_i}]^T$  du repère  $R_i$  par rapport à  $R_0$  :

$${}^0\hat{W}_i = \begin{bmatrix} 0 & -{}^0w_{z_i} & {}^0w_{y_i} \\ {}^0w_{z_i} & 0 & -{}^0w_{x_i} \\ -{}^0w_{y_i} & {}^0w_{x_i} & 0 \end{bmatrix}.$$

Puisque  ${}^0\hat{W}_i {}^0r_i = {}^0W_i \times {}^0r_i$ , l'équation (3.26) devient :

$${}^0V_{g_i} = {}^0V_{O_i} + {}^0W_i \times {}^0r_i. \quad (3.27)$$

La vitesse angulaire du point  $g_i$ , ou celle de n'importe quel point du corps  $i$  est la même. Elle correspond à  ${}^0W_i$ . Ainsi, nous en déduisons les expressions des vitesses linéaire et angulaire du point  $g_i$  :

$$\begin{bmatrix} {}^0V_{g_i} \\ {}^0W_{g_i} \end{bmatrix} = \begin{bmatrix} {}^0V_{O_i} + {}^0W_{O_i} \times {}^0O_i g_i \\ {}^0W_{O_i} \end{bmatrix} = \begin{bmatrix} {}^0V_{O_i} + {}^0W_i \times {}^0r_i \\ {}^0W_i \end{bmatrix}. \quad (3.28)$$

### Ré-écriture de ${}^0V_{g_i}$

A partir de l'équation (3.27), nous allons formuler autrement l'expression de la vitesse  ${}^0V_{g_i}$ . Puisque le produit vectoriel est anticommutatif, (3.27) devient :

$${}^0V_{g_i} = {}^0V_{O_i} - {}^0r_i \times {}^0W_i. \quad (3.29)$$

En profitant de la relation entre le produit vectoriel et la matrice antisymétrique associée au premier vecteur du produit, il vient :

$${}^0V_{g_i} = {}^0V_{O_i} - {}^0\hat{r}_i {}^0W_i. \quad (3.30)$$

En remplaçant  ${}^0V_{O_i}$  et  ${}^0W_i$  par leurs expressions, données dans (3.16), il vient :

$$\begin{aligned} {}^0V_{g_i} &= J_{v_i}(q)\dot{q} - {}^0\hat{r}_i J_{w_i}(q)\dot{q} \\ &= (J_{v_i}(q) - {}^0\hat{r}_i J_{w_i}(q))\dot{q} \\ &= J_{g_i}(q)\dot{q}. \end{aligned} \quad (3.31)$$

Sachant que  ${}^0\hat{r}_i = {}^0R_i {}^i\hat{r}_i ({}^0R_i)^T$ ,  $J_{g_i}(q)$  devient :

$$J_{g_i}(q) = J_{v_i}(q) - {}^0R_i {}^i\hat{r}_i ({}^0R_i)^T J_{w_i}(q). \quad (3.32)$$

Pour finir, nous regroupons les vitesses linéaire et angulaire du centre  $g_i$  dans l'équation suivante :

$$\begin{bmatrix} {}^0V_{g_i} \\ {}^0W_{g_i} \end{bmatrix} = \begin{bmatrix} J_{g_i}(q) \\ J_{w_i}(q) \end{bmatrix} \dot{q}. \quad (3.33)$$

**Expression de  $\frac{\partial J_{g_i}}{\partial q_j}$**

La dérivation de  $J_{g_i}$  par rapport à  $q_j$  est donnée par l'expression suivante :

$$\begin{aligned} \frac{\partial J_{g_i}}{\partial q_j} &= \frac{\partial \left( J_{v_i}(q) - {}^0R_i {}^i\hat{r}_i ({}^0R_i)^T J_{w_i}(q) \right)}{\partial q_j} \\ &= \frac{\partial J_{v_i}(q)}{\partial q_j} - \frac{\partial ({}^0R_i)}{\partial q_j} {}^i\hat{r}_i ({}^0R_i)^T J_{w_i}(q) \\ &\quad - {}^0R_i {}^i\hat{r}_i \frac{\partial \left( ({}^0R_i)^T \right)}{\partial q_j} J_{w_i}(q) - {}^0R_i {}^i\hat{r}_i ({}^0R_i)^T \frac{\partial J_{w_i}(q)}{\partial q_j}. \end{aligned} \quad (3.34)$$

**Expression de  $\frac{\partial {}^0R_i}{\partial q_j}$**

L'expression de la dérivée  $\frac{\partial {}^0R_i}{\partial q_j}$  est extraite directement de  $\frac{\partial {}^0T_i}{\partial q_j}$  expliquée dans la sous-section 3.3.1.  $\frac{\partial {}^0R_i}{\partial q_j}$  est présentée par la sous-matrice formée en supprimant les dernières ligne et colonne de  $\frac{\partial {}^0T_i}{\partial q_j}$ .

### 3.3.3 Modèle dynamique

Afin de calculer le modèle dynamique des robots manipulateurs, nous allons utiliser le formalisme d'Euler-Lagrange [63]. Pour cela, il est nécessaire de commencer d'abord par le calcul des expressions des énergies cinétique et potentielle du robot.

#### Énergie cinétique

L'énergie cinétique  $K$  d'un robot manipulateur à  $n$  articulations est la somme des énergies cinétiques de translation  $K_{tr}$  et de rotation  $K_{rot}$  des  $n$  corps constituant le robot :

$$\begin{aligned} K &= K_{tr} + K_{rot} \\ &= \frac{1}{2} \sum_{i=1}^n \left( m_i {}^0V_{g_i} \bullet {}^0V_{g_i} \right) + \frac{1}{2} \sum_{i=1}^n \left( {}^0W_i \bullet \left( {}^0R_i {}^iI_{g_i} {}^0R_i^T \right) {}^0W_i \right), \end{aligned} \quad (3.35)$$



où  $m_i$  est la masse du  $i^{eme}$  corps,  ${}^i I_{g_i}$  est la matrice du tenseur d'inertie autour du centre de masse  $g_i$ , et le symbole  $\bullet$  désigne le produit scalaire. En remplaçant l'équation (3.33) dans (3.35), il vient :

$$\begin{aligned} K &= \frac{1}{2} \sum_{i=1}^n \left( m_i \dot{q}^T J_{g_i}^T J_{g_i} \dot{q} \right) + \frac{1}{2} \sum_{i=1}^n \left( \dot{q}^T J_{w_i}^T {}^0 R_i {}^i I_{g_i} {}^0 R_i^T J_{w_i} \dot{q} \right) \\ &= \frac{1}{2} \dot{q}^T \left[ \sum_{i=1}^n \left( m_i J_{g_i}^T J_{g_i} + J_{w_i}^T {}^0 R_i {}^i I_{g_i} {}^0 R_i^T J_{w_i} \right) \right] \dot{q}. \end{aligned} \quad (3.36)$$

Sous une forme compacte, l'énergie cinétique est donnée par l'équation suivante :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q}, \quad (3.37)$$

où  $M(q)$  est appelée la matrice d'inertie. Elle est donnée par l'expression suivante :

$$M(q) = \sum_{i=1}^n \left( m_i J_{g_i}^T J_{g_i} + J_{w_i}^T {}^0 R_i {}^i I_{g_i} {}^0 R_i^T J_{w_i} \right). \quad (3.38)$$

Les matrices  $J_{g_i}^T J_{g_i}$  et  $J_{w_i}^T {}^0 R_i {}^i I_{g_i} {}^0 R_i^T J_{w_i}$  sont des matrices symétriques. Par conséquent, la matrice d'inertie  $M(q)$  est symétrique aussi.

L'énergie cinétique peut être aussi exprimée en fonction du tenseur d'inertie  ${}^i I_i$  autour de l'origine  $O_i$ . La relation entre  ${}^i I_i$  et  ${}^i I_{g_i}$  est donnée par l'équation suivante [81] :

$$\begin{aligned} {}^i I_i &= {}^i I_{g_i} + m_i \left( {}^i r_i^T {}^i r_i E_3 - {}^i r_i {}^i r_i^T \right), \\ &= {}^i I_{g_i} - m_i \left( {}^i \hat{r}_i \right)^2. \end{aligned} \quad (3.39)$$

Afin d'avoir des expressions moins couteuses en calcul, nous nous intéressons dans cette section, à l'écriture du modèle dynamique du robot en fonction des  ${}^i I_{g_i}$ , comme c'est proposé dans [63]. Signalons par ailleurs que d'autres références font appel à un modèle exprimé en fonction des  ${}^i I_i$ , et non pas des  ${}^i I_{g_i}$  [27].

## Énergie potentielle

Comme l'énergie cinétique, l'énergie potentielle du robot est la somme des énergies potentielles des  $n$  corps qui le constituent :

$$P = \sum_{i=1}^n \left( m_i V_g \bullet {}^0 O_0 g_i \right), \quad (3.40)$$

où  $V_g$  est le vecteur de gravité exprimé dans le repère  $R_0$ . D'après l'équation (3.24) :

$$\begin{aligned} {}^0 O_0 g_i &= {}^0 O_0 O_i + {}^0 R_i {}^i O_i g_i \\ &= {}^0 t_i + {}^0 R_i {}^i r_i. \end{aligned}$$

Ainsi, l'équation (3.40) devient :

$$P = \sum_{i=1}^n \left( m_i V_g^T \left[ {}^0t_i + {}^0R_i {}^i r_i \right] \right). \quad (3.41)$$

### Équations d'Euler-Lagrange

Le lagrangien  $L$  est défini comme la différence entre l'énergie cinétique et potentielle :

$$L = K - P \quad (3.42)$$

Le modèle dynamique du robot est donné par les équations suivantes, dites d'Euler-Lagrange :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = \tau_{m_j} + \tau_{f_j} + \tau_{ext_j}, \quad j = 1, \dots, n, \quad (3.43)$$

où,  $\tau_{m_j}$ ,  $\tau_{f_j}$  et  $\tau_{ext_j}$  correspondent aux couples d'entraînement du motoréducteur d'un corps sur l'autre, aux couples articulaires dus au frottement et ceux résultant d'une force extérieure  $F_{ext}$ , respectivement. Un modèle typique de frottement, tel que dans [27], peut être décrit par la somme des frottements visqueux, caractérisés par un coefficient  $F_{v_j}$ , et des frottements secs, caractérisés par le coefficient  $F_{s_j}$ , de sorte que le couple articulaire dû au frottement pour la liaison  $j$  est donné par :

$$\tau_{f_j} = -F_{s_j} \text{sign}(\dot{q}_j) - F_{v_j} \dot{q}_j. \quad (3.44)$$

Le couple  $\tau_{ext_j}$  résulte de l'action d'une force extérieure  $F_{ext}$  appliquée sur l'organe terminal du robot. A partir de cette force  $F_{ext}$ , le couple résultant de cette force, noté  $\tau_{ext_j}$ , au niveau de l'articulation  $j$  [63] est donné par :

$$\tau_{ext_j} = \left( J_{v_{(:,j)}} \right)^T (q) F_{ext}. \quad (3.45)$$

$J_{v_{(:,j)}}$  correspond au vecteur contenant toutes les composantes de la  $j^{\text{ème}}$  colonne de la matrice  $J_v$ .

Dans ce qui suit, nous allons procéder au calcul de l'expression du premier terme de l'équation (3.43). Nous commençons d'abord par les expressions de la dérivée partielle du lagrangien par rapport à  $\dot{q}_j$ . Celles-ci sont données par les équations suivantes :

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial \dot{q}_j} = \frac{\partial K}{\partial \dot{q}_j} - \frac{\partial P}{\partial \dot{q}_j}, \\ \\ \frac{\partial K}{\partial \dot{q}_j} = \frac{1}{2} \left[ \begin{array}{cccc} 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 0 \end{array} \right] M(q) \dot{q} + \frac{1}{2} \dot{q}^T M(q) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ \\ = \frac{1}{2} M_{(j,:)}(q) \dot{q} + \frac{1}{2} \dot{q}^T M_{(:,j)}(q) \\ \\ = M_{(j,:)}(q) \dot{q}, \quad (\text{M est symétrique + commutativité du produit scalaire}) \\ \\ \frac{\partial P}{\partial \dot{q}_j} = 0. \end{array} \right. \quad (3.46)$$

Les seules composantes non nulles, puisqu'égalées à 1, dans les vecteurs ligne et colonne de l'expression de  $\frac{\partial K}{\partial \dot{q}_j}$  correspondent à la  $j^{\text{ème}}$  composante du vecteur  $\dot{q}$ .  $M_{(j,:)}(q)$  et  $M_{(:,j)}(q)$  correspondent aux vecteurs contenant toutes les composantes de la  $j^{\text{ème}}$  ligne et la  $j^{\text{ème}}$  colonne de la matrice  $M(q)$ , respectivement.  $\frac{\partial P}{\partial \dot{q}_j}$  est nulle car l'énergie potentielle n'est pas fonction des vitesses articulaires. Ensuite, nous calculons la dérivée  $\frac{\partial L}{\partial \dot{q}_j}$  par rapport au temps. Il vient :

$$\left\{ \begin{array}{l} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) = \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_j} \right) - \frac{d}{dt} \left( \frac{\partial P}{\partial \dot{q}_j} \right), \\ \\ \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_j} \right) = M_{(j,:)}(q) \ddot{q} + \dot{M}_{(j,:)}(q) \dot{q}, \\ \\ \frac{d}{dt} \left( \frac{\partial P}{\partial \dot{q}_j} \right) = 0. \end{array} \right. \quad (3.47)$$

où  $\ddot{q}$  est le vecteur des accélérations articulaires. La dérivation du lagrangien par rapport à  $q_j$  donne les expressions suivantes :

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial q_j} = \frac{\partial K}{\partial q_j} - \frac{\partial P}{\partial q_j}, \\ \frac{\partial K}{\partial q_j} = \frac{1}{2} \dot{q}^T \frac{\partial M(q)}{\partial q_j} \dot{q}, \\ \frac{\partial P}{\partial q_j} = \sum_{i=1}^n \left( m_i |V_g|^T \left[ \frac{\partial {}^0 t_i}{\partial q_j} + \frac{\partial {}^0 R_i}{\partial q_j} {}^i r_i \right] \right) \\ = G_j(q). \end{array} \right. \quad (3.48)$$

### Modèle dynamique

A travers les équations (3.47) et (3.48), l'équation d'Euler-Lagrange pour la  $j^{eme}$  articulation est donnée par :

$$M_{(j,:)}(q)\ddot{q} + \dot{M}_{(j,:)}(q)\dot{q} - \frac{1}{2} \dot{q}^T \frac{\partial M(q)}{\partial q_j} \dot{q} + G_j(q) = \tau_{m_j} + \tau_{f_j} + \tau_{ext_j}, \quad (3.49)$$

d'où :

$$M_{(j,:)}(q)\ddot{q} + \left( \dot{M}_{(j,:)}(q) - \frac{1}{2} \dot{q}^T \frac{\partial M(q)}{\partial q_j} \right) \dot{q} + G_j(q) = \tau_{m_j} + \tau_{f_j} + \tau_{ext_j}. \quad (3.50)$$

Pour toutes les articulations du robot de 1 à  $n$ , on peut présenter l'équation (3.50) sous sa forme vectorielle comme suit :

$$\begin{bmatrix} \vdots \\ M_{(j,:)}(q) \\ \vdots \end{bmatrix} \ddot{q} + \begin{bmatrix} \vdots \\ \dot{M}_{(j,:)}(q) - \frac{1}{2} \dot{q}^T \frac{\partial M(q)}{\partial q_j} \\ \vdots \end{bmatrix} \dot{q} + \begin{bmatrix} \vdots \\ G_j(q) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \tau_{m_j} + \tau_{f_j} + \tau_{ext_j} \\ \vdots \end{bmatrix}, \quad (3.51)$$

ce qui conduit à :

$$M(q)\ddot{q} + \left( \dot{M}(q) - \frac{1}{2} \begin{bmatrix} \vdots \\ \dot{q}^T \frac{\partial M(q)}{\partial q_j} \\ \vdots \end{bmatrix} \right) \dot{q} + G(q) = \tau + \tau_f + \tau_{ext}. \quad (3.52)$$

Le modèle dynamique du robot, avec une écriture plus compacte, est donné par l'équation suivante :

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = \tau + \tau_f + J_v^T(q)F_{ext}, \quad (3.53)$$

où  $N(q, \dot{q})\dot{q} = \left( \begin{array}{c} \dot{M}(q) - \frac{1}{2} \begin{bmatrix} \vdots \\ \dot{q}^T \frac{\partial M(q)}{\partial q_j} \\ \vdots \end{bmatrix} \end{array} \right) \dot{q}$  est appelé le vecteur des couples de Coriolis et centrifuge, et  $G(q)$  le vecteur des couples de gravité [27]. La matrice  $\dot{M}(q)$  peut être exprimée comme suit :

$$\dot{M}(q) = \frac{dM(q)}{dt} = \sum_{j=1}^n \frac{\partial M(q)}{\partial q_j} \dot{q}_j. \quad (3.54)$$

**Expression de  $\frac{\partial M(q)}{\partial q_j}$**

L'expression de la dérivée de  $M(q)$  par rapport à  $q_j$  est donnée par :

$$\frac{\partial M(q)}{\partial q_j} = \sum_{i=1}^n \left( m_i \frac{\partial (J_{g_i}^T J_{g_i})}{\partial q_j} + \frac{\partial (J_{w_i}^T {}^0R_i {}^iI_{g_i} {}^0R_i^T J_{w_i})}{\partial q_j} \right), \quad (3.55)$$

où :

$$\frac{\partial (J_{g_i}^T J_{g_i})}{\partial q_j} = \frac{\partial (J_{g_i}^T)}{\partial q_j} J_{g_i} + J_{g_i}^T \frac{\partial (J_{g_i})}{\partial q_j}, \quad (3.56)$$

et :

$$\begin{aligned} \frac{\partial (J_{w_i}^T {}^0R_i {}^iI_{g_i} {}^0R_i^T J_{w_i})}{\partial q_j} &= \frac{\partial (J_{w_i}^T)}{\partial q_j} {}^0R_i {}^iI_{g_i} {}^0R_i^T J_{w_i} + J_{w_i}^T \frac{\partial {}^0R_i {}^iI_{g_i} {}^0R_i^T J_{w_i}}{\partial q_j} \\ &\quad + J_{w_i}^T {}^0R_i {}^iI_{g_i} \frac{\partial ({}^0R_i^T)}{\partial q_j} J_{w_i} + J_{w_i}^T {}^0R_i {}^iI_{g_i} {}^0R_i^T \frac{\partial J_{w_i}}{\partial q_j}. \end{aligned} \quad (3.57)$$

Dans cette section, nous avons élaboré les équations décrivant dans l'espace articulaire, la dynamique d'un robot manipulateur série à articulations rotoïdes. Le modèle dynamique du robot est nécessaire pour développer des schémas de commande, simuler le comportement du robot, et aussi pour l'identification des paramètres. Les expressions des différents blocs du modèle dynamique nous ont permis de conclure sa dépendance linéaire aux masses  $m_i$ , tenseurs d'inertie  ${}^iI_{g_i}$ , coordonnées des centres de masse  $r_i$  et aux coefficients de frottement  $F_{s_i}, F_{v_i}$ . Cette propriété nous a aidé à simplifier et à réduire les expressions développées. Elle sera aussi exploitée pour l'identification des paramètres dynamiques du robot.

## 3.4 Modélisation du robot Denso VP-6242G

### 3.4.1 Description du robot Denso VP-6242G et de son module de contrôle

Le robot Denso VP-6242G est un robot conçu et fabriqué par la firme japonaise *Denso*. Il est destiné à des tâches d'assemblage et de manutention nécessitant un espace réduit et une capacité relativement faible, dans différents secteurs comme celui de l'électronique, l'agro-alimentaire, l'industrie pharmaceutique, la plasturgie [2] *etc.* Ce robot possède six articulations rotoïdes. Il pèse environ 15 kg et sa charge utile est de 2 kg. La figure 3.7a présente le robot Denso VP-6242G dont nous disposons au laboratoire.



(a) Denso VP-6242G.



(b) Capteur de force + outil.

FIGURE 3.7 – Le robot Denso VP-6242G équipé d'un capteur de force et d'une outil.

Comme l'indique les deux photos de la figure 3.7, le robot est équipé d'un capteur de force de type **Gamma** distribué par la société *ATI Industrial Automation*. Le capteur est fixé sur le corps de la dernière articulation du robot (figure 3.8a). Un outil sphérique est installé sur le capteur de force (figure 3.8b). La masse totale du capteur avec l'outil est  $m_t = 0.94$  kg. Cette installation robotique doit permettre d'utiliser ce robot pour des essais expérimentaux de co-manipulation homme-robot. Le but de ces essais expérimentaux est de faire la démonstration de la faisabilité de notre approche pour la co-manipulation homme-robot en vue de la manutention de charges lourdes.

Au laboratoire, le robot *Denso* est disponible avec un module de contrôle à architecture ouverte de la société *Quanser* (figure 3.8c). Le système robot + module de contrôle

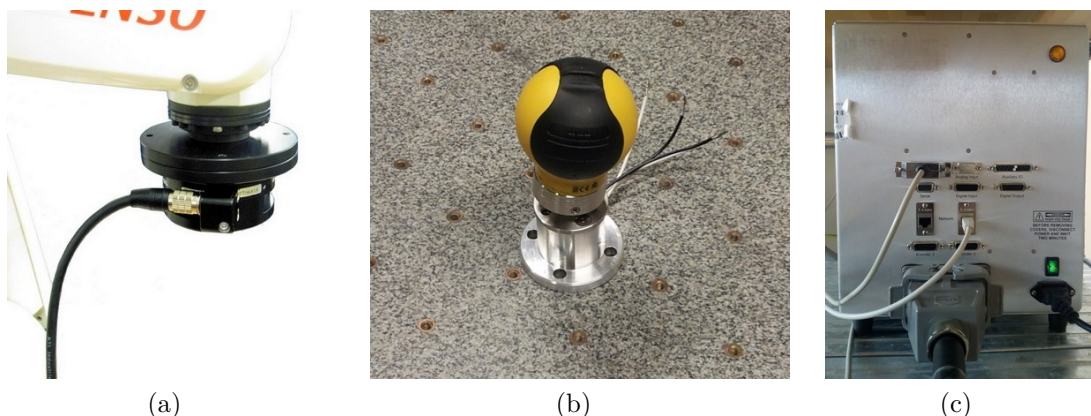
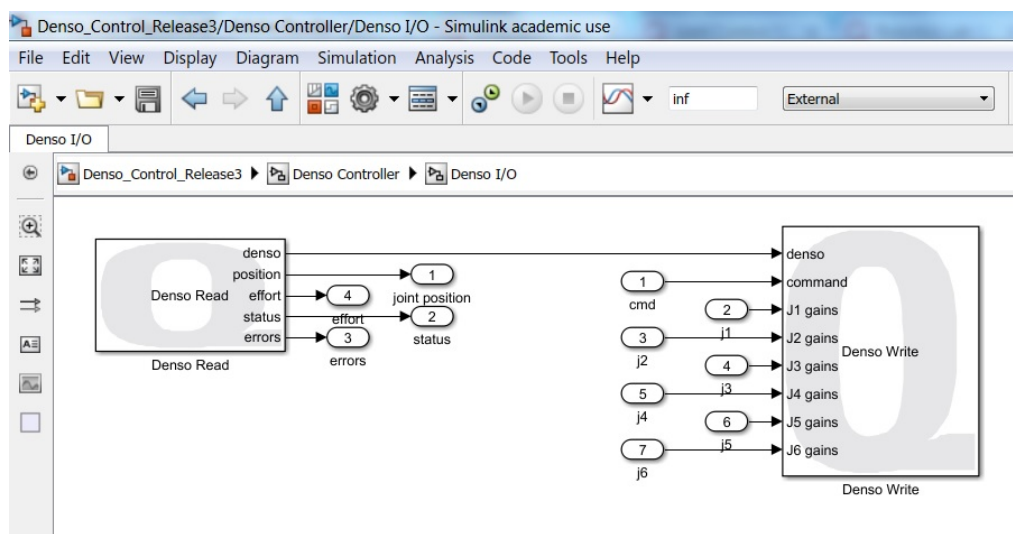


FIGURE 3.8 – Le capteur de force (a), l'outil installé (b) et le module de contrôle (c).

permet d'avoir les capacités d'un système industriel interfacé avec le logiciel de contrôle en temps réel *Quarc*, compatible avec *Matlab/Simulink*. Le bras robotique contient six encodeurs qui permettent de mesurer les positions angulaires des six arbres de chaque axe (moto-réducteur). Les longueurs des corps constituant le robot sont indiquées dans la figure 3.10. Pour plus de détails techniques sur ce robot, son module de contrôle et son l'interface, se référer aux sites web [2], [5] et [6]. La communication avec le robot se fait via l'interface *Quarc* dans *Simulink* à travers les blocs *Denso Read* et *Denso Write* (figure 3.9).

FIGURE 3.9 – Les blocs *Denso Read* et *Write*.

Ces blocs fonctionnent à une fréquence d'échantillonnage de  $1\text{ ms}$ . Le bloc *Denso Read* permet de communiquer les positions des encodeurs articulaires, qui sont, après calibration, transformés en degrés. Le bloc *Denso Read* permet aussi de récupérer la mesure du courant électrique consommé pour chaque moteur, des codes informant sur l'état du robot et de son contrôleur (déconnecté, erreur, normal,...) et un code d'erreur pour chaque axe. Le bloc *Denso Write* est utilisé pour envoyer des signaux de commande en position/vitesse ou en couple au robot. Il est possible de commander le robot à travers deux modes : en position/vitesse (entrée 1 du bloc *Denso Write*) ou en

couple (entrées 2 à 7). Pour le premier mode, si la position désirée est constante, donc il s'agit d'une commande en position. Si les positions articulaires désirées évoluent au cours du temps, le signal de commande est la variation de la position désirée. Comme cette variation est appliquée à chaque pas de temps (1  $ms$ ), elle est équivalente à une commande en vitesse. Pour commander le robot en couple, il faut lui appliquer le signal de couple désiré pour chaque axe, qui sera par la suite converti en courant électrique désiré.

### 3.4.2 Modélisation du robot

Afin de calculer le modèle dynamique du robot Denso VP-6242G, nous allons présenter d'abord dans la figure 3.10, les différents repères attachés à chaque corps du robot. Les repères sont définis selon la convention DHM décrite précédemment.

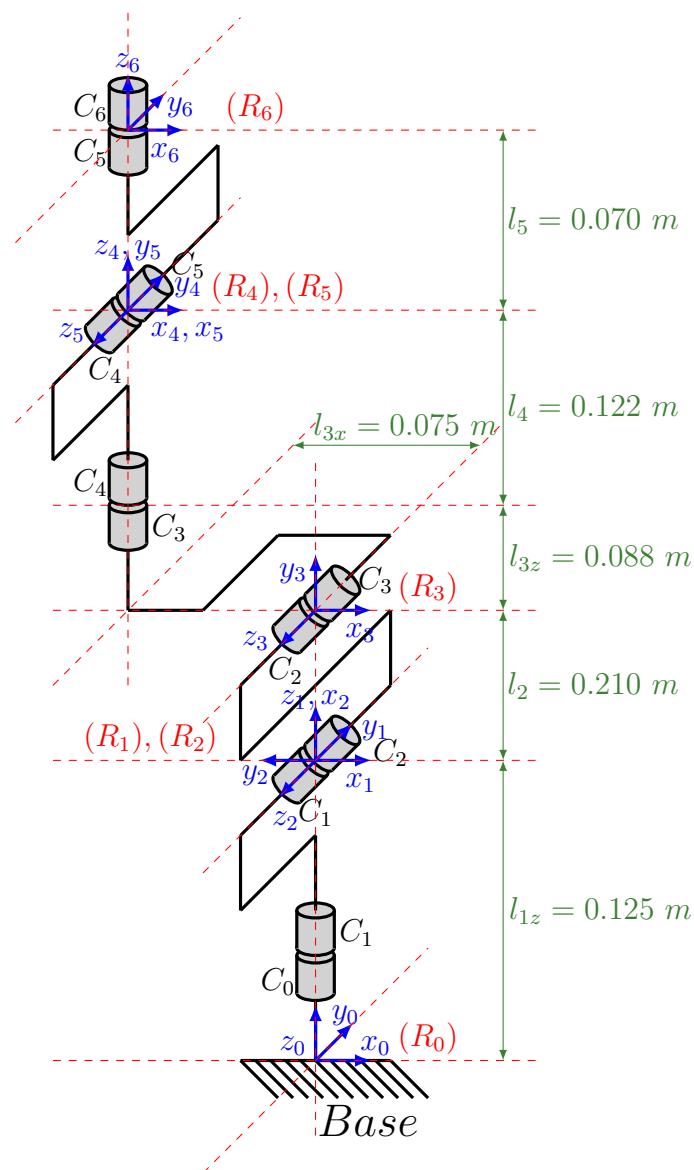


FIGURE 3.10 – Repères attachés au robot Denso VP-6242G selon la convention DHM. Le tableau 3.1 regroupe les différents paramètres (géométriques) de la convention DHM du robot Denso VP-6242G, extraits à partir de la figure 3.10.



TABLE 3.1 – Paramètres de la convention DHM correspondant au robot Denso VP-6242G.

$i$	$\alpha_i$	$d_i$	$r_i$	$\theta_i$
1	$\widehat{z_0, z_1}/x_0 = 0$	$\overline{O_0, z_1}/x_0 = 0$	$\overline{x_0, O_1}/z_1 = l_{1z}$	$\widehat{x_0, x_1}/z_1 = q_1$
2	$\widehat{z_1, z_2}/x_1 = \frac{\pi}{2}$	$\overline{O_1, z_2}/x_1 = 0$	$\overline{x_1, O_2}/z_2 = 0$	$\widehat{x_1, x_2}/z_2 = q_2 + \frac{\pi}{2}$
3	$\widehat{z_2, z_3}/x_2 = 0$	$\overline{O_2, z_3}/x_2 = l_2$	$\overline{x_2, O_3}/z_3 = 0$	$\widehat{x_2, x_3}/z_3 = q_3 - \frac{\pi}{2}$
4	$\widehat{z_3, z_4}/x_3 = -\frac{\pi}{2}$	$\overline{O_3, z_4}/x_3 = -l_{3x}$	$\overline{x_3, O_4}/z_4 = l_{3z} + l_4$	$\widehat{x_3, x_4}/z_4 = q_4$
5	$\widehat{z_4, z_5}/x_4 = \frac{\pi}{2}$	$\overline{O_4, z_5}/x_4 = 0$	$\overline{x_4, O_5}/z_5 = 0$	$\widehat{x_4, x_5}/z_5 = q_5$
6	$\widehat{z_5, z_6}/x_5 = -\frac{\pi}{2}$	$\overline{O_5, z_6}/x_5 = 0$	$\overline{x_5, O_6}/z_6 = l_5$	$\widehat{x_5, x_6}/z_6 = q_6$

Les paramètres du tableau 3.1 ont été utilisés avec le logiciel OpenSYMORO [50, 47] pour modéliser le robot Denso VP-6242G. OpenSYMORO est un logiciel publiquement disponible destiné à la modélisation symbolique des robots manipulateurs à articulations rigides ou flexibles, de type série (chaîne cinématique ouverte ou arborescente), parallèles ou mobiles à roues. Le logiciel génère les modèles géométriques, cinématiques et dynamiques, directs et inverses. Il est à noter que OpenSYMORO utilise les tenseurs d'inertie  ${}^i I_i$  pour calculer le modèle dynamique (et pas  ${}^i I_{g_i}$ ). Le logiciel permet aussi de générer le modèle dynamique pour l'identification et les paramètres dynamiques de base (voir sous section 3.5.2). Sa puissance de calcul nous a permis de réduire le temps consacré à la partie modélisation. Les programmes implémentés pour générer les différents modèles et l'interface graphique sont développés sous le langage de programmation Python [4]. L'interface graphique (figure 3.11) était développée avec la bibliothèque wxPython [7].

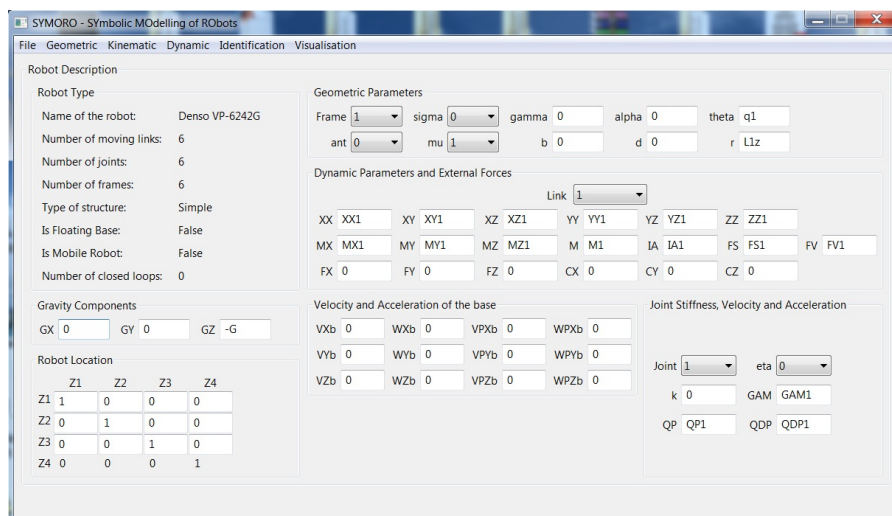


FIGURE 3.11 – Page principale du logiciel OpenSYMORO.

L'interface permet de saisir les paramètres nécessaires pour définir le robot à modéliser, notamment :

- le nombre d'articulation ;
- le type du robot : série ouvert, série arborescent, parallèle ;
- l'emplacement et la direction du repère de base (galiléen) ;
- les composantes du vecteur de gravité ;

et pour chaque articulation :

- \* les paramètres géométriques :  $\alpha_i$ ,  $d_i$ ,  $r_i$  et  $\theta_i$  ;
- \* les types d'articulation : rotoïdes ou prismatiques, rigides ou flexibles ;
- \* les paramètres dynamiques (détaillés dans la sous-section suivante 3.5.1) ;
- \* et les éventuelles forces/couples appliqués par chaque corps du robot sur l'environnement.

La version publiée dans [50] du logiciel contient aussi un outil de visualisation pour voir la structure du robot. La figure 3.12 illustre les différentes fonctions dans OpenSYMORO.

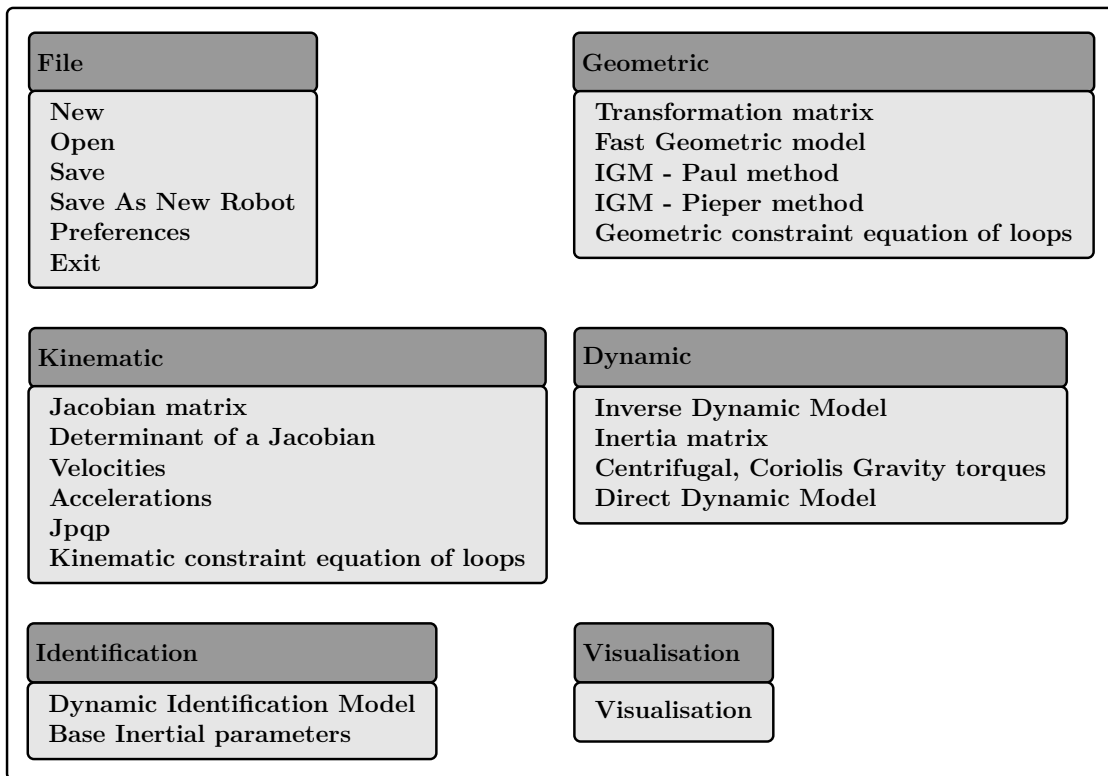


FIGURE 3.12 – Options de menu disponibles dans OpenSYMORO.

## 3.5 Identification des paramètres dynamiques du robot

Dans le domaine de la commande des systèmes, la plupart des commandes développées nécessitent la connaissance des valeurs numériques des paramètres qui décrivent la dynamique du système. L'identification est une technique qui permet de trouver une estimation des valeurs de ces paramètres. En parallèle de son importance, l'identification n'est pas une tâche facile, étant donné les problèmes que nous avons rencontrés durant cette étape. Dans cette section, nous allons présenter en quoi consiste l'identification des paramètres dynamiques pour des robots manipulateurs et comment s'y prendre.

### 3.5.1 Modèle dynamique pour l'identification

Le modèle dynamique destiné à l'identification est une ré-écriture de l'équation (3.53), sous la forme d'une relation linéaire en les paramètres à identifier, regroupés dans un vecteur noté  $\theta$  de dimension  $(n_\theta \times 1)$  [17] :

$$\tau = D(q, \dot{q}, \ddot{q})\theta \quad (3.58)$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \cdot \\ \cdot \\ \cdot \\ \tau_n \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & \cdot & \cdot & \cdot & D_{1n} \\ 0 & D_{22} & \cdot & \cdot & \cdot & D_{2n} \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 & D_{nn} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_n \end{bmatrix}, \quad (3.59)$$

où  $D$  est une matrice de dimension  $(n \times n_\theta)$  qui ne dépend que des paramètres géométriques du robot et des coordonnées articulaires  $q$ ,  $\dot{q}$  et  $\ddot{q}$ . Cette matrice est appelée *matrice d'observation* [17]. En utilisant la même notation que celle dans [47] pour une articulation  $i$ , nous pouvons définir les *paramètres dynamiques* suivants :

- $xx_i, xy_i, xz_i, yy_i, yz_i, zz_i$  : ce sont les composantes de la matrice du tenseur d'inertie  ${}^i I_i$  (symétrique) autour de l'origine  $O_i$  du repère  $R_i$  :

$${}^i I_i = \begin{bmatrix} xx_i & xy_i & xz_i \\ xy_i & yy_i & yz_i \\ xz_i & yz_i & zz_i \end{bmatrix}.$$

$xx_i, yy_i, zz_i$  s'appellent les moments d'inertie par rapport aux axes  $Ox_i, Oy_i, Oz_i$ , respectivement.  $xx_i, yy_i, zz_i$  sont toujours positifs.  $xy_i, xz_i$  et  $yz_i$  s'appellent les produits d'inertie ;

- $mx_i, my_i, mz_i$  : ce sont les moments autour des trois axes de  $R_i$  ;
- $m_i$  est la masse du corps rattaché à l'articulation ;
- $Fv_i, Fs_i$  sont les coefficients des frottements visqueux et de Coulomb respectivement, pour un modèle de frottement linéaire en ces paramètres.

Ces paramètres sont alors regroupés dans le vecteur suivant :

$$\theta_i := \left[ xx_i \quad xy_i \quad xz_i \quad yy_i \quad yz_i \quad zz_i \quad mx_i \quad my_i \quad mz_i \quad m_i \quad Fs_i \quad Fv_i \right]^T, \quad (3.60)$$

où  $T$  indique le transposé. Le vecteur  $\theta$  est construit par la concaténation verticale des vecteurs  $\theta_i$  ( $i = 1, \dots, n$ ). Les dix premiers paramètres sont appelés *paramètres inertiels* de l'articulation  $i$ , comme indiqué dans [81].

### 3.5.2 Vecteur des paramètres inertiels minimaux

Dans le cadre de l'identification des paramètres d'un robot, nous pouvons nous retrouver confrontés à l'un des deux cas suivants :

- Certains paramètres inertiels, pour certaines configurations n'interviennent pas dans la dynamique du robot. Du coup, ces paramètres ne sont pas identifiables et peuvent être éliminés du vecteur des paramètres à identifier ;
- Comme la relation entre le vecteur des paramètres et la matrice d'observation est linéaire, certains paramètres inertiels peuvent avoir les mêmes coefficients, donc pondérés par des expressions identiques dans la matrice d'observation. Il est alors indispensable de regrouper ces paramètres.

La matrice d'observation obtenue dans les deux cas n'est pas de rang plein, ce qui ne permet pas de déterminer les valeurs des paramètres d'une manière exacte. Afin de pouvoir traiter ce problème, il est nécessaire de définir un vecteur des paramètres minimaux à identifier. Pour cela, nous éliminons les paramètres qui n'affectent pas la dynamique du robot et nous regroupons les paramètres qui sont pondérés des mêmes coefficients. Les travaux [33, 34, 36, 37] se sont intéressés à :

- la détermination des paramètres qui n'ont pas d'influence sur la dynamique du robot ;
- les conditions de regroupement des paramètres inertiels ;
- les relations qui permettent le regroupement des paramètres inertiels (pour les cas des articulations prismatiques et rotoïdes).

Les paramètres inertiels regroupés sont appelés *paramètres inertiels minimaux*, et le nouveau vecteur des paramètres est appelé *vecteur des paramètres de base* qui a  $n_{\theta_B}$  composantes. La relation (3.58) peut être ré-écrite comme suit :

$$\tau = D_B(q, \dot{q}, \ddot{q})\theta_B, \quad (3.61)$$

où  $D_B$  est une matrice de rang plein. Sa dimension est  $(n \times n_{\theta_B})$ , avec  $n_{\theta_B} < n_{\theta}$ .

### 3.5.3 Signaux d'excitation pour l'identification

Dans le but de conserver l'intégrité du robot, il est recommandé de maîtriser la configuration du robot durant son mouvement, afin d'éviter les auto-collisions et les collisions du robot avec son environnement. Pour cela, nous concevons une commande en position articulaire avec des gains proportionnels ajustables. Cette commande

permet de suivre les trajectoires générées qui satisfont les exigences désirées. En identification, le choix du signal d'excitation est important, car un signal d'excitation doit permettre d'identifier tous les paramètres. Un signal possédant cette caractéristique est appelé *signal à excitation persistante*. Un signal à excitation persistante utilisé pour identifier  $n_{\theta_B}$  paramètres doit contenir au moins  $n_{\theta_B}/2$  sinusoïdes de fréquences différentes [35, 38, 61]. En d'autre terme, un signal sinusoïdal monochromatique est un signal d'excitation persistante d'ordre 2. De plus, la trajectoire désirée doit être conçue pour être périodique avec un nombre de périodes supérieur au nombre de degrés de liberté du robot, afin de faciliter l'analyse statistique des données à acquérir (calcul de la matrice de covariance du vecteur des couples et de son inverse) [75]. Aussi, les amplitudes des signaux doivent être incluses dans la plage de mouvement du robot afin de préserver son intégrité physique. Si les gains proportionnels sont suffisamment élevés, les erreurs de suivi entre les positions articulaires mesurées et celles désirées sont proches de zéro. Ce qui fait que les positions articulaires mesurées ont des propriétés similaires aux trajectoires périodiques désirées. Ainsi, chaque période mesurée peut être considérée comme ayant fait l'objet d'un signal d'excitation persistante en entrée.

### 3.5.4 Traitement des signaux mesurés

En général, les couples et positions articulaires mesurés, et les vitesses et accélérations articulaires calculées sont bruités. Ces données ne peuvent pas être utilisées directement dans le processus d'identification en raison de la boucle de commande qui pourrait créer des corrélations entre les bruits et les signaux du système en boucle fermée. Ceci peut fortement influencer les valeurs des paramètres à identifier [24, 23]. Afin d'éliminer l'effet indésirable des bruits, il est nécessaire de filtrer les données avant les utiliser dans la procédure de l'estimation. Ceci est fait hors ligne grâce à un filtre passe-bas (de type **Butterworth** par exemple), avec un gain unitaire aux basses fréquences, et une fréquence de coupure fixée une décade après la fréquence la plus élevée existante dans les trajectoires désirées. En pratique, le signal résultant d'un filtrage vers l'avant, des données bruitées est un signal déphasé (retardé) par rapport au celui mesuré. De même pour un filtrage vers l'arrière, qui donne un signal filtré en avance par rapport au signal mesuré. Par conséquent, un filtrage vers l'avant plus vers l'arrière (aller-retour) des données bruitées est effectué afin d'obtenir un filtrage à déphasage nulle de ces données. En outre, puisque les signaux d'excitation appliqués contiennent de nombreuses périodes, une période moyenne de chaque signal est calculée sur toutes les périodes disponibles afin de réduire la charge de calcul de l'algorithme d'estimation. Étant donné un signal périodique en temps discret, noté  $x(k)$ , avec  $P$  périodes et  $K$  échantillons par période. La valeur moyenne de  $x(k)$ , notée  $\bar{x}(k)$ , est obtenue à travers la relation suivante :

$$\bar{x}(k) = \frac{1}{P} \sum_{p=1}^{p=P} x(k + K(p - 1)). \quad (3.62)$$

Enfin, il est parfois utile de sous-échantillonner les données si la fréquence d'échantillonnage est trop élevée par rapport à la bande passante d'intérêt.

Soient  $\hat{q}$  et  $\hat{\tau}$  les positions et couples articulaires après avoir appliqué l'ensemble du traitement de signal décrit avant,  $\dot{\hat{q}}$  et  $\ddot{\hat{q}}$  sont les vitesses et accélérations articulaires résultantes, calculées convenablement par dérivation numérique de  $\hat{q}$ . A l'instant d'échantillonnage  $k$ , les données sont supposées vérifier la relation suivante, dérivée de (3.61) :

$$\hat{\tau}(k) = D_B \left( \hat{q}(k), \dot{\hat{q}}(k), \ddot{\hat{q}}(k) \right) \theta_B + \rho(k) \quad (3.63)$$

où  $\rho$  est le vecteur ( $n \times 1$ ) des erreurs, regroupant les bruits et les erreurs de modélisation (dynamiques non modélisées). L'ensemble des  $N$  échantillons retenus après le traitement du signal complet décrit ci-dessus sont utilisés pour définir :

$$Y := \begin{bmatrix} \hat{\tau}(1) \\ \vdots \\ \hat{\tau}(N) \end{bmatrix}, W := \begin{bmatrix} D_B \left( \hat{q}(1), \dot{\hat{q}}(1), \ddot{\hat{q}}(1) \right) \\ \vdots \\ D_B \left( \hat{q}(N), \dot{\hat{q}}(N), \ddot{\hat{q}}(N) \right) \end{bmatrix}, R := \begin{bmatrix} \rho(1) \\ \vdots \\ \rho(N) \end{bmatrix}, \quad (3.64)$$

ou  $Y$  et  $R$  sont des vecteurs de dimensions ( $n.N \times 1$ ), et  $W$  est une matrice de dimension ( $n.N \times n_{\theta_B}$ ). Ensuite, l'équation (3.63) conduit à :

$$Y = W \theta_B + R. \quad (3.65)$$

L'équation (3.65) présente un système surdéterminé d'équations linéaires en les paramètres inconnus  $\theta_B$ .

Le conditionnement associé à la matrice  $W$  permet de mesurer la sensibilité de la solution  $\theta_B$ , aux bruits affectant les mesures de  $\tau$ ,  $q$  et éventuellement  $\dot{q}$  et  $\ddot{q}$ . Par conséquent, les données post-traitées devraient réduire autant que possible le nombre de conditionnement, pour assurer la convergence du processus d'identification vers un vecteur de paramètres dynamiques non biaisés [38, 41].

### 3.5.5 Estimation du vecteur des paramètres de base

Pour une matrice d'observation  $W$  de rang plein, et un bruit gaussien additif, indépendant et de moyenne nulle pour  $R$ , une estimation  $\hat{\theta}_B$  de  $\theta_B$  peut être obtenue par la résolution du problème d'optimisation pondéré suivant [75] :

$$\hat{\theta}_B = \arg \min_{\theta_B} \| W^T \Sigma^{-1/2} (Y - W \theta_B) \|^2, \quad (3.66)$$

dont la solution optimale est obtenue par la méthode des Moindres Carrés Pondérés (MCP) :

$$\hat{\theta}_B = C W^T \Sigma^{-1/2} Y, \quad (3.67)$$

ou  $\Sigma$  indique la matrice de covariance du vecteur des couples, et  $C := \left( W^T \Sigma^{-1/2} W \right)^{-1}$  est la matrice de covariance du vecteur des paramètres estimés [75]. L'utilisation de la technique des moindres carrés pondérés permet d'éviter les effets de données inexactes [75]. La matrice  $\Sigma$  est une matrice diagonale par bloc, dont les blocs sont les  $N$  matrices de dimension ( $n \times n$ ) notées  $\Sigma_k$ , correspondant à la matrice de covariance

des couples calculés à l'instant d'échantillonnage  $k$ . Son expression est donnée par :

$$\Sigma_k = \frac{1}{(P-1)} \sum_{p=1}^P (T_p(k) - \hat{\tau}(k)) (T_p(k) - \hat{\tau}(k))^T, \quad (3.68)$$

où  $T_p(k)$  représente la  $p^{\text{ème}}$  période à l'instant d'échantillonnage  $k$  des couples mesurés  $\tau$ , sur lequel était appliqué seulement les étapes de filtrage et de sous-échantillonnage. Le calcul de  $\hat{\theta}_B$  en utilisant la relation (3.66) est facile à implémenter, mais cette solution ne garantit pas que la matrice d'inertie correspondante  $M(q)$  soit définie positive. Dans ce qui suit, nous considérons cette contrainte additionnelle au problème (3.66), qui devient :

$$\begin{aligned} \hat{\theta}_B &= \arg \min_{\theta_B} \| W^T \Sigma^{-1/2} (Y - W\theta_B) \|^2 \\ &\text{tel que } M(q, \hat{\theta}_B) > 0 \quad \forall q \in \mathbb{R}^n. \end{aligned} \quad (3.69)$$

Le problème (3.69) consiste à minimiser la carré de la norme de l'erreur pondérée entre les couples articulaires mesurés et calculés, sous la contrainte d'avoir une matrice d'inertie du robot  $M(q)$  toujours définie positive, c'est-à-dire avoir des valeurs propres positives, quelle que soit la configuration du robot. Une condition suffisante pour avoir une matrice d'inertie définie positive pour n'importe quelle configuration du robot est donnée dans [81]. Les auteurs ont montré que des contraintes sur les paramètres physiques (masses, tenseurs d'inertie autour des centres de masse, coordonnées des centres de masse) des articulations qui composent le robot conduisent à une matrice d'inertie définie positive. Les contraintes sur les articulations sont [81] :

$$\begin{cases} m_i > 0, & i = 1, \dots, n \\ {}^i I_{g_i} > 0, & {}^i I_{g_i} = \begin{bmatrix} xx_{g_i} & xy_{g_i} & xz_{g_i} \\ xy_{g_i} & yy_{g_i} & yz_{g_i} \\ xz_{g_i} & yz_{g_i} & zz_{g_i} \end{bmatrix} \end{cases} \quad (3.70)$$

où  $m_i$  et  ${}^i I_{g_i}$  sont la masse et la matrice du tenseur d'inertie autour du centre de masse  $g_i$  du corps  $i$ , respectivement. La relation entre  ${}^i I_i$  et  ${}^i I_{g_i}$  est donnée par l'équation suivante [81] :

$${}^i I_i = {}^i I_{g_i} + m_i ({}^i r_i^T {}^i r_i E_3 - {}^i r_i {}^i r_i^T), \quad (3.71)$$

où  ${}^i r_i = [x_i \ y_i \ z_i]^T$  est le vecteur entre les points  $O_i$  et  $g_i$ , et  $E_3$  désigne la matrice identité de dimension  $(3 \times 3)$ . On note par  $\theta_{I_i}$  les paramètres inertiels articulaires mentionnés dans (3.60) :

$$\theta_{I_i} := [xx_i \ xy_i \ xz_i \ yy_i \ yz_i \ zz_i \ mx_i \ my_i \ mz_i \ m_i]^T. \quad (3.72)$$

Les expressions analytiques des composantes du vecteur des paramètres de base  $\theta_B$  défini dans la sous-section 3.5.2, sont fonction de  $\theta_{I_i}$ ,  $Fs_i$  et  $Fv_i$ ,  $i = 1, \dots, n$  :

$$\theta_B = f(\{\theta_{I_i}, Fs_i, Fv_i\}_{i=1, \dots, n}). \quad (3.73)$$

Les logiciels destinés à la modélisation symbolique des robots comme `OpenSYMORO` permettent d'avoir les expressions des composantes de  $\theta_B$ , qui sont en effet, des regroupements des paramètres inertiels. On définit les *paramètres physiques articulaires*  $\theta_{P_i}$  comme le vecteur des éléments de  ${}^iI_{g_i}$ ,  ${}^i r_i$  et  $m_i$  :

$$\theta_{P_i} := \left[ \begin{array}{cccccccccccc} xx_{g_i} & xy_{g_i} & xz_{g_i} & yy_{g_i} & yz_{g_i} & zz_{g_i} & x_i & y_i & z_i & m_i \end{array} \right]^T. \quad (3.74)$$

On compare (3.72) avec (3.74), quelques éléments de  $\theta_{I_i}$  sont les mêmes dans  $\theta_{P_i}$  comme  $m_i$ . Par contre, les autres composantes de  $\theta_{I_i}$  dépendent linéairement ( $mx_i$ ,  $my_i$ ,  $mz_i$ ) ou non linéairement ( $xx_i$ ,  $xy_i$ ,  $xz_i$ ,  $yy_i$ ,  $yz_i$ ,  $zz_i$ ) des éléments de  $\theta_{P_i}$ .

On note par  $h_i$  la relation générale entre les vecteurs  $\theta_{I_i}$  et  $\theta_{P_i}$ , pour  $i = 1, \dots, n$  :

$$\begin{aligned} \theta_{I_i} &= h_i(\{\theta_{P_i}\}), \\ \theta_{I_i} &= \begin{bmatrix} xx_i \\ xy_i \\ xz_i \\ yy_i \\ yz_i \\ zz_i \\ mx_i \\ my_i \\ mz_i \\ m_i \end{bmatrix} = \begin{bmatrix} xx_{g_i} + m_i(y_i^2 + z_i^2) \\ xy_{g_i} - m_i x_i y_i \\ xz_{g_i} - m_i x_i z_i \\ yy_{g_i} + m_i(x_i^2 + z_i^2) \\ yz_{g_i} - m_i y_i z_i \\ zz_{g_i} + m_i(x_i^2 + y_i^2) \\ m_i x_i \\ m_i y_i \\ m_i z_i \\ m_i \end{bmatrix}. \end{aligned} \quad (3.75)$$

En utilisant les relations (3.73) et (3.75), le vecteur des paramètres de base peut être ré-écrit comme une fonction non linéaire de  $\vartheta_i^T := [\theta_{P_i}^T \quad F s_i \quad F v_i]$  :

$$\begin{aligned} \theta_B &= f(\{\theta_{I_i}, F s_i, F v_i\}_{i=1, \dots, n}) \\ &= f(\{h_i(\{\theta_{P_i}\}), F s_i, F v_i\}_{i=1, \dots, n}) \\ &= g(\{\vartheta_i\}_{i=1, \dots, n}). \end{aligned} \quad (3.76)$$

Selon [81], afin d'obtenir une matrice d'inertie définie positive  $M(q)$ , il suffit que tous les  $m_i$  et  ${}^iI_{g_i}$  contenus dans l'équation (3.76) satisfassent la condition (3.70). L'ensemble des  $m_i$  et  ${}^iI_{g_i}$  pouvant satisfaire la condition (3.70) n'est pas unique. Dans [81], le vecteur des paramètres trouvé avec ces ensembles est appelé *vecteur des paramètres virtuels*. Le problème (3.69) est transformé en un nouveau problème d'optimisation sous contraintes non linéaires, pour lequel la solution est le vecteur  $\vartheta^T = [\vartheta_1^T \quad \dots \quad \vartheta_n^T]$  :



$$\hat{\vartheta} = \arg \min_{\vartheta} \| W^T \Sigma^{-1} (Y - Wg(\vartheta)) \|^2$$

$$\text{tel que } \begin{cases} m_i > 0, & {}^i I_{g_i} > 0, \\ F s_i > 0, & F v_i > 0, \end{cases} \quad i = 1, \dots, n. \quad (3.77)$$

La contrainte sur la matrice du tenseur d'inertie estimée  ${}^i I_{g_i}$  est équivalente pour que cette dernière ait toutes ses valeurs propres positives. Les valeurs estimées de  $F s_i$  et  $F v_i$  ne pouvant être que positives, cela justifie l'ajout des contraintes sur ces paramètres dans (3.77). Pour terminer la procédure, le vecteur estimé  $\hat{\vartheta}$  est substitué dans (3.76) pour obtenir l'estimation correspondante au vecteur des paramètres de base  $\hat{\theta}_B$ , nécessaire pour construire une matrice d'inertie estimée  $\hat{M}(q)$ . En raison de sa caractéristique non linéaire, il convient de noter que le problème d'optimisation dans (3.77) peut contenir plusieurs minima locaux. Il devrait ensuite être abordé par des solveurs d'optimisation globale. La recherche d'un minimum global peut être une tâche fastidieuse.

### 3.5.6 Identification des paramètres de la charge utile du robot

Les couples articulaires dans le vecteur  $Y$  (3.64) peuvent être vus comme la somme de ceux résultant des corps du robot seul, dans un vecteur noté  $Y_r$ , et ceux dus à sa charge utile, dans un vecteur noté  $Y_p$  [75], plus un vecteur regroupant l'effet des bruits et des dynamiques non modélisées, vecteur noté  $R$ . Il vient alors :

$$Y = Y_r + Y_p + R \quad (3.78)$$

$$= Y_r + W_p \theta_p + R, \quad (3.79)$$

où  $W_p$  est la matrice d'observation de dimension  $(n.N \times n_{\theta_p})$ ;  $\theta_p$  est le vecteur des paramètres composé des paramètres inertiels de la charge, et les variations des paramètres de frottement articulaire ( $\Delta F s_i$  et  $\Delta F v_i$ ) pour toutes les articulations. Les  $\Delta F s_i$  et  $\Delta F v_i$  sont ajoutés au vecteur des paramètres inertiels de la charge car en effet, les paramètres de frottement peuvent changer quand une charge utile est ajoutée au robot [75]. Comme le repère attaché à la charge est fixe par rapport à la  $n^{\text{ème}}$  articulation,  $W_p$  est composée des colonnes de  $W$  qui correspondent à tous les paramètres de frottement articulaire, et aux paramètres inertiels de base de la  $n^{\text{ème}}$  articulation. Notons que  $W$  est la matrice d'observation du robot avec sa charge utile. Pour identifier le vecteur des paramètres  $\theta_p$ , l'acquisition de deux séries de données est nécessaire. La première correspond au robot sans charge, tandis que la deuxième est acquise pour le robot avec cette dernière. L'estimation de  $\theta_p$  peut être obtenue par la résolution du problème suivant :

$$\hat{\theta}_p = \arg \min_{\theta_p} \| W_p^T \Sigma^{-1/2} (Y - Y_r - W_p \theta_p) \|^2$$

$$\text{tel que } \begin{cases} m_p > 0, & {}^n I_{g_p} > 0, \\ \Delta F s_i > 0, & \Delta F v_i > 0, \end{cases} \quad i = 1, \dots, n \quad (3.80)$$

où  $Y_r$  contient les données des couples de la première série de données (sans charge utile), alors que  $Y$  et  $W_p$  contiennent les données de la deuxième série (avec charge). Le problème (3.80) se résout par la méthode décrite dans la sous-section précédente.

### 3.5.7 Validation

Afin de tester la validité des paramètres estimés par la procédure décrite précédemment, ces paramètres sont utilisés pour reconstruire les couples articulaires. Pour cela, une deuxième série de données différente de celle utilisée en identification est exploitée. En effet, il est préférable d'utiliser plusieurs jeux de données avec des caractéristiques différentes afin de valider le modèle dynamique bloc par bloc ( $M$ ,  $N$ ,  $G$  et  $\tau_f$ ), notamment : des signaux de position à excitation persistante, invariants dans le temps, et à variation régulière dans le temps (la pente du signal est constante). Ensuite, les vecteurs des positions, vitesses et accélérations articulaires mesurées/calculées et post-traitées sont utilisées comme des entrées pour le MDI (3.53) avec les paramètres identifiés. Les couples obtenus sont comparés avec ceux mesurés. Il est indispensable de garantir, avant de procéder à l'application d'un schéma de commande sur le robot réel, une bonne qualité de reconstruction des couples, avec une erreur minimale centrée autour de zéro. D'où, l'étape de validation est nécessaire afin d'avoir une information sur l'intervalle de confiance des paramètres estimés, et en même temps, sur la validité du MDD du robot.

## 3.6 Identification des paramètres dynamiques du robot Denso VP-6242G

### 3.6.1 Vecteur des paramètres de base

Le logiciel `OpenSYMORO` est utilisé pour calculer le modèle dynamique pour l'identification du robot. Le vecteur des paramètres de base  $\theta_B$  est de dimension  $(48 \times 1)$  :

$$\theta_B = \begin{bmatrix} zzr_1 & Fs_1 & Fv_1 & xxr_2 & xy_2 & xzr_2 & yz_2 & zzr_2 & mxr_2 & my_2 & Fs_2 & Fv_2 \\ xxr_3 & xyr_3 & xz_3 & yz_3 & zzr_3 & mxr_3 & myr_3 & Fs_3 & Fv_3 & xxr_4 & xy_4 & xz_4 \\ yz_4 & zzr_4 & mx_4 & myr_4 & Fs_4 & Fv_4 & xxr_5 & xy_5 & xz_5 & yz_5 & zzr_5 & mx_5 \\ myr_5 & Fs_5 & Fv_5 & xxr_6 & xy_6 & xz_6 & yz_6 & zz_6 & mx_6 & my_6 & Fs_6 & Fv_6 \end{bmatrix}^T. \quad (3.81)$$

Les équations (3.82) et (3.83) présentent les expressions des paramètres de base  $\theta_B$  (équation (3.73)) en fonction des paramètres inertiels  $\theta_{I_i}$ ,  $Fs_i$  et  $Fv_i$  de chaque articulation  $i$ . Les expressions sont fournies par le logiciel `OpenSYMORO`.

$$\begin{cases} zzr_1 &= (m_3 + m_4 + m_5 + m_6)l_2^2 + yy_2 + (m_4 + m_5 + m_6)l_{3x}^2 + yy_3 + zz_1; \\ Fs_1 &= Fs_1; \\ Fv_1 &= Fv_1; \\ xxr_2 &= xx_2 - (m_3 + m_4 + m_5 + m_6)l_2^2 - yy_2; \end{cases} \quad (3.82)$$

$$\left\{ \begin{array}{l}
xy_2 = xy_2; \\
xZR_2 = xz_2 - mz_3l_2; \\
yZ_2 = yz_2; \\
zZR_2 = (m_3 + m_4 + m_5 + m_6)l_2^2 + zZ_2; \\
mXR_2 = (m_3 + m_4 + m_5 + m_6)l_2 + mX_2; \\
mY_2 = mY_2; \\
Fs_2 = Fs_2; \\
Fv_2 = Fv_2; \\
xXR_3 = (m_4 + m_5 + m_6)(l_{3z} + l_4)^2 + 2mz_4(l_{3z} + l_4) + xX_3 + yY_4 \\
\quad - (m_4 + m_5 + m_6)l_{3x}^2 - yY_3; \\
xYR_3 = mz_4l_{3x} + (m_4 + m_5 + m_6)(l_{3z} + l_4)l_{3x} + xY_3; \\
xZ_3 = xz_3; \\
yZ_3 = yz_3; \\
zZR_3 = (m_4 + m_5 + m_6)[(l_{3z} + l_4)^2 + l_{3x}^2] + 2mz_4(l_{3z} + l_4) + yY_4 + zZ_3; \\
mXR_3 = mX_3 - (m_4 + m_5 + m_6)l_{3x}; \\
mYR_3 = (m_4 + m_5 + m_6)(l_{3z} + l_4) + mY_3 + mz_4; \\
Fs_3 = Fs_3; \\
Fv_3 = Fv_3; \\
xXR_4 = xX_4 - yY_4 + yY_5; \\
xY_4 = xY_4; \\
xZ_4 = xz_4; \\
yZ_4 = yz_4; \\
zZR_4 = yY_5 + zZ_4; \\
mX_4 = mX_4; \\
mYR_4 = mY_4 - mz_5; \\
Fs_4 = Fs_4; \\
Fv_4 = Fv_4; \\
xXR_5 = m_6l_5^2 + 2mz_6l_5 + xX_5 - yY_5 + yY_6; \\
xY_5 = xY_5; \\
xZ_5 = xz_5; \\
yZ_5 = yz_5; \\
zZR_5 = m_6l_5^2 + 2mz_6l_5 + yY_6 + zZ_5; \\
mX_5 = mX_5; \\
mYR_5 = m_6l_5 + mY_5 + mz_6; \\
Fs_5 = Fs_5; \\
Fv_5 = Fv_5; \\
xXR_6 = xX_6 - yY_6; \\
xY_6 = xY_6; \\
xZ_6 = xz_6; \\
yZ_6 = yz_6; \\
zZ_6 = zZ_6; \\
mX_6 = mX_6; \\
mY_6 = mY_6; \\
Fs_6 = Fs_6; \\
Fv_6 = Fv_6;
\end{array} \right.$$

(3.83)

Le vecteur des paramètres  $\theta_p$  contenant les paramètres de la charge et les variations des coefficients des frottements est de dimension  $(19 \times 1)$  :

$$\theta_p = \begin{bmatrix} \Delta F s_1 & \Delta F v_1 & \Delta F s_2 & \Delta F v_2 & \Delta F s_3 & \Delta F v_3 & \Delta F s_4 & \Delta F v_4 & \Delta F s_5 & \Delta F v_5 \\ x x r_p & x y_p & x z_p & y z_p & z z_p & m x_p & m y_p & \Delta F s_6 & \Delta F v_6 \end{bmatrix}^T. \quad (3.84)$$

### 3.6.2 Justification du modèle de frottement

Au cours des essais, nous avons constatés que les frottements articulaires sont compliqués à modéliser. Le modèle simplifié, qui est la combinaison des frottements visqueux et sec, est souvent utilisé pour décrire ce phénomène. Le modèle dynamique du robot peut être écrit suivant l'équation suivante [66] :

$$D_I(q, \dot{q}, \ddot{q})\theta_I + \tau_f = \tau \quad (3.85)$$

où  $\theta_I$  est le vecteur des paramètres inertiels, et  $D_I$  est la matrice d'observation associée à  $\theta_I$ . Pour chaque articulation, nous choisissons dans la mesure du possible une configuration du robot pour laquelle la gravité n'a pas d'effet sur l'articulation d'intérêt. Cela permet de minimiser la dimension de  $\theta_I$ , et de supprimer l'influence de la gravité. Ensuite, seule cette articulation est excitée avec un signal d'excitation qui a l'allure illustrée par la figure 3.13 gauche, tout en respectant les contraintes en position et en vitesse articulaires. Cela permet d'avoir une vitesse articulaire positive et négative, croissante et décroissante (figure 3.13 droite).

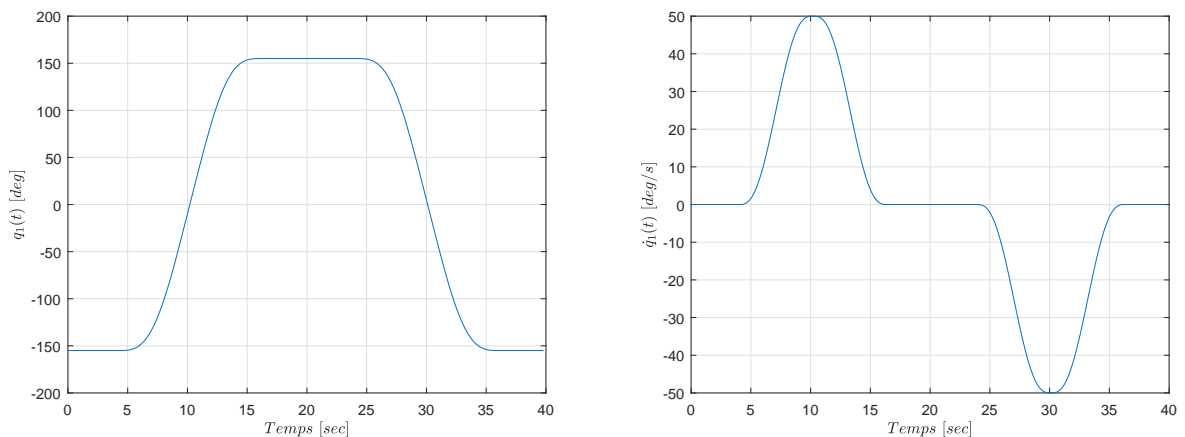


FIGURE 3.13 – Allure de la position angulaire (gauche) et de la vitesse articulaire (droite) correspondante pour l'axe 1, avec une vitesse  $\dot{q}_1$  maximale de  $50^\circ/sec$ .

Le vecteur  $\theta_I$  est estimé par la méthode des moindres carrés décrite précédemment, sans contraintes a priori. Ensuite, nous procédons au tracé de la caractéristique  $\tau_{f_i} = f(\dot{q}_i)$  en tenant compte de la relation

$$\tau_{f_i} = \tau_i - D_{I_i}\theta_{I_i}. \quad (3.86)$$

Les figures suivantes présentent, pour les axes 1, 2, 3 et 5, les couples de frottement en fonction des vitesses articulaires pour différentes amplitudes maximales des vitesses articulaires.

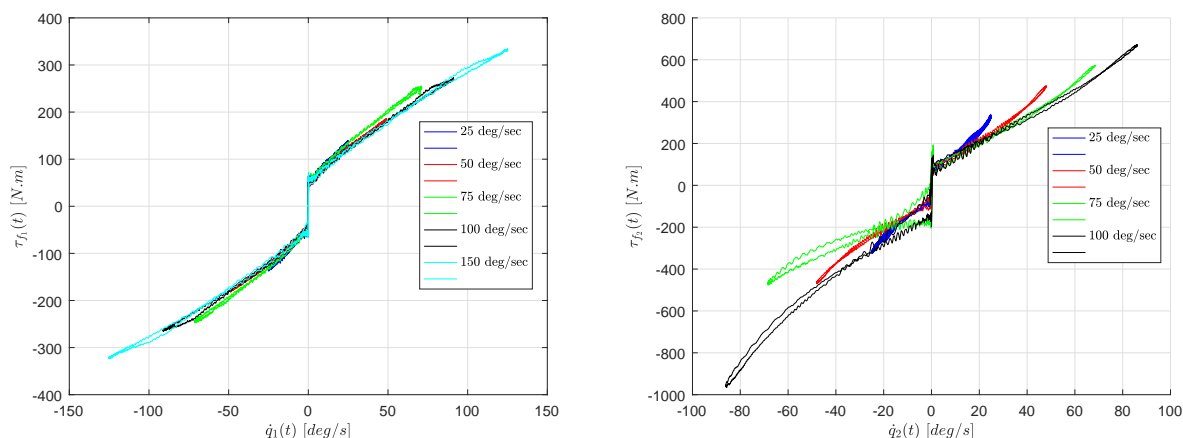


FIGURE 3.14 – Caractéristique couple de frottement en fonction de la vitesse articulaire pour les axes 1 (gauche) et 2 (droite).

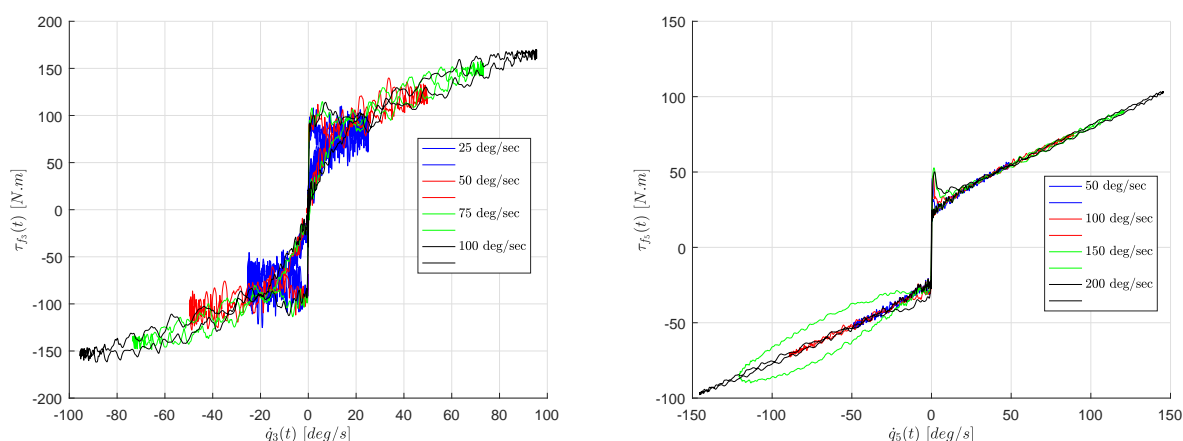


FIGURE 3.15 – Caractéristique couple de frottement en fonction de la vitesse articulaire pour les axes 3 (gauche) et 5 (droite).

Nous remarquons que presque tous les signaux contiennent une partie non linéaire, pour laquelle l'exercice d'un couple articulaire  $\tau_i$  n'entraîne pas l'articulation en mouvement ( $\dot{q}_i$  correspondante reste nulle). Aussi, nous constatons, en dehors de la zone non linéaire, une relation linéaire entre les  $\tau_{f_i}$  et les  $\dot{q}_i$ . Les signaux sont quasi-symétriques par rapport au point ( $\tau_{f_i} = 0, \dot{q}_i = 0$ ). Nous pouvons conclure qu'un modèle de frottement composé des termes sec et visqueux peut décrire les frottements articulaires. Malgré ça, il est vraiment difficile d'avoir un modèle exact des frottements à cause de plusieurs facteurs notamment, la qualité des surfaces en contact, et les caractéristiques physiques des matériaux composant les corps du robot : les variations de la température des matériaux en fonction de la durée de frottement et de la vitesse du mouvement articulaire.

### 3.6.3 Signaux d'excitation

Afin d'appliquer les signaux d'excitation au robot, nous avons utilisé une commande en position articulaire avec un correcteur proportionnel pour chaque axe. Les signaux sont générés par la fonction `Matlab idinput`, permettant de générer un signal qui est une somme de sinusoides (multisine). La fonction sélectionne les fréquences des sinusoides pour qu'elles soient réparties d'une manière égale sur une grille choisie, et attribue à chaque sinusoides une phase aléatoire. Ensuite, `idinput` essaie plusieurs phases aléatoires pour chaque sinusoides et sélectionne les phases donnant le plus petit signal propagé. La propagation du signal est la différence entre la valeur minimale et maximale du signal sur tous les échantillons. L'amplitude du signal multisine généré est mise à l'échelle pour satisfaire l'amplitude prédéfinie. La commande `idinput` permet aussi de spécifier les caractéristiques des sinusoides utilisés pour générer le signal, telles que leurs nombre et fréquences. On considère que les signaux générés par cette commande sont typiquement utilisés en identification comme des trajectoires désirées pour les positions articulaires. Chaque signal contient 55 sinusoides à des fréquences différentes, qui sont multiples d'une fréquence fondamentale  $f_{0_i}$ . Le nombre des sinusoides est justifié par la condition qu'il doit être supérieur à la dimension du vecteur des paramètres de base à estimer, divisé par 2, pour assurer que les signaux résultants possèdent la propriété d'excitation persistante d'ordre appropriée. Aussi, toutes les fréquences sont inférieures à une fréquence limite, définie comme suit :

$$f_i = \frac{A_{\dot{q},i}}{2\pi A_{q,i}} > 55 f_{0_i}, \quad (3.87)$$

où  $A_{q,i}$  et  $A_{\dot{q},i}$  sont les amplitudes maximales des positions et vitesses articulaires pour chaque axe  $i$ . Les  $A_{q,i}$  sont déterminées afin d'éviter les collisions entre le robot et son environnement, et aussi entre ses propres corps [75]. Les  $A_{\dot{q},i}$  sont calculés à partir de la caractéristique temps de positionnement de chaque moteur, qui donne une relation entre le temps de positionnement et le mouvement angulaire, selon le poids de la charge portée par l'organe terminal du robot. La figure 3.16 montre cette caractéristique pour les six axes du robot.

Sur la figure 3.16, la caractéristique temps de positionnement est présentée pour différentes valeurs de la masse de la charge portée par le robot. Dans notre cas, nous avons utilisé par précaution, les courbes correspondant à une masse de 2.5 kg, bien que l'organe terminal ne pèse que 0.940 kg, afin de se prémunir d'un éventuel ajout de pièce supplémentaire. Nous avons déterminé les  $A_{\dot{q},i}$  comme les pentes des parties droites de chaque courbe.

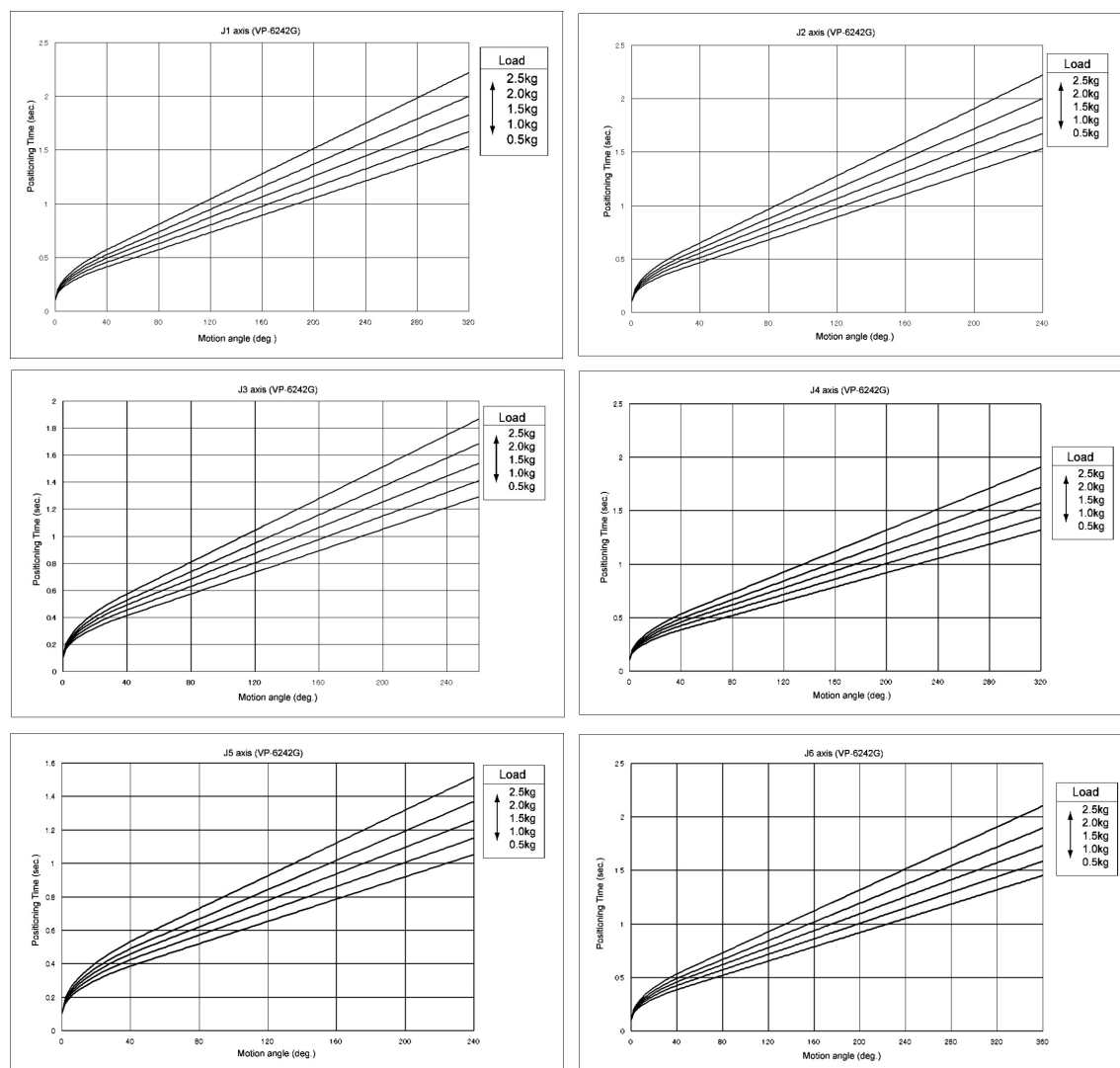


FIGURE 3.16 – Caractéristique temps de positionnement pour les six axes du robot Denso VP-6242G.

Afin d'améliorer la fiabilité de l'estimation, les signaux d'excitation périodiques ont été construits avec 12 périodes identiques, chaque période possédant une durée de 380 *sec*. Le pas d'échantillonnage est imposé à  $10^{-3}$  *sec*. La durée de la période dépend essentiellement du nombre de fréquences à générer et la fréquence limite. Si la fréquence limite est petite et le nombre de fréquences est important, la durée de la période est par conséquent grande, et le contraire est vrai. Afin d'assurer que le robot ne dépasse pas les limites des intervalles de mouvement articulaire prédéfinies, nous imposons aux positions articulaires initiales d'être au milieu de ces intervalles. De plus, le mouvement commence et se termine avec des vitesses et accélérations articulaires nulles. Pour cela, une phase d'immobilité de 4 *sec* est ajoutée au début et à la fin du mouvement. Durant cette phase, le robot reste immobile à sa position initiale. Une phase de connexion de 5 *sec* est aussi ajoutée pour raccorder le profil de la phase initiale (position, vitesse et accélération) du robot avec les signaux multisines. La durée totale de l'excitation est de 4578 *sec*. Le tableau 3.2 présente, pour chaque axe  $J_i$ , l'intervalle de mouvement articulaire  $[q_{i,m}, q_{i,M}]$ , les amplitudes  $A_{q,i}$  et  $A_{\dot{q},i}$ , la fréquence limite  $f_i$ , et le gain  $K_{p_i}$  du correcteur proportionnel utilisé pour l'asservissement en position

articulaire.

TABLE 3.2 – Paramètres des signaux d’excitation générés.

Axe	$[q_{i,m}, q_{i,M}]$ [°]	$A_{q,i}$ [°]	$A_{\dot{q},i}$ [°/s]	$f_i$ [Hz]	$K_{p_i}$
$J_1$	[-135, +135]	135	$\pm 168.7882$	0.1989	1.2838
$J_2$	[-45, +90]	67.5	$\pm 126.4223$	0.2981	1.6683
$J_3$	[-90, -40]	25	$\pm 168.3147$	1.0715	4.4711
$J_4$	[-155, +155]	155	$\pm 202.1053$	0.2075	1.3116
$J_5$	[-90, +90]	90	$\pm 205.1282$	0.3627	1.7577
$J_6$	[-160, +160]	160	$\pm 203.6364$	0.2025	1.6678

### 3.6.4 Traitement des signaux mesurés

D’abord, nous récupérons les mesures des positions et courants articulaires. Pour calculer les couples, les courants articulaires sont multipliés par les constantes de couple  $K_{c_i}$  et les rapports de réduction de vitesse  $G_{r_i}$  correspondants, qui sont présentés dans le tableau 3.3.

TABLE 3.3 – Spécifications des moteurs du robot Denso VP-6242G.

Axe	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$	$J_6$
$K_{c_i}$	0.38	0.38	0.22	0.21	0.21	0.21
$G_{r_i}$	120	160	120	100	100	100

Ensuite, les signaux sont filtrés par un filtre passe-bas de type **Butterworth** d’ordre 1, et une fréquence de coupure de 10 Hz. Pour cela, on utilise la fonction **Butter** de **Matlab**. Pour éviter les déphasages dus aux propriétés inhérentes des filtres linéaires, la fonction non-causale de **Matlab** **filtfilt** est utilisée pour avoir un filtrage avec un déphasage nul pour toutes les mesures. Après, les vitesses et accélérations articulaires sont calculées par dérivation numérique (schéma d’Euler implicite) des signaux de positions articulaires filtrées. Afin d’éliminer les éventuels bruits dus à la dérivation numérique, les signaux de vitesse et d’accélération sont aussi filtrés avec le même filtre. Dans une étape suivante, les phases d’immobilité et de raccordement sont supprimées du jeu de données, pour ne conserver que les parties périodiques des signaux. Les signaux moyens sont ensuite calculés sur les 12 périodes disponibles grâce à la relation (3.62). Pour finir, puisque la fréquence d’échantillonnage ( $10^3$  Hz) est suffisamment élevée par rapport à la fréquence de coupure du filtre (10 Hz), nous procédons au sous-échantillonnage des données avec un facteur de sous-échantillonnage égal à 10. La fréquence d’échantillonnage des données sous-échantillonnées est alors



de  $10^2$  Hz, ce qui reste élevé par rapport à la plus haute fréquence contenue dans les signaux d'excitation (tableau 3.2). Nous constatons que les données ne sont pas affectées par l'opération de sous-échantillonnage. En parallèle, cette opération nous a permis de réduire la taille des données, en éliminant des composantes fréquentielles à hautes-fréquences, sans intérêts pour notre identification.

Un zoom entre les instants 1200 sec et 1400 sec des signaux mesurés des positions et couples, et des signaux calculés des vitesses et accélérations pour chaque axe, avant (bleu) et après (rouge) filtrage sont illustrés sur les figures 3.17, 3.18, 3.19, 3.20, 3.21 et 3.22 à gauche. Les erreurs correspondantes sont illustrées à droite sur les mêmes figures. Tous les signaux présentés dans cette sous-section sont récupérés quand la charge utile (capteur de force + outil) est fixée sur le dernier corps du robot.

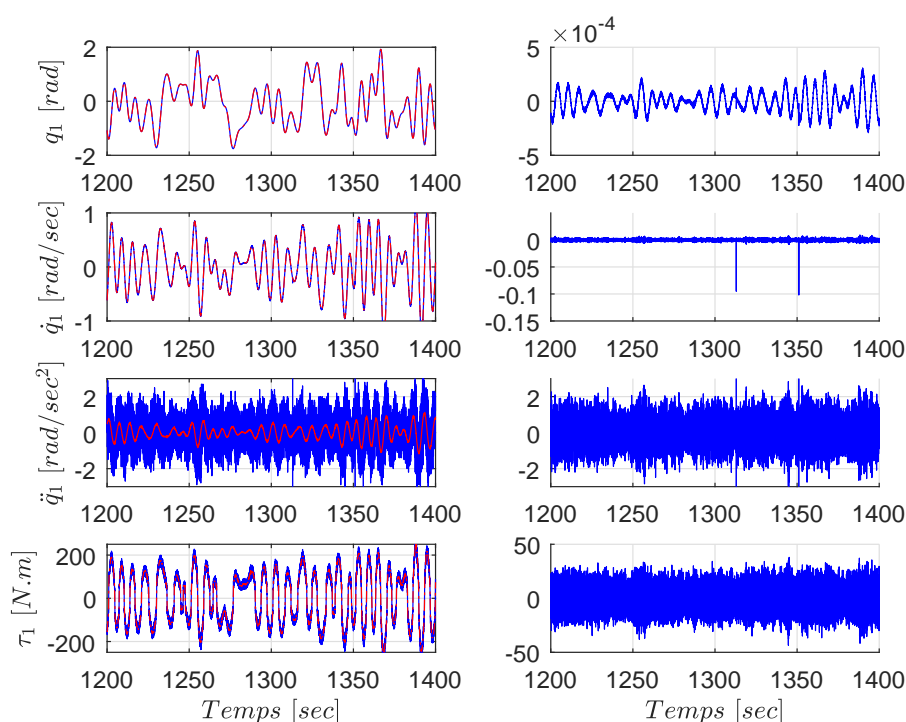


FIGURE 3.17 – Signaux de données mesurées  $q_1$ ,  $\tau_1$ , et calculées  $\dot{q}_1$ ,  $\ddot{q}_1$  de l'axe 1 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

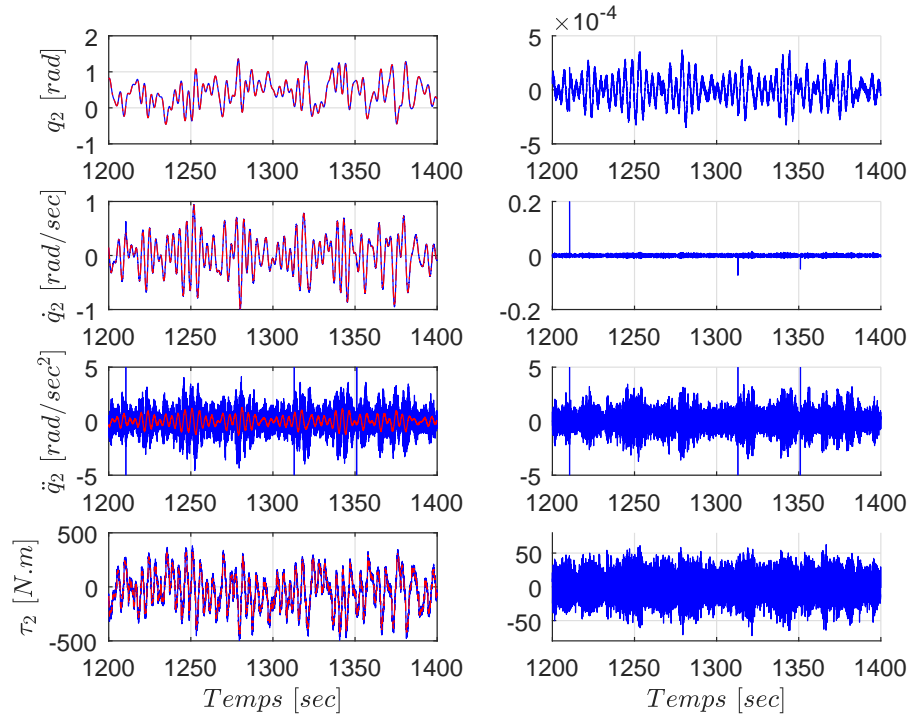


FIGURE 3.18 – Signaux de données mesurées  $q_2$ ,  $\tau_2$ , et calculées  $\dot{q}_2$ ,  $\ddot{q}_2$  de l'axe 2 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

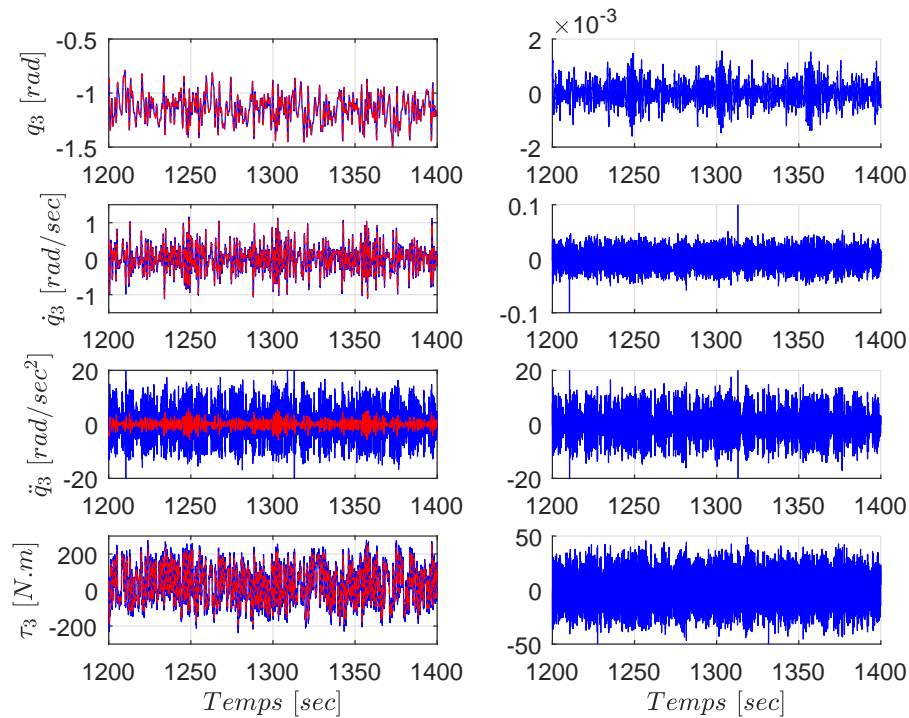


FIGURE 3.19 – Signaux de données mesurées  $q_3$ ,  $\tau_3$ , et calculées  $\dot{q}_3$ ,  $\ddot{q}_3$  de l'axe 3 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

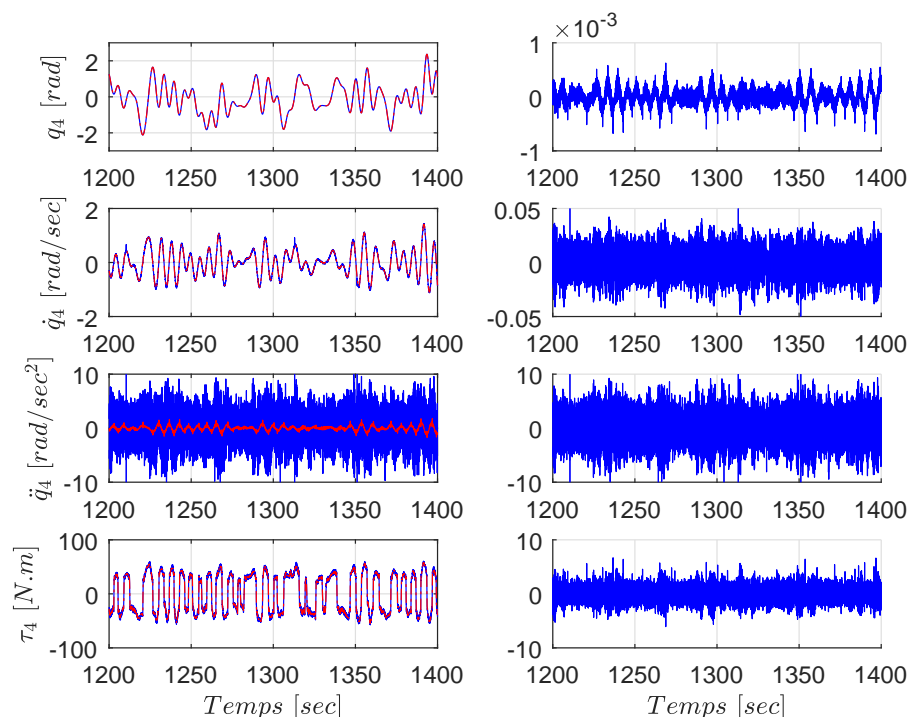


FIGURE 3.20 – Signaux de données mesurées  $q_4$ ,  $\tau_4$ , et calculées  $\dot{q}_4$ ,  $\ddot{q}_4$  de l'axe 4 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

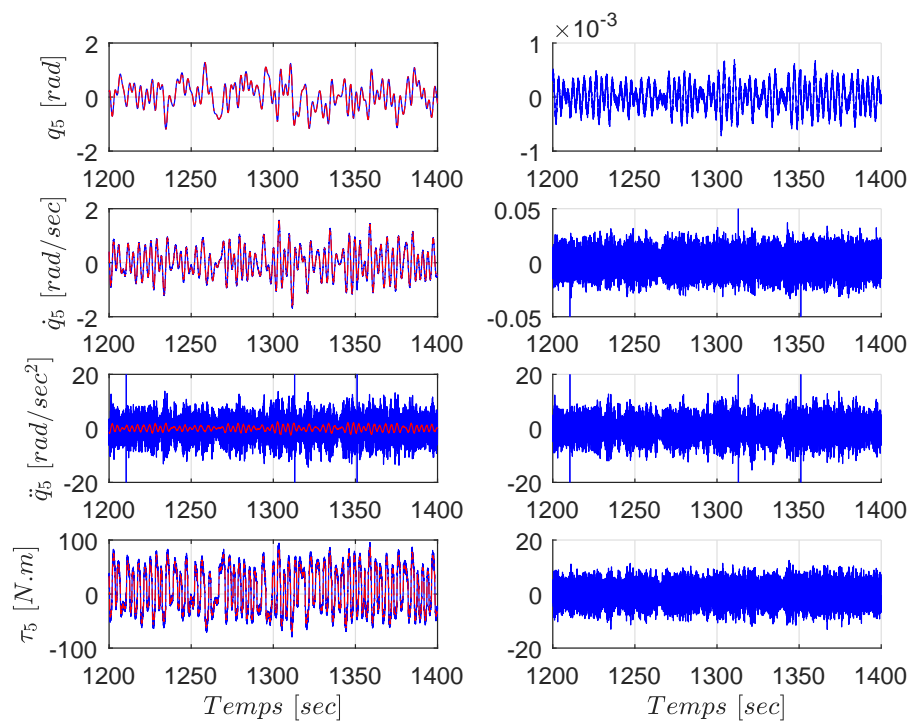


FIGURE 3.21 – Signaux de données mesurées  $q_5$ ,  $\tau_5$ , et calculées  $\dot{q}_5$ ,  $\ddot{q}_5$  de l'axe 5 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

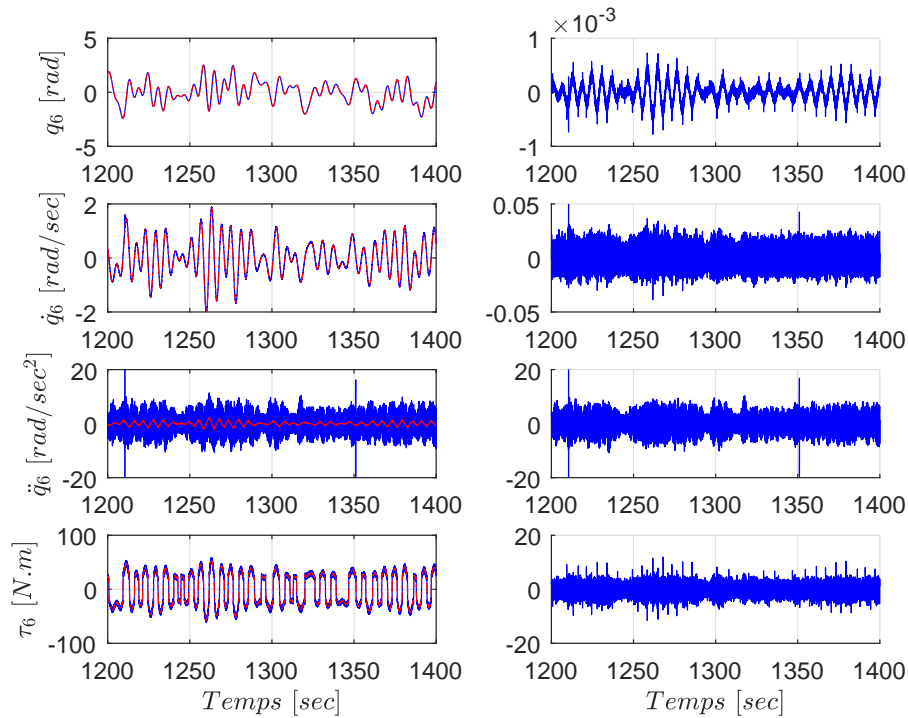


FIGURE 3.22 – Signaux de données mesurées  $q_6$ ,  $\tau_6$ , et calculées  $\dot{q}_6$ ,  $\ddot{q}_6$  de l'axe 6 avant (bleu) et après (rouge) filtrage (à gauche), et l'erreur correspondante (à droite).

Nous pouvons remarquer que les données ont été filtrées correctement, en diminuant considérablement l'effet des bruits de mesure. Nous en concluons que le choix des paramètres du filtre (fréquence de coupure et ordre) est approprié. L'erreur importante au niveau du signal de l'accélération articulaire est traduite par le bruit significatif associé au signal, comme il résulte d'une double dérivation numérique de la position.

Les signaux des mesures de position et de couples, ainsi que les signaux calculés de vitesse et d'accélération articulaires, après le traitement détaillé dans cette sous-section (filtrage, calcul du signal moyen et sous-échantillonnage) sont présentés sur les figures 3.23, 3.24 et 3.25.

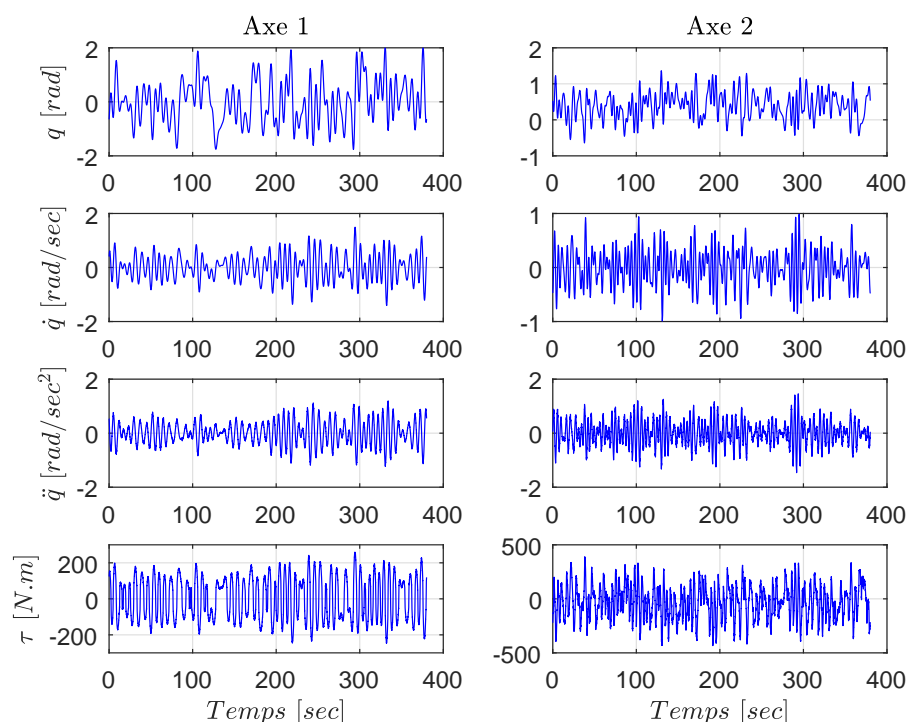


FIGURE 3.23 – Signaux de données mesurées et calculées des axes 1 et 2 résultant du traitement décrit ci-dessus.

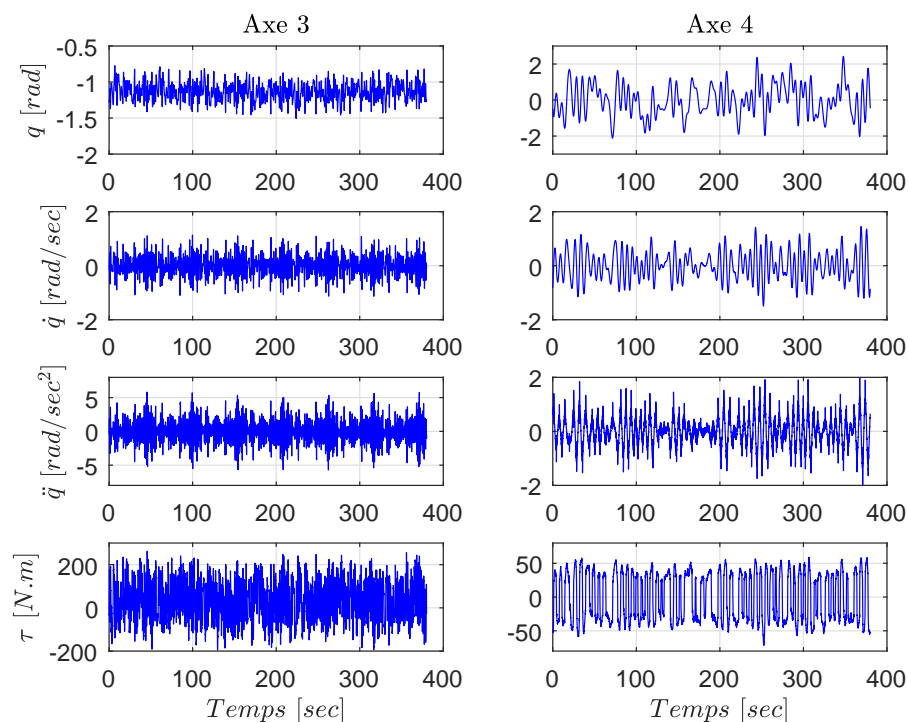


FIGURE 3.24 – Signaux de données mesurées et calculées des axes 3 et 4 résultant du traitement décrit ci-dessus.

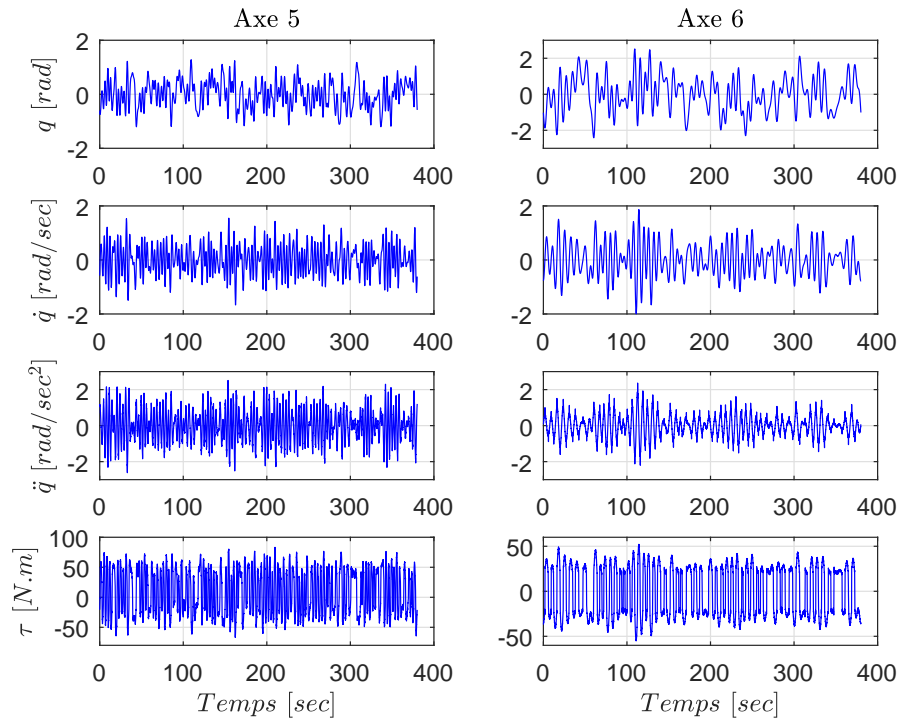


FIGURE 3.25 – Signaux de données mesurées et calculées des axes 5 et 6 résultant du traitement décrit ci-dessus.

Étant donné la qualité du filtrage obtenu, le nombre de période des signaux (supérieur au nombre des axes du robot), et le facteur de sous-échantillonnage appliqué (qui permet de conserver les informations essentielles dans les signaux), nous pouvons considérer que les données résultantes peuvent être exploitées pour la procédure d'identification des paramètres.

### 3.6.5 Estimation des paramètres

D'abord, nous commençons par la construction des matrices d'observation  $W$  et de covariance  $\Sigma$ , ainsi que le vecteur des couples  $Y$  pour les deux cas de figure du robot, avec et sans charge. Le rang des matrices d'observation est vérifié de rang plein. Le conditionnement de la matrice d'observation  $W$  est de 217 pour les deux cas, qui est considéré assez réduit. On conclut que l'influence des éventuels bruits encore associés aux données poste-traitées n'est pas significative. Et de là, ça n'affecte pas le processus de l'identification.

Pour le cas du robot Denso VP-6242G,  $\vartheta$  est de dimension  $(66 \times 1)$  et  $\vartheta_p$  est de dimension  $(22 \times 1)$ ,  $\vartheta_p$  correspondant aux paramètres physiques du vecteur  $\theta_p$ . L'estimation des vecteurs des paramètres  $\vartheta$  avec et sans charge et  $\vartheta_p$  nécessite la résolution des problèmes d'optimisation sous contraintes non linéaires exprimés dans les équations (3.77) et (3.80). Pour cela, nous avons utilisé un algorithme d'optimisation globale de Matlab, codé dans la fonction `GlobalSearch`. Cette fonction repose sur la fonction `fmincon` qui intègre un algorithme de résolution d'un problème d'optimisation

non-linéaire, sous différents types de contraintes. Cela permet de trouver la solution minimale pour le problème (3.77). Le tableau 3.4 regroupe les valeurs numériques des paramètres estimés du robot avec et sans charge, par la résolution du problème (3.66) sans contraintes (MCP avec) & (MCP sans), et par la résolution du problème d'optimisation sous contraintes (3.77) (MCPC avec) & (MCPC sans). L'acronyme MCPC signifie Moindres Carrés Pondérés sous Contraintes. Les valeurs des paramètres sont exprimées dans le système international d'unités (SI). Les quatrième et septième colonnes présentent l'écart type  $\sigma$  du vecteur des paramètres estimés, défini comme la racine carrée des éléments diagonaux de la matrice de covariance  $C$ .

TABLE 3.4 – Valeurs numériques de  $\theta_B$  avec et sans charge (exprimées en SI), et l'écart type des paramètres  $\sigma$ .

$\theta_B$	MCPC (avec)	MCP (avec)	$\sigma$ (avec)	MCPC (sans)	MCP (sans)	$\sigma$ (sans)
$z z r_1$	7.10	9.05	0.0364	5.16	9.21	0.0313
$F s_1$	49.74	49.38	0.0076	47.44	47.02	0.0070
$F v_1$	150.81	151.39	0.0278	122.89	124.02	0.0231
$x x r_2$	-5.96	0.20	0.0446	-3.28	-1.01	0.0400
$x y_2$	-1.01	-1.22	0.0215	-1.60	-0.98	0.0188
$x z r_2$	-1.17	-1	0.0135	-1.30	-1.20	0.0120
$y z_2$	-0.97	-0.81	0.0219	-0.85	-0.53	0.0194
$z z r_2$	17.29	17.71	0.0256	15.32	15.87	0.0217
$m x r_2$	18.67	18.77	0.0025	16.01	16.01	0.0023
$m y_2$	0.28	0.31	0.0012	0.14	0.18	0.0011
$F s_2$	63.64	64.61	0.0112	60.72	61.48	0.0093
$F v_2$	185.75	180.34	0.0480	168.22	164.72	0.0399
$x x r_3$	9.04	-0.32	0.0289	7.43	0.22	0.0249
$x y r_3$	0.41	2.27	0.0191	0.15	1.23	0.0158
$x z_3$	0.05	0.10	0.0056	0.07	0.1	0.0050
$y z_3$	0.04	0.04	0.0059	0.01	-0.00	0.0052
$z z r_3$	4.37	4.20	0.0041	3.07	2.90	0.0035
$m x r_3$	-5.16	-4.98	0.0012	-4.15	-4.03	0.0010
$m y r_3$	8.77	8.79	0.0013	5.76	5.78	0.0011
$F s_3$	54.78	54.87	0.0081	52.23	52.25	0.0069
$F v_3$	87.92	86.78	0.0295	83.55	83.30	0.0263
$x x r_4$	-0.21	0.13	0.0073	-0.16	0.16	0.0063
$x y_4$	0.19	0.19	0.0033	0.14	0.12	0.0028

$\theta_B$	MCPC (avec)	MCP (avec)	$\sigma$ (avec)	MCPC (sans)	MCP (sans)	$\sigma$ (sans)
$xz_4$	0.23	0.21	0.0028	0.25	0.23	0.0023
$yz_4$	-0.06	-0.07	0.0028	-0.03	-0.05	0.0024
$zzr_4$	0.06	-0.86	0.0057	0.06	-0.83	0.0048
$mx_4$	0.30	0.24	0.0007	0.29	0.24	0.0006
$myr_4$	0.20	0.20	0.0006	0.21	0.21	0.0005
$Fs_4$	24.63	24.62	0.0047	24.04	24.12	0.0040
$Fs_4$	21.55	21.17	0.0113	19.20	18.82	0.0090
$x xr_5$	0.22	0.29	0.0048	0.05	0.12	0.0041
$xy_5$	0.01	0.01	0.0024	0.00	0.01	0.0020
$xz_5$	-0.02	-0.01	0.0017	0.01	0.00	0.0015
$yz_5$	0.01	0.01	0.0020	0.03	0.03	0.0018
$zzr_5$	0.56	0.59	0.0019	0.28	0.33	0.0018
$mx_5$	0.13	0.13	0.0004	0.08	0.08	0.0004
$myr_5$	2.12	2.11	0.0004	0.40	0.40	0.0004
$Fs_5$	20.45	20.40	0.0037	19.30	19.31	0.0036
$Fv_5$	29.29	29.12	0.0096	24.77	24.62	0.0095
$x xr_6$	-0.07	-0.08	0.0023	-0.02	-0.03	0.0020
$xy_6$	0.03	0.03	0.0012	-0.02	-0.01	0.0011
$xz_6$	0.02	0.02	0.0012	-0.02	-0.02	0.0011
$yz_6$	0.00	0.00	0.0013	0.00	0.00	0.0012
$zz_6$	0.00	0.05	0.0023	0.05	0.12	0.0020
$mx_6$	0.10	0.10	0.0004	-0.02	-0.01	0.0003
$my_6$	-0.05	-0.05	0.0003	-0.01	-0.01	0.0003
$Fs_6$	19.08	19.11	0.0039	18.72	18.74	0.0035
$Fv_6$	15.74	15.51	0.0079	11.85	11.76	0.0073

L'écart type  $\sigma$  de chaque paramètre dans le tableau 3.4 est relativement petit par rapport à la valeur correspondante du paramètre estimé. Cela nous permet d'avoir confiance en la procédure d'estimation qu'on a appliqué. Le tableau 3.5 regroupe les valeurs numériques de  $\theta_p$  contenant les paramètres de base de la charge, ainsi que les variations des coefficients de frottement articulaire, toujours exprimées dans le système international d'unités (SI).



TABLE 3.5 – Valeurs numériques de  $\theta_p$  exprimées dans SI.

$\theta_p$	Valeur	$\theta_p$	Valeur	$\theta_p$	Valeur	$\theta_p$	Valeur
$\Delta F s_1$	3.11	$\Delta F v_3$	14.37	$x x r_p$	2.65	$m x_p$	0.43
$\Delta F v_1$	33.35	$\Delta F s_4$	1	$x y_p$	-0.20	$m y_p$	0.05
$\Delta F s_2$	4.48	$\Delta F v_4$	3.61	$x z_p$	0.43	$\Delta F s_6$	0.45
$\Delta F v_2$	22.26	$\Delta F s_5$	1.68	$y z_p$	0.05	$\Delta F v_6$	4.62
$\Delta F s_3$	2.17	$\Delta F v_5$	4.61	$z z_p$	0.15		

Le tableau 3.6 présente pour chaque articulation  $i$ , la masse virtuelle  $m_i$  et la valeur propre minimale de la matrice du tenseur d'inertie virtuel  ${}^i I_{g_i}$ , pour les deux cas, avec et sans charge. Le tableau illustre le fait que la condition (3.70) est satisfaite, ce qui permet de garantir que la matrice d'inertie déduite des paramètres estimés est bien définie positive.

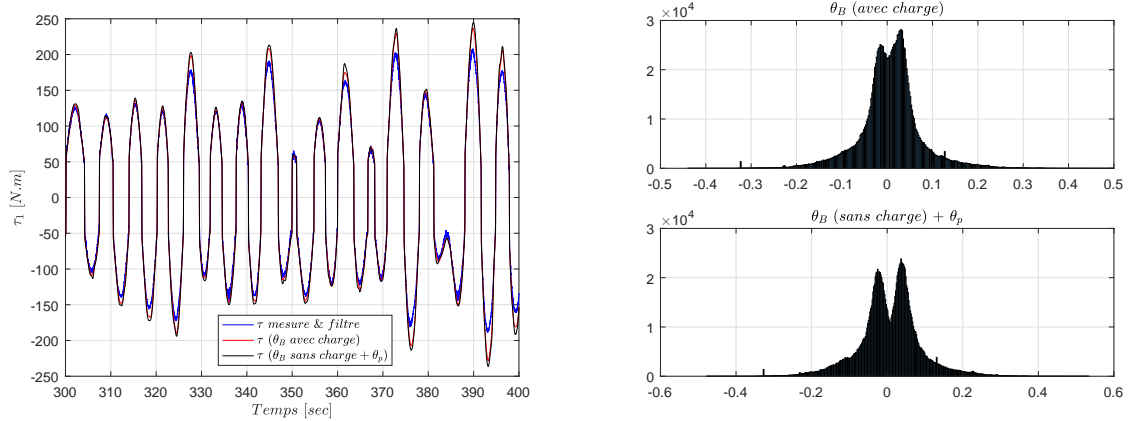
TABLE 3.6 – Vérification des contraintes dans (3.70).

Axe	$\lambda_{\min} \{ {}^i I_{g_i} \}$ (avec)	$m_i$ (avec)	$\lambda_{\min} \{ {}^i I_{g_i} \}$ (sans)	$m_i$ (sans)
1	0.01	6.22	0.01	76.90
2	0.01	2.07	0.01	76.72
3	0.01	17.99	0.01	11.63
4	0.01	20.95	0.01	34.66
5	0.01	1.93	0.01	2.44
6	0.01	38.45	0.01	8.74

### 3.6.6 Reconstruction des couples

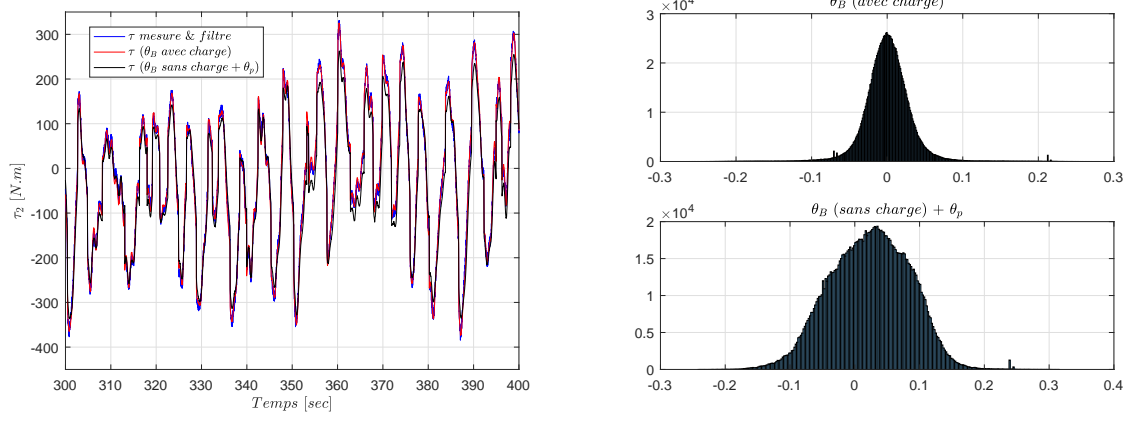
La reconstruction des couples est une étape qui permet de vérifier la validité des paramètres estimés et celle du modèle dynamique correspondant. Pour cela, une deuxième série de mesures, différente de celle utilisée pour l'identification, a été acquise pour le cas du robot avec sa charge. La deuxième série possède les mêmes caractéristiques que la première (signaux à excitation persistante, périodiques, prise en compte des limites articulaires, ...). Le schéma de commande en position utilisé possède les mêmes gains proportionnels mentionnés dans le tableau 3.2. Les signaux mesurés sont filtrés avec les mêmes paramètres de filtrage indiqués dans les sous-sections ci-dessus. Nous utilisons deux méthodes pour reconstruire les couples du robot avec sa charge. La première consiste à utiliser les paramètres calculés lors de l'identification dans le cas du robot avec charge. Pour la deuxième méthode, nous utilisons les paramètres estimés pour le cas du robot sans charge. Nous ajoutons

les variations des coefficients de frottement aux paramètres de frottement, et les paramètres de la charge à ceux de la 6<sup>ème</sup> articulation. Les figures 3.26a, 3.26c, 3.26e, 3.27a, 3.27c et 3.27e à gauche montrent pour les six axes un zoom entre les instants 300 et 400 *sec* sur les couples mesurés après filtrage (bleu), superposés avec ceux résultant des paramètres identifiés avec charge (rouge), et ceux de l'addition des paramètres sans charge plus les paramètres  $\theta_p$  (noir). Les figures à droite présentent l'histogramme de l'erreur relative entre les couples mesurés et calculés pour les deux cas.



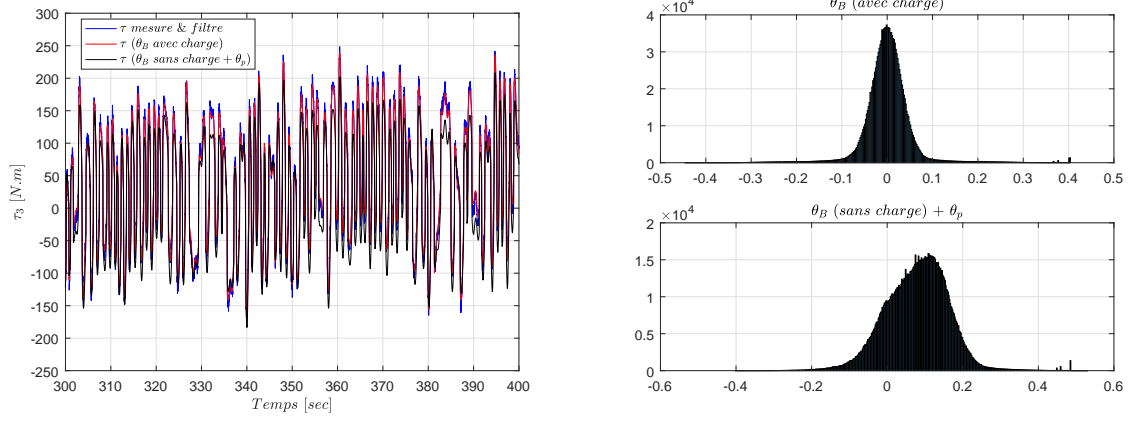
(a) Axe 1.

(b)



(c) Axe 2.

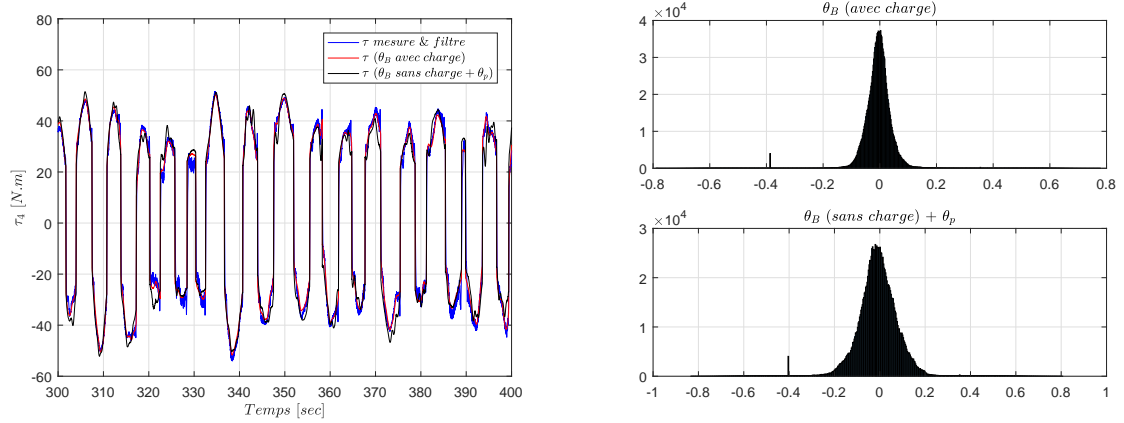
(d)



(e) Axe 3.

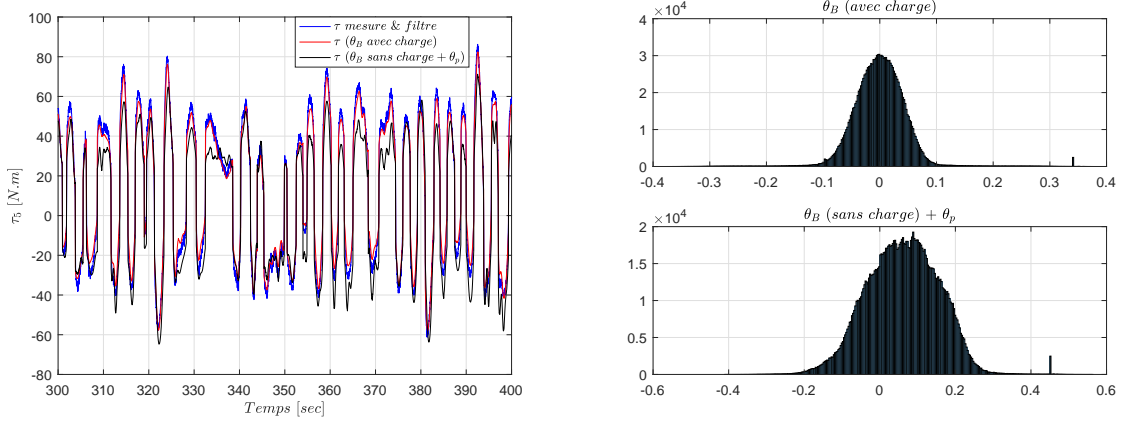
(f)

FIGURE 3.26 – Comparaison des couples mesurés après filtrage avec les couples calculés dans les deux cas (à gauche) ; Histogramme de l'erreur relative associée (à droite).



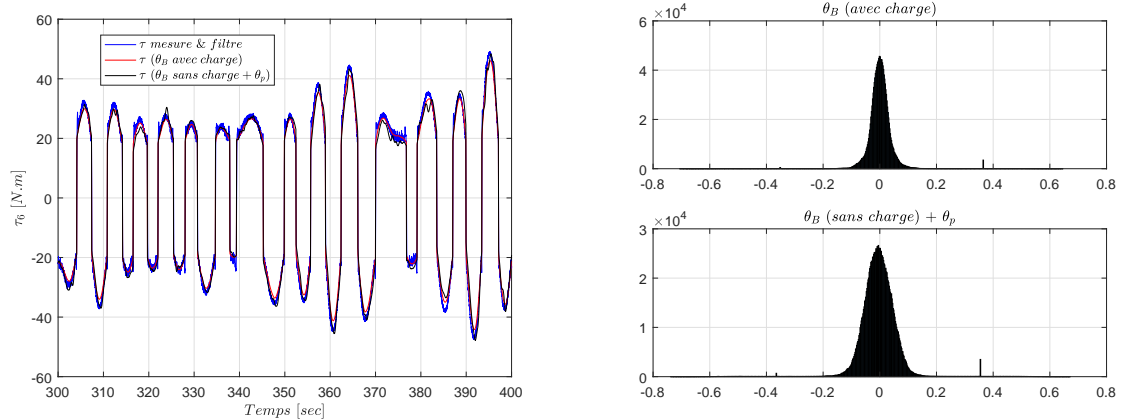
(a) Axe 4.

(b)



(c) Axe 5.

(d)



(e) Axe 6.

(f)

FIGURE 3.27 – Comparaison des couples mesurés après filtrage avec les couples calculés dans les deux cas (à gauche); Histogramme de l'erreur relative associée (à droite).

A travers ces figures, nous constatons une bonne qualité de reconstruction des couples, avec une erreur relative qui a la caractéristique d'un bruit gaussien. L'erreur relative est centrée autour de zéro. Elle demeure en général dans l'intervalle  $[-0.15, +0.15]$  sur l'axe 1 (erreur relative), que nous pouvons considérer comme une erreur relativement faible et acceptable. Il est évident que la reconstruction avec les paramètres identifiés dans le cas de ceux estimés quand la charge est fixée sur le robot, donne des résultats meilleurs par rapport au cas de la somme des paramètres sans charge avec  $\theta_p$ , à cause du nombre des sources d'erreurs. Pour ce cas, comme le vecteur des paramètres sans charge et  $\theta_p$  sont attachés par des erreurs d'estimation, l'addition des deux vecteurs amplifie l'erreur associée au vecteur résultant.

### 3.6.7 Paramètres virtuels associés aux paramètres de base

Le tableau 3.7 présente les valeurs numériques des paramètres virtuels trouvés par l'algorithme d'optimisation utilisé, pour le cas du robot avec sa charge utile. Les paramètres  $z_1, xx_{g_1}, xy_{g_1}, xz_{g_1}, yy_{g_1}, yz_{g_1}$  ne sont pas présentés dans le tableau, car ils n'interviennent pas dans la dynamique du robot, d'où le modèle dynamique n'est pas fonction de ces derniers.

TABLE 3.7 – Valeurs numériques des paramètres virtuels (exprimées en SI).

$\vartheta_i$	Valeur	$\vartheta_i$	Valeur	$\vartheta_i$	Valeur	$\vartheta_i$	Valeur	$\vartheta_i$	Valeur
$m_1$	6.22	$xz_{g_2}$	-0.84	$Fs_3$	54.78	$m_5$	1.93	$m_6$	38.45
$x_1$	0.00	$yz_{g_2}$	-0.94	$Fv_3$	87.92	$x_5$	0.06	$x_6$	0.00
$y_1$	-0.00	$Fs_2$	63.64	$m_4$	20.95	$y_5$	-0.02	$y_6$	-0.00
$zz_{g_1}$	0.01	$Fv_2$	185.75	$x_4$	0.01	$z_5$	-0.01	$z_6$	-0.01
$Fs_1$	49.74	$m_3$	17.99	$y_4$	0.01	$xx_{g_5}$	0.03	$xx_{g_6}$	0.02
$Fv_1$	150.81	$x_3$	-0.03	$z_4$	-0.14	$yy_{g_5}$	0.01	$yy_{g_6}$	0.09
$m_2$	2.07	$y_3$	-0.06	$xx_{g_4}$	1.81	$zz_{g_5}$	0.33	$zz_{g_6}$	0.00
$x_2$	0.97	$z_3$	0.04	$yy_{g_4}$	2.04	$xy_{g_5}$	0.00	$xy_{g_6}$	0.03
$y_2$	0.13	$xx_{g_3}$	5.43	$zz_{g_4}$	0.03	$xz_{g_5}$	-0.02	$xz_{g_6}$	0.02
$z_2$	0.08	$yy_{g_3}$	0.02	$xy_{g_4}$	0.19	$yz_{g_5}$	0.01	$yz_{g_6}$	0.00
$xx_{g_2}$	0.65	$zz_{g_3}$	0.01	$xz_{g_4}$	0.19	$Fs_5$	20.45	$Fs_6$	19.08
$yy_{g_2}$	1.21	$xy_{g_3}$	-0.29	$yz_{g_4}$	-0.08	$Fv_5$	29.29	$Fv_6$	15.74
$zz_{g_2}$	11.80	$xz_{g_3}$	0.03	$Fs_4$	24.63				
$xy_{g_2}$	-0.74	$yz_{g_3}$	-0.00	$Fv_4$	21.55				

Il est clair que ces paramètres virtuels ne correspondent pas aux paramètres réels. A titre d'exemple, les valeurs des masses  $m_3, m_4$  et  $m_6$  sont 17.99 kg, 20.95 kg et 38.45 kg respectivement, sachant que la masse totale du robot avec le capteur et l'outil est

de l'ordre de 16 kg. Des tentatives d'optimisation sur des intervalles plausibles pour chaque paramètre (par exemple  $1 \text{ kg} < m_3 < 3 \text{ kg}$ ) ont conduit à des paramètres qui donnent une mauvaise qualité de reconstruction des couples, notamment pour les axes 2, 3 et 5. Nous expliquons la non convergence vers les paramètres "réels" du robot, par l'existence de dynamiques qui ne sont pas prises en compte par le modèle que nous avons adopté.

Un algorithme d'optimisation basé sur la minimisation d'un critère mesurant l'écart des couples reconstruits avec ceux mesurés, tend à inclure les valeurs des paramètres des dynamiques non modélisées dans les paramètres existants, pour minimiser le critère le plus possible (sous les contraintes imposées, notamment pour avoir  $M(q) > 0$ ). Cela explique la bonne qualité de reconstruction des couples et la non plausibilité de certains paramètres.

### 3.6.8 Reconstruction des couples de Coriolis, de gravité et de frottement

Pour les commandes basées-modèle, et en particulier la commande en impédance introduite plus loin (chapitre 5), nous avons besoin de valider la reconstruction des couples générés par chaque terme du modèle (3.53), ou au moins du terme  $M(q)\ddot{q}$  séparé de  $(N(q, \dot{q})\dot{q} + G(q) - \tau_f)$  appelé *terme de Coriolis, gravité, frottement (Cgf)*, avant de procéder à la commande du système réel. Il est indispensable de garantir que ce terme se compense correctement avec le terme *Cgf* réel. Cette validation, en plus de celle de la sous-section 3.6.6, permettrait d'assurer la compensation de la matrice d'inertie réelle du système avec celle estimée, dans l'objectif de suivre la dynamique désirée par le robot.

D'abord, on désigne par  $\tau_m$  et  $\tau_c$  les couples mesurés filtrés et les couples calculés, respectivement. Les couples  $\tau_c$  sont calculés à l'aide du MDI du robot, les paramètres estimés et les positions, vitesses et accélérations articulaires filtrées résultant de l'application de  $\tau_m$ . La sous-section 3.6.6 nous a permis d'avoir une bonne qualité de reconstruction des couples :

$$\tau_c \approx \tau_m. \quad (3.88)$$

Nous désignons par  $M_m$ ,  $M_c$ ,  $Cgf_m$  et  $Cgf_c$  les matrices d'inertie réelle et calculée, et les termes de *Cgf* réel et calculé, respectivement. Nous pouvons ré-écrire l'équation (3.88) comme suit :

$$\tau_c = M_c\ddot{q} + Cgf_c \approx \tau_m = M_m\ddot{q} + Cgf_m. \quad (3.89)$$

L'acquisition d'une deuxième série de données caractérisées par des vitesses articulaires constantes, permettent d'avoir des couples articulaires résultant du terme *Cgf*, uniquement :

$$\dot{q} = \text{constante} \Rightarrow \ddot{q} = 0 \Rightarrow \begin{cases} \tau_c = Cgf_c \\ \tau_m = Cgf_m \end{cases}. \quad (3.90)$$

Si la reconstruction des couples dans ce cas est de bonne qualité aussi ( $Cgf_c \approx Cgf_m$ ),

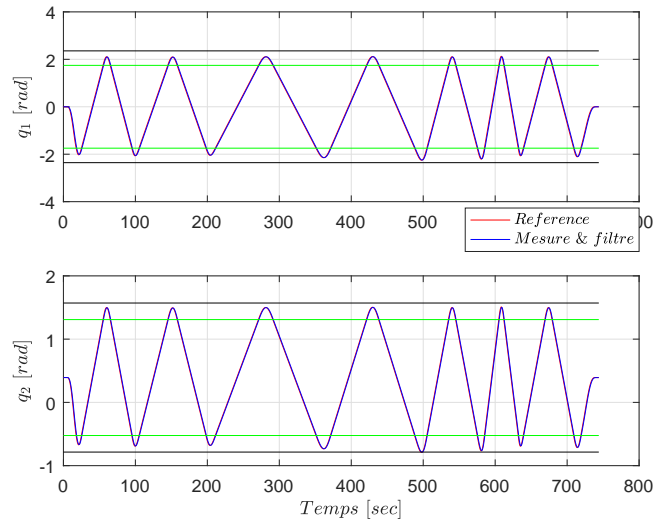
et comme tous les paramètres identifiés qui existent dans la matrice d'inertie, existent aussi dans  $Cgf$ , nous en concluons que la matrice  $M_c$  tend à compenser la matrice d'inertie réelle :

$$\left. \begin{array}{l} M_c \ddot{q} + Cgf_c \approx M_m \ddot{q} + Cgf_m \\ Cgf_c \approx Cgf_m \end{array} \right\} \Rightarrow M_c \approx M_m. \quad (3.91)$$

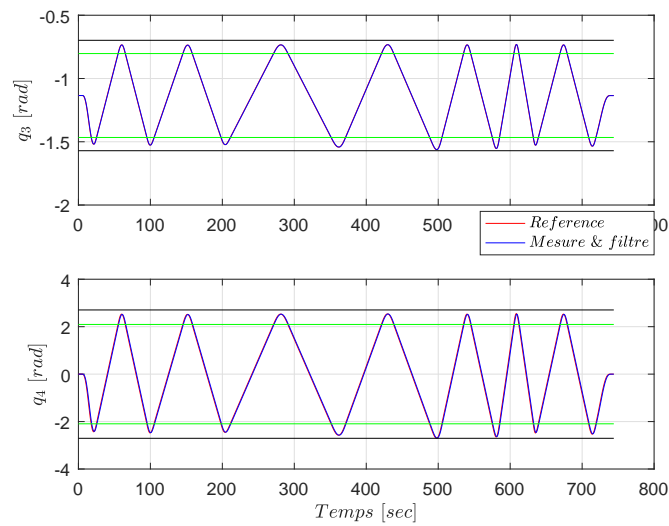
Sur le plan pratique, nous avons généré une série de signaux d'excitation en position qui est caractérisée par des vitesses constantes sur la plupart des morceaux de la trajectoire, sauf pendant des phases de raccordement, tout en respectant les positions et vitesses articulaires limites. Ces signaux d'excitation en position (mesurés filtrés et de référence) sont représentés sur les figures 3.28a, 3.28b et 3.28c. Les lignes vertes délimitent les intervalles où le signal de position est à pentes constantes, tandis que les lignes noires représentent les limites articulaires à ne pas dépasser. Les figures 3.29a, 3.29c, 3.29e, 3.29b, 3.29d et 3.29f représentent les vitesses et accélérations articulaires correspondantes (calculées filtrées et de référence) aux signaux d'excitation en position qui sont appliqués.

Les courbes des figures 3.30a, 3.30c et 3.30e représentent, pour chaque axe, les couples mesurés après filtrage (bleu), superposés avec ceux estimés en utilisant le modèle dynamique du robot, et ses paramètres identifiés. Les courbes des figures 3.30b, 3.30d et 3.30f représentent les histogrammes de l'erreur relative associée aux couples estimés.

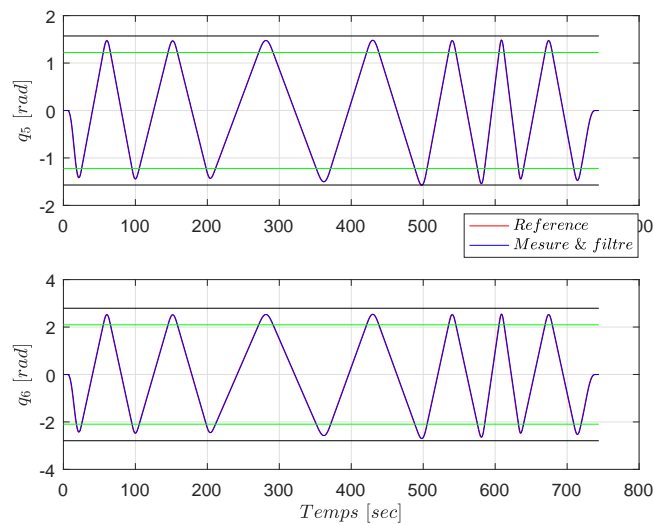
Nous pouvons remarquer que, pour des signaux d'excitation contenant des morceaux à vitesses articulaires constantes, l'erreur relative n'est pas centrée, et n'a pas du tout la caractéristique d'un bruit gaussien. Malgré tout, elle reste relativement faible, et la reconstruction des couples est tout de même de bonne qualité. Ces résultats nous permettent de constater une bonne estimation du terme  $Cgf$ . En plus des résultats obtenus dans la sous-section précédente 3.6.6, nous pouvons conclure une bonne prédiction de la matrice d'inertie  $M_c$  (équation (3.91)), et donc  $M_c$  tend à compenser la matrice d'inertie réelle lors d'un schéma de commande basée-modèle.



(a) Axes 1 &amp; 2.



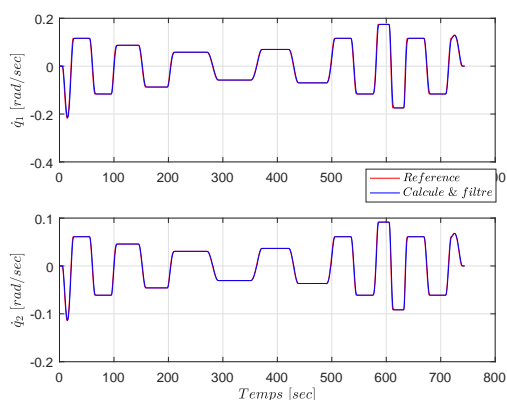
(b) Axes 3 &amp; 4.



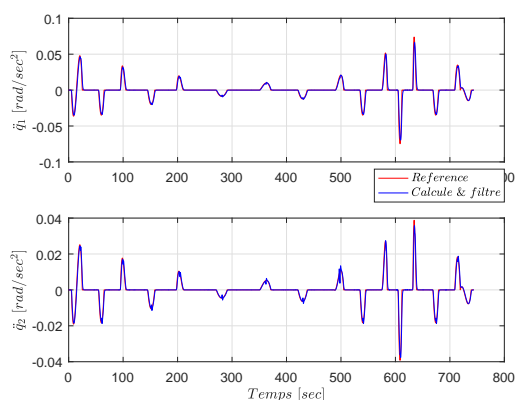
(c) Axes 5 &amp; 6.

FIGURE 3.28 – Signaux d'excitation en position pour des morceaux de trajectoires articulaires à vitesse constante (accélération nulle). Les lignes rouges indiquent les limites articulaires. Les lignes vertes délimitent le mouvement articulaires à vitesse constante.

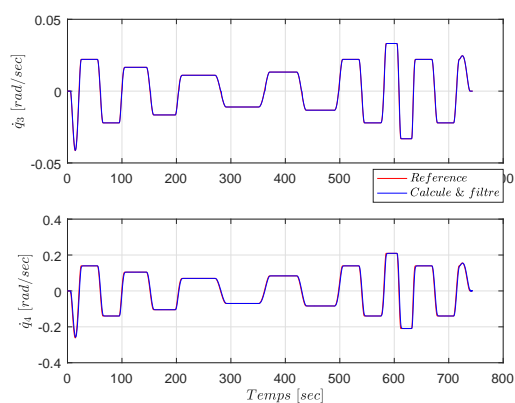




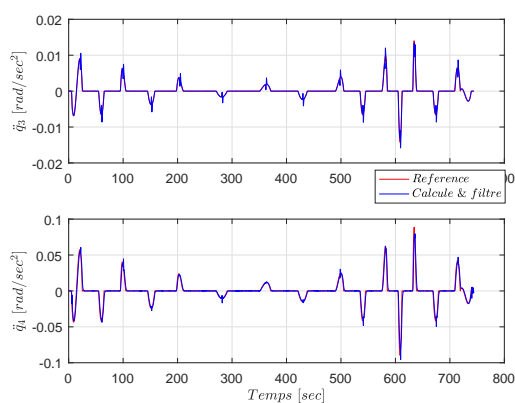
(a) Axes 1 &amp; 2



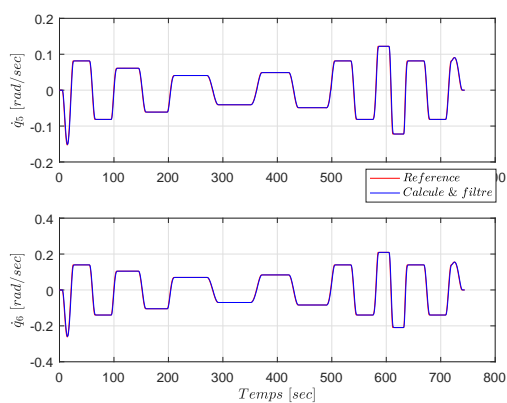
(b) Axes 1 &amp; 2



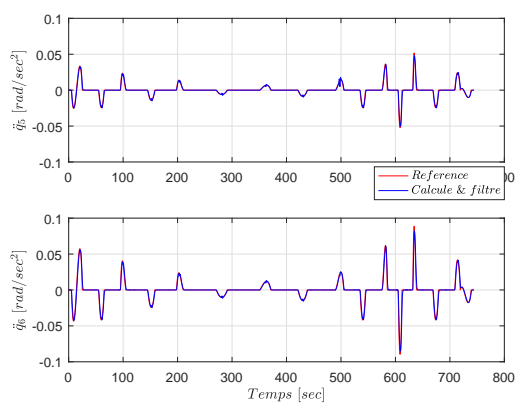
(c) Axes 3 &amp; 4



(d) Axes 3 &amp; 4

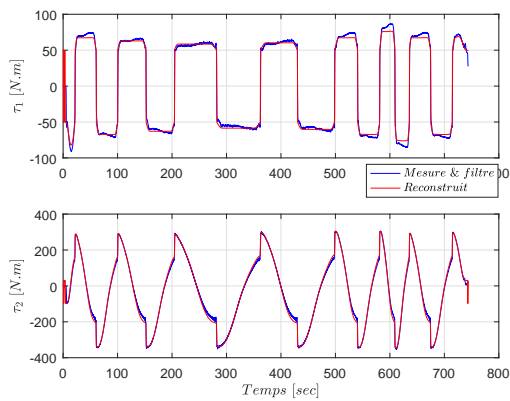


(e) Axes 5 &amp; 6

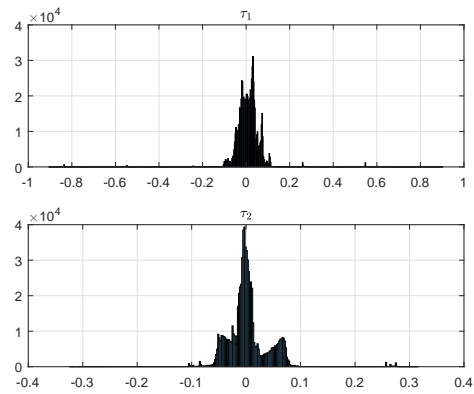


(f) Axes 5 &amp; 6

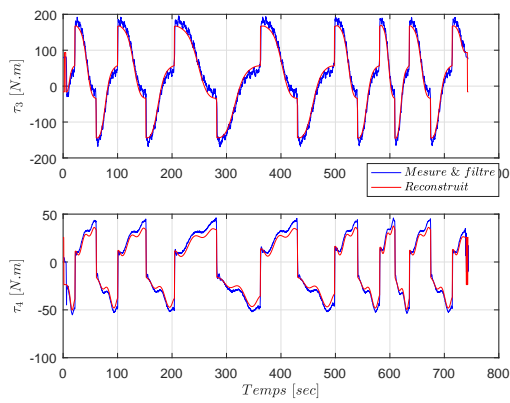
FIGURE 3.29 – Vitesses et accélérations de référence correspondant aux positions de références de la figure 3.28.



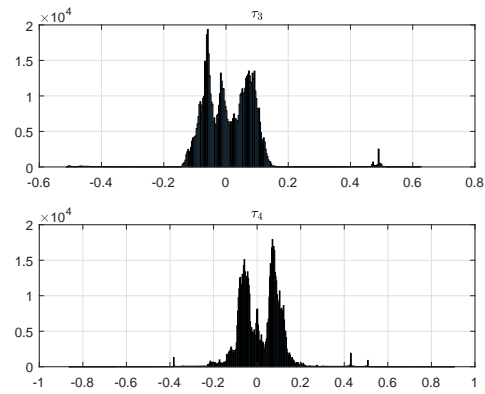
(a) Axes 1 &amp; 2.



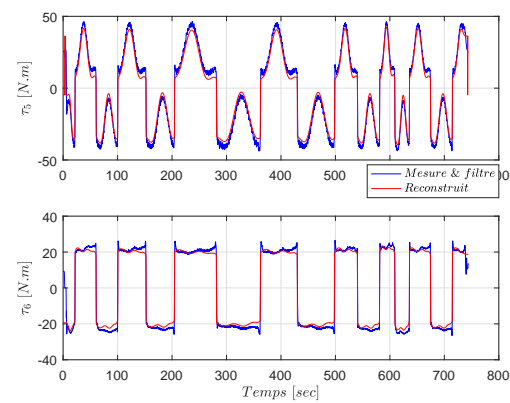
(b) Axes 1 &amp; 2



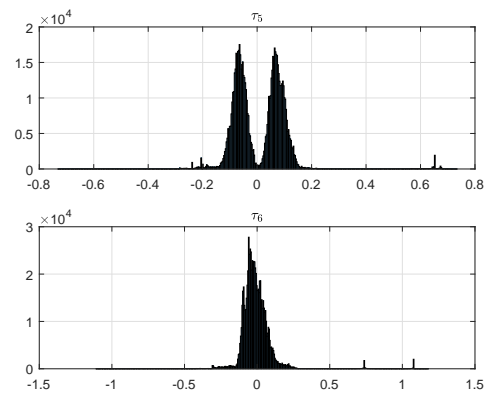
(c) Axes 3 &amp; 4.



(d) Axes 3 &amp; 4



(e) Axes 5 &amp; 6.



(f) Axes 5 &amp; 6

FIGURE 3.30 – Comparaison des couples mesurés après filtrage (bleu) avec ceux calculés (rouge) à gauche ; Histogramme de l'erreur relative associée à droite.

## 3.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la modélisation et à l'identification des paramètres dynamiques des robots manipulateurs. Pour la modélisation, nous avons utilisé le formalisme d'Euler-Lagrange pour calculer le modèle dynamique. Après la présentation du robot **Denso VP-6242G** avec son module de contrôle, nous avons utilisé le logiciel **OpenSYMORO** pour calculer son modèle dynamique. **OpenSYMORO** nous a permis de réduire le temps dédié à la partie modélisation, grâce à sa puissance de calcul symbolique.

Pour l'identification, nous avons discuté le modèle dynamique destiné à l'identification, et le regroupement des paramètres dynamiques à identifier. Nous avons expliqué également les caractéristiques des signaux utilisés dans ce cadre, qui sont des signaux à excitation persistante. Le traitement des signaux mesurés pour qu'ils deviennent exploitables, était aussi discuté. La technique d'optimisation utilisée repose sur un critère quadratique (moindres carrés pondérés) sous contraintes. Les contraintes imposées sur les paramètres dits physiques permettent de garantir que la matrice d'inertie est définie positive peu importe la configuration du robot.

Nous avons étudié le cas du robot **Denso VP-6242G** sur lequel nous travaillons au L2S. La méthode d'identification nous a permis d'avoir un vecteur des paramètres qui garantit une bonne qualité de reconstruction des couples lorsque les vitesses articulaires sont constantes, ou variables au cours du temps. Cela nous a permis de conclure que la matrice d'inertie calculée est approximativement égale à la matrice d'inertie réelle. Par conséquent, on peut se reposer sur le modèle obtenu pour implémenter sur le robot réel, une commande à base du modèle identifié.

# Génération de trajectoires en temps-réel

## Sommaire

---

4.1	Introduction	83
4.2	La génération de trajectoire	84
4.3	Générateur de trajectoire pour la co-manipulation	90
4.4	Tests de validation	108
4.5	Conclusion	126

---

## 4.1 Introduction

En robotique, un générateur de trajectoire permet de produire des consignes articulaires ou cartésiennes en position, vitesse et accélération pour un robot. Ces consignes se traduisent par une suite de configurations articulaires ou de situations désirées de l'organe terminal, à savoir sa position ou son orientation.

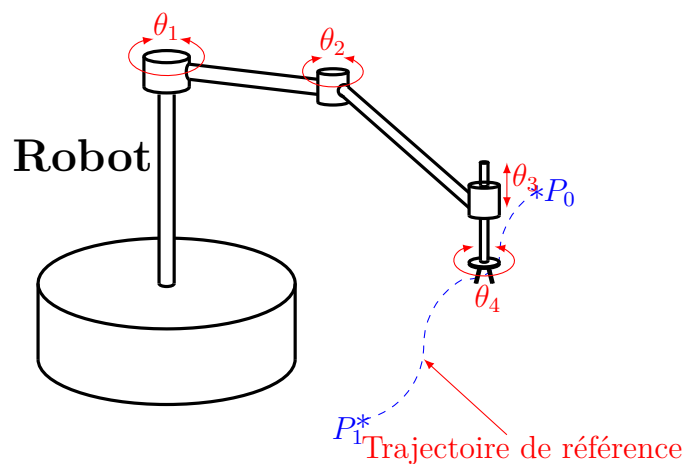


FIGURE 4.1 – Exemple d'une trajectoire de référence à suivre par un manipulateur.

La génération de trajectoire peut se faire dans l'espace articulaire pour définir la trajectoire de référence de chaque articulation, ou dans l'espace cartésien afin de

déterminer la trajectoire de l'outil. Suivant les spécifications du cahier des charges, les trajectoires désirées pour un robot peuvent être générées *hors-ligne*, c'est-à-dire avant l'exécution de la tâche, ou bien *en ligne*, c'est-à-dire au cours de cette exécution. Ce dernier cas correspond aux générateurs de trajectoire en temps réel.

Notre étude s'inscrit dans le cadre de la co-manipulation homme-robot pour la manutention des charges lourdes, dans un espace caractérisé par des obstacles dynamiques au cours du temps, c'est-à-dire que leur position n'est pas fixe. Pour cette raison, c'est à l'opérateur humain de décider de la position que le robot doit suivre. Dans [42], les auteurs se sont intéressés à la conception d'un générateur des trajectoires planes **en temps réel**, qui permettent de traduire l'intention du mouvement de l'opérateur humain en une trajectoire de référence pour le robot. Cependant, lorsque des limites virtuelles sont imposées au mouvement du robot pour différentes raisons, il est difficile pour un opérateur humain guidant le robot de les prendre en compte, puisqu'elles ne sont pas visibles en général. Par conséquent, le générateur doit produire des trajectoires conformes à l'intention de mouvement de l'opérateur, tout en respectant les limites de la région de travail imposées au robot. Dans ce chapitre, nous nous intéressons à l'extension de la génération de trajectoire du plan vers l'espace, et à la proposition d'une méthode pour contraindre le mouvement du robot aux limites de l'espace de travail, avec une co-manipulation la plus transparente possible.

Le chapitre est organisé comme suit. Nous commencerons d'abord par la présentation de quelques généralités sur la génération de trajectoires ainsi que les différents types existants. Ensuite, nous présenterons le générateur de trajectoire proposé pour répondre aux contraintes expliquées ci-dessus.

## 4.2 La génération de trajectoire

### 4.2.1 Génération de trajectoire dans l'espace articulaire

C'est une méthode facile à implémenter, qui consiste à générer des trajectoires articulaires de référence sous les contraintes de butée articulaire et vitesse maximale définies pour chaque articulation [73]. Elle permet d'éviter les problèmes de singularités. Cependant, la trajectoire générée dans l'espace opérationnel est libre, ce qui présente des risques de collisions avec l'environnement (opérateurs, autres robots, pièces, ...). Cette méthode de génération de trajectoire ne convient donc pas pour certaines tâches.

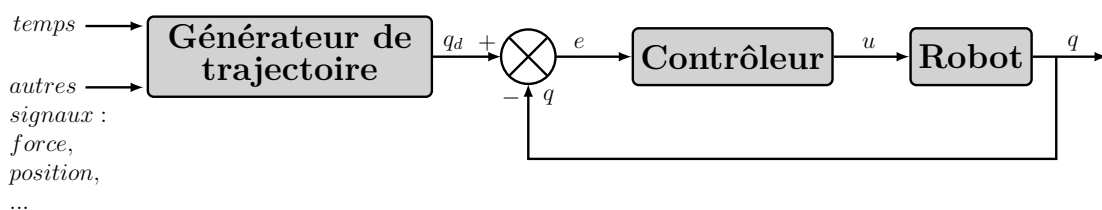


FIGURE 4.2 – Génération de trajectoire en espace articulaire

La génération de trajectoire dans l'espace articulaire est utilisée en général pour générer les consignes des signaux d'excitation lors de l'identification des paramètres d'un robot [20, 19]. Pour cela, pour chaque articulation, une trajectoire de référence en position, vitesse et accélération est générée, selon les contraintes des limites articulaire en position et en vitesse (ou en couple).

### 4.2.2 Génération de trajectoire dans l'espace opérationnel

La génération de trajectoire dans l'espace opérationnel consiste à produire la trajectoire de référence en position et en orientation de l'outil (figure 4.3). Cela permet de contrôler la position et la vitesse de l'outil lors de la réalisation d'une tâche. Cependant, les positions et vitesses articulaires qui résultent doivent répondre aux contraintes des butées articulaires et vitesses maximales de chaque moteur [73].

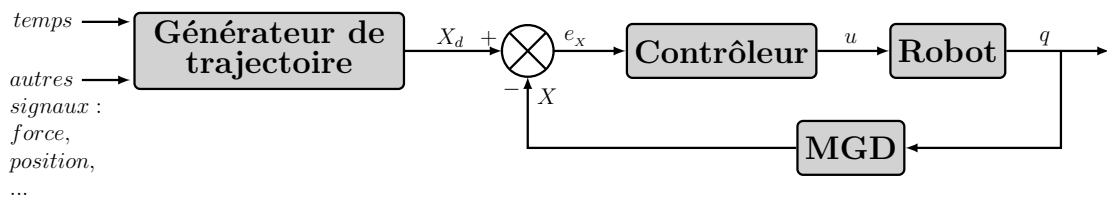


FIGURE 4.3 – Génération de trajectoire en espace opérationnel

De plus, les trajectoires articulaires associées doivent éviter les singularités du robot, et le calcul du modèle géométrique inverse (MGI dans la figure 4.4) du robot est nécessaire pour certains types de contrôleurs (commandes dans l'espace articulaire). La figure 4.4 illustre un générateur de trajectoire en coordonnées cartésiennes inséré dans une boucle de commande élaborée dans l'espace articulaire.

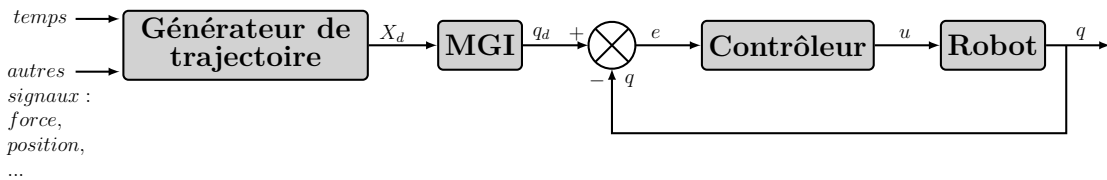


FIGURE 4.4 – Génération de trajectoire en espace opérationnel

La génération de trajectoire en espace opérationnel est utilisée pour la réalisation de toutes les tâches qui nécessitent l'intervention de l'outil monté sur le poignet du manipulateur, comme le soudage, le transport des pièces, l'assemblage, le découpage, l'emballage, ...

### 4.2.3 Types des trajectoires générées

Nous pouvons distinguer deux types de trajectoires à générer, que nous allons décrire dans les prochains paragraphes.

#### Trajectoires point-à-point

Les trajectoires point-à-point sont caractérisées par des contraintes imposées sur les points de départ et d'arrivée uniquement [74]. Le robot part de son point de départ

et arrive au point d'arrivée avec des vitesses nulles. Le mouvement entre ces deux points n'est pas soumis à des contraintes. Ce type de trajectoire est utilisé pour des tâches dans un environnement libre, comme par exemple dans le cas du soudage point-à-point (figure 4.5).

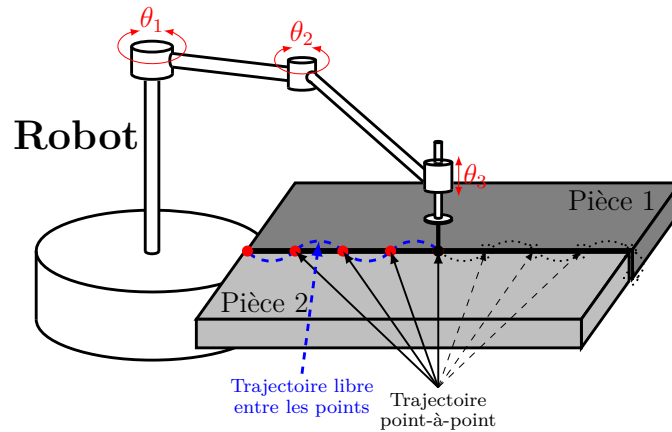


FIGURE 4.5 – Exemple d'une trajectoire point-à-point pour une opération de soudage.

Dans l'exemple de la figure 4.5, la vitesse du robot à chaque point de soudure est nulle. Son mouvement entre les points concernés par la soudure est libre dans le plan horizontal.

### Trajectoires à points intermédiaires

Les trajectoires à points intermédiaires sont caractérisées par des contraintes sur les points de départ, d'arrivée et des points intermédiaires. Les vitesses d'arrivée à un point intermédiaire et de départ de ce même point sont égales et non nulles [74] (courbe de vitesse lisse au niveau des points intermédiaires). Ce type de trajectoires est utilisé notamment pour l'évitement d'obstacles, ou bien pour différentes opérations de découpage (figure 4.6).

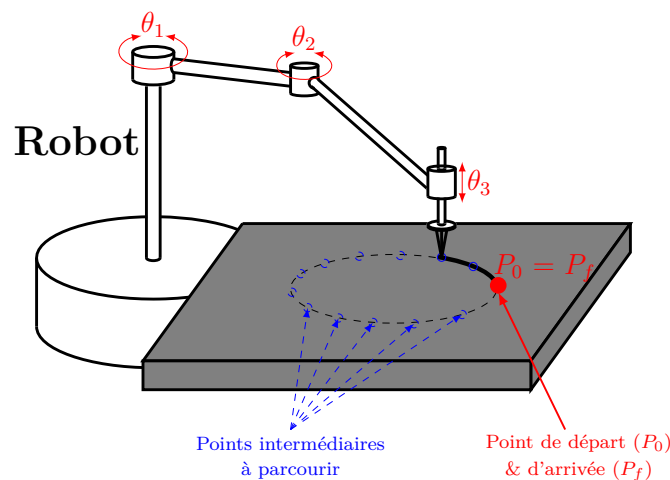


FIGURE 4.6 – Exemple d'une trajectoire à points intermédiaires pour une opération d'oxycoupage.

Dans l'exemple de la figure 4.6, le robot doit parcourir les points bleus avec des vitesses non nulles. Afin d'avoir une découpe précise, il faut que les points intermédiaires soient proches les uns avec les autres.

#### 4.2.4 Génération de trajectoire par interpolation

Dans cette section, nous allons nous intéresser à la génération de trajectoire par interpolation, qui peut être envisagée dans l'espace articulaire ou cartésien. La génération de trajectoire par interpolation peut être utilisée pour générer des trajectoires point-à-point ou à points intermédiaires, selon les contraintes imposées.

##### La spline cubique

Une spline cubique est définie par un polynôme de degré 3. Elle permet d'imposer des contraintes sur la position  $x(t)$  et la vitesse  $\dot{x}(t)$  des points de départ ( $P_0$ ) et d'arrivée ( $P_f$ ) :

$$\begin{cases} x(t = t_0) = x_0, & x(t = t_f) = x_f, \\ \dot{x}(t = t_0) = \dot{x}_0, & \dot{x}(t = t_f) = \dot{x}_f, \end{cases} \quad (4.1)$$

L'expression polynômiale de la position permettant de satisfaire ces contraintes est la suivante :

$$x(t) = a_3(t - t_0)^3 + a_2(t - t_0)^2 + a_1(t - t_0) + a_0. \quad (4.2)$$

Par dérivation, on calcule l'expression de la vitesse :

$$\dot{x}(t) = 3a_3(t - t_0)^2 + 2a_2(t - t_0) + a_1. \quad (4.3)$$

Nous utilisons alors les contraintes (4.1) et les expressions (4.2) et (4.3) afin de calculer les coefficients du polynôme  $x(t)$  :

$$\begin{cases} x_0 = a_3(t_0 - t_0)^3 + a_2(t_0 - t_0)^2 + a_1(t_0 - t_0) + a_0, \\ x_f = a_3(t_f - t_0)^3 + a_2(t_f - t_0)^2 + a_1(t_f - t_0) + a_0, \\ \dot{x}_0 = 3a_3(t_0 - t_0)^2 + 2a_2(t_0 - t_0) + a_1, \\ \dot{x}_f = 3a_3(t_f - t_0)^2 + 2a_2(t_f - t_0) + a_1. \end{cases} \quad (4.4)$$

Cela permet d'aboutir à :

$$\begin{cases} x_0 = a_0, \\ x_f = a_3(t_f - t_0)^3 + a_2(t_f - t_0)^2 + a_1(t_f - t_0) + a_0, \\ \dot{x}_0 = a_1, \\ \dot{x}_f = 3a_3(t_f - t_0)^2 + 2a_2(t_f - t_0) + a_1. \end{cases} \quad (4.5)$$

Le système (4.5) est un système d'équations linéaires à quatre inconnues, qui admet la solution suivante :



$$\begin{cases} a_3 = \frac{(\dot{x}_f + \dot{x}_0)}{(t_f - t_0)^2} - 2 \frac{(x_f - x_0)}{(t_f - t_0)^3}, \\ a_2 = -\frac{(\dot{x}_f + 2\dot{x}_0)}{(t_f - t_0)} + 3 \frac{(x_f - x_0)}{(t_f - t_0)^2}, \\ a_1 = \dot{x}_0, \\ a_0 = x_0. \end{cases} \quad (4.6)$$

La figure 4.7 présente l'allure des courbes de position, vitesse et accélération correspondant à la spline cubique pour :  $t_0 = 0 \text{ sec}$ ,  $t_f = 4 \text{ sec}$ ,  $x_0 = 0 \text{ m}$ ,  $x_f = 10 \text{ m}$ ,  $\dot{x}_0 = 0 \text{ m/sec}$ ,  $\dot{x}_f = 0 \text{ m/sec}$ .

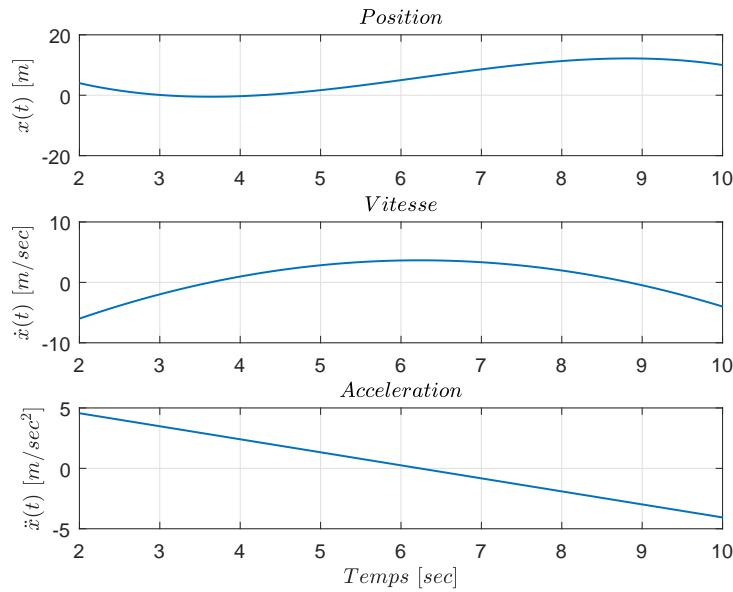


FIGURE 4.7 – Trajectoires en position, vitesse et accélération obtenues avec une spline cubique.

A partir de la figure 4.7, nous pouvons remarquer qu'une répétition de la spline pour un déplacement supplémentaire vers la position  $20 \text{ m}$  par exemple, va engendrer une discontinuité sur le tracé de l'accélération. Aussi, cette dernière n'est pas contrôlée puisque nous n'avons pas imposé de contraintes sur l'accélération, ce qui peut endommager les actionneurs du robot.

### La spline quintique

La spline quintique permet d'imposer des contraintes sur la position  $x(t)$ , la vitesse  $\dot{x}(t)$  et l'accélération  $\ddot{x}(t)$  des points de départ ( $P_0$ ) et d'arrivée ( $P_f$ ) :

$$\begin{cases} x(t = t_0) = x_0, & x(t = t_f) = x_f, \\ \dot{x}(t = t_0) = \dot{x}_0, & \dot{x}(t = t_f) = \dot{x}_f, \\ \ddot{x}(t = t_0) = \ddot{x}_0, & \ddot{x}(t = t_f) = \ddot{x}_f. \end{cases} \quad (4.7)$$

Le polynôme de plus faible degré permettant de satisfaire ces contraintes est de degré 5, dont l'expression de la position est donnée par :

$$x(t) = a_5(t - t_0)^5 + a_4(t - t_0)^4 + a_3(t - t_0)^3 + a_2(t - t_0)^2 + a_1(t - t_0) + a_0. \quad (4.8)$$

Par dérivation, nous en déduisons les expressions de la vitesse et de l'accélération :

$$\begin{cases} \dot{x}(t) = 5a_5(t - t_0)^4 + 4a_4(t - t_0)^3 + 3a_3(t - t_0)^2 + 2a_2(t - t_0) + a_1, \\ \ddot{x}(t) = 20a_5(t - t_0)^3 + 12a_4(t - t_0)^2 + 6a_3(t - t_0) + 2a_2. \end{cases} \quad (4.9)$$

En utilisant les contraintes (4.7) ainsi que les expressions (4.8) et (4.9), nous pouvons en déduire les expressions des coefficients du polynôme  $x(t)$  :

$$\begin{cases} x_0 = a_5(t_0 - t_0)^5 + a_4(t_0 - t_0)^4 + a_3(t_0 - t_0)^3 + a_2(t_0 - t_0)^2 + a_1(t_0 - t_0) + a_0, \\ x_f = a_5(t_f - t_0)^5 + a_4(t_f - t_0)^4 + a_3(t_f - t_0)^3 + a_2(t_f - t_0)^2 + a_1(t_f - t_0) + a_0, \\ \dot{x}_0 = 5a_5(t_0 - t_0)^4 + 4a_4(t_0 - t_0)^3 + 3a_3(t_0 - t_0)^2 + 2a_2(t_0 - t_0) + a_1, \\ \dot{x}_f = 5a_5(t_f - t_0)^4 + 4a_4(t_f - t_0)^3 + 3a_3(t_f - t_0)^2 + 2a_2(t_f - t_0) + a_1, \\ \ddot{x}_0 = 20a_5(t_0 - t_0)^3 + 12a_4(t_0 - t_0)^2 + 6a_3(t_0 - t_0) + 2a_2, \\ \ddot{x}_f = 20a_5(t_f - t_0)^3 + 12a_4(t_f - t_0)^2 + 6a_3(t_f - t_0) + 2a_2. \end{cases} \quad (4.10)$$

En simplifiant, il vient :

$$\begin{cases} x_0 = a_0, \\ x_f = a_5(t_f - t_0)^5 + a_4(t_f - t_0)^4 + a_3(t_f - t_0)^3 + a_2(t_f - t_0)^2 + a_1(t_f - t_0) + a_0, \\ \dot{x}_0 = a_1, \\ \dot{x}_f = 5a_5(t_f - t_0)^4 + 4a_4(t_f - t_0)^3 + 3a_3(t_f - t_0)^2 + 2a_2(t_f - t_0) + a_1, \\ \ddot{x}_0 = 2a_2, \\ \ddot{x}_f = 20a_5(t_f - t_0)^3 + 12a_4(t_f - t_0)^2 + 6a_3(t_f - t_0) + 2a_2. \end{cases} \quad (4.11)$$

Le système (4.11) est un système d'équations linéaires à six inconnus, qui admet la solution suivante :

$$\begin{cases} a_5 = 6 \frac{x_f - x_0}{(t_f - t_0)^5} - 3 \frac{\dot{x}_f + \dot{x}_0}{(t_f - t_0)^4} + \frac{1}{2} \frac{\ddot{x}_f - \ddot{x}_0}{(t_f - t_0)^3}, \\ a_4 = -15 \frac{x_f - x_0}{(t_f - t_0)^4} + \frac{7\dot{x}_f + 8\dot{x}_0}{(t_f - t_0)^3} + \frac{1}{2} \frac{3\ddot{x}_0 - 2\ddot{x}_f}{(t_f - t_0)^2}, \\ a_3 = 10 \frac{x_f - x_0}{(t_f - t_0)^3} - 2 \frac{2\dot{x}_f + 3\dot{x}_0}{(t_f - t_0)^2} + \frac{1}{2} \frac{\ddot{x}_f - 3\ddot{x}_0}{(t_f - t_0)}, \\ a_2 = \frac{\ddot{x}_0}{2}, \\ a_1 = \dot{x}_0, \\ a_0 = x_0. \end{cases} \quad (4.12)$$

La figure 4.8 présente l'allure des courbes de position, vitesse, accélération et jerk correspondants à la spline quintique pour :  $t_0 = 0 \text{ sec}$ ,  $t_f = 4 \text{ sec}$ ,  $x_0 = 0 \text{ m}$ ,  $x_f = 10 \text{ m}$ ,  $\dot{x}_0 = 0 \text{ m/sec}$ ,  $\dot{x}_f = 0 \text{ m/sec}$ ,  $\ddot{x}_0 = 0 \text{ m/sec}^2$ ,  $\ddot{x}_f = 0 \text{ m/sec}^2$ .

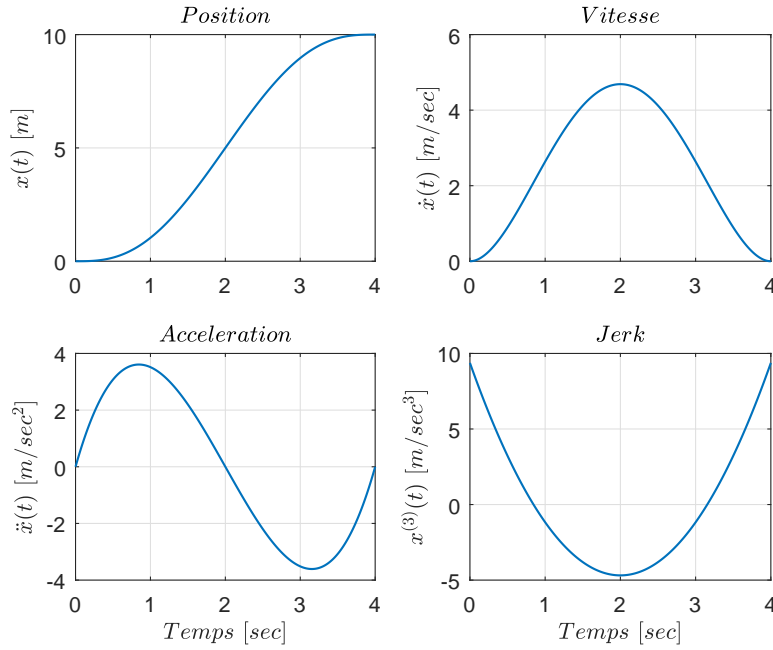


FIGURE 4.8 – Trajectoires de position, vitesse, accélération et jerk obtenus avec une spline quintique.

Les splines cubique et quintique peuvent être utilisées pour la génération de trajectoires de type point-à-point ou bien à points intermédiaires. Dans le dernier cas, les points de départ et d'arrivée sont, à chaque fois, pris comme deux points successifs : point de départ et premier point intermédiaire, deux points intermédiaires successifs, dernier point intermédiaire et point d'arrivée.

## 4.3 Générateur de trajectoire pour la co-manipulation

### 4.3.1 Formulation du problème de la co-manipulation

Notre étude s'inscrit dans le cadre de la manutention des charges lourdes (compensation du poids), transportées par un robot, et dirigées/co-manipulées par un opérateur humain. Le robot et l'opérateur sont en contact direct par le biais d'une charge correspondant à un solide rigide. L'étude présente une continuité des travaux présentés dans [42, 45, 44, 43]. Dans ces travaux, les auteurs ont développé un générateur de trajectoire pour le mouvement du robot modulé par la force exercée par l'opérateur. Par le biais de ce générateur, disposé dans un schéma de commande incluant un bouclage des mesures de position articulaires et de vitesses articulaires calculées, sur les couples de commande, ils ont voulu assurer :

1. ***l'assistance du robot pendant la co-manipulation*** : de telle sorte que l'opérateur n'applique qu'un effort minimal au niveau du point d'interaction avec le robot. Ce dernier s'occupe, à travers ses actionneurs, de fournir l'effort nécessaire pour compenser et/ou déplacer la charge. Ces efforts sont calculés via le schéma de commande qui sera proposé dans le chapitre suivant ;

2. **la transparence de la co-manipulation** : de sorte que le robot se laisse dirigé selon le désir de l'opérateur. Le concept de la transparence entre l'opérateur et le robot se traduit à une relation de colinéarité entre la vitesse désirée  $\dot{X}_d \in \mathbb{R}^3$  et l'effort appliqué par l'opérateur  $F_h \in \mathbb{R}^3$ , donnée par la relation suivante :

$$\dot{X}_d \times F_h = 0, \quad (4.13a)$$

$$\dot{X}_d \bullet F_h > 0. \quad (4.13b)$$

La relation (4.13a) est vérifiée si et seulement si, les vecteurs de la vitesse et la force sont colinéaires. D'où, l'angle entre les deux vecteurs est, soit nulle, soit  $\pi$  :

$$\dot{X}_d \times F_h = 0 \Rightarrow \widehat{\dot{X}_d, F_h} = \begin{cases} 0 \\ \text{ou} \\ \pi \end{cases}. \quad (4.14)$$

Si cette relation est vérifiée et de plus, si le produit scalaire entre  $\dot{X}_d$  et  $F_h$  est positif (relation (4.13b)), l'angle entre les vecteurs  $\dot{X}_d$  et  $F_h$  est donc nul. Ces relations assurent que les deux vecteurs ont la même direction. Autrement dit, la vitesse désirée du robot est générée de manière à suivre l'intention de l'opérateur.

3. **la sécurité de l'opérateur** : qui présente l'exigence la plus importante pendant une co-manipulation homme-robot. Le robot et la charge co-manipulée (ou transportée) ne doivent en aucun cas présenter un danger sur la vie de l'opérateur ou sur son intégrité physique. Pour cela, il est nécessaire d'assurer la stabilité du schéma de commande proposé, et de limiter la vitesse du robot à une valeur compatible avec la norme en vigueur [14]. Les auteurs ont constaté qu'une relation proportionnelle entre la force et la vitesse pouvait être dangereuse pour l'opérateur.

L'étude menée dans [42] a visé la co-manipulation dans l'espace bidimensionnel. Dans cette étude, nous cherchons à étendre le travail du plan vers l'espace tridimensionnel, qui présente un cas plus général. Nous proposons la conception d'un générateur modifié (amélioré) à la base du générateur conçu dans [42]. La figure 4.9 illustre le diagramme du générateur proposé.

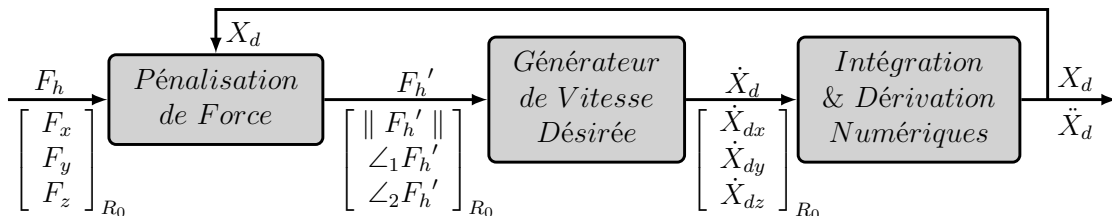


FIGURE 4.9 – Générateur de trajectoire en temps réel.

Comme il est présenté sur la figure 4.9, le générateur reçoit comme entrée la force appliquée par l'opérateur  $F_h$ , qui est estimée en exploitant les mesures de la force

d'interaction entre ce dernier et le robot. La méthode de récupération des mesures, et d'estimation de la force de l'opérateur seront détaillées dans les deux sous-sections suivantes 4.3.2 et 4.3.3, respectivement.

Aussi, nous nous intéressons à la prise en compte, par le générateur de trajectoire, des limites virtuelles prédéfinies de l'espace de co-manipulation. Pour cela, nous allons proposer une pénalisation de la force de l'opérateur, présentée par le bloc *Pénalisation de Force* sur la figure 4.9, afin de générer une image  $F_h'$  de  $F_h$  qui sera l'entrée du bloc *Générateur de Vitesse Désirée*. Ce bloc, constitué essentiellement d'un automate à quatre états, permet de générer la vitesse désirée  $\dot{X}_d$  du point d'interaction. La position et l'accélération désirées  $X_d$  et  $\ddot{X}_d$  sont calculées par intégration et dérivation numériques de  $\dot{X}_d$ , respectivement. Suite à la pénalisation de  $F_h$ , la position générée respecte les limites virtuelles de la région de travail. La pénalisation de  $F_h$ , la génération de la vitesse désirée, et des tests de validité du générateur seront présentés en détail au cours de ce chapitre.

### 4.3.2 Signal mesuré par le capteur de force

Soit  $F_d$  le signal délivré par le capteur de force. Ce signal correspond à la force résultante de l'ensemble des forces appliquées sur le capteur. De plus, ce signal est généralement biaisé :

$$F_d = \sum F_{app} + b_r, \quad (4.15)$$

où :  $\sum F_{app}$  est la somme des forces appliquées sur le capteur, et  $b_r$  est le biais réel inconnu *a priori*. Il est donc nécessaire d'éliminer le biais afin de ne pas tromper les algorithmes du générateur de trajectoire. Pour cela, après avoir mis le robot en marche, nous le laissons dans une position statique, sans appliquer de force durant quelques secondes (10 *sec* par exemple). La seule force appliquée est bien la force de gravité  $P_t$  correspondante à la masse  $m_t$  de l'outil fixé sur le capteur. Pendant ces quelques secondes, nous récupérons un signal de force auquel nous avons soustrait la force de gravité due à l'outil qui s'applique au capteur de force. Ensuite, nous calculons la valeur moyenne de ce signal sur cette durée. Nous considérons cette valeur comme une estimation du biais, notée par  $b_e$  :

$$b_e \simeq b_r. \quad (4.16)$$

La figure 4.10 illustre la méthode d'estimation du biais pendant les premières secondes de la mise en marche du robot.

La valeur estimée du biais  $b_e$  sera soustraite au signal mesuré pendant toute la durée d'exécution. Le signal  $F_{sf}$  récupéré au niveau du repère attaché au capteur de force est :

$$F_{sf} = F_d - b_e - m_t {}^{sf}R_{ee} {}^{ee}R_0 V_g, \quad (4.17)$$

où  $V_g$  est le vecteur de gravité exprimé dans un repère absolu fixe,  ${}^{ee}R_0$  et  ${}^{sf}R_{ee}$  sont les matrices de rotation permettant le passage du repère absolu vers le repère attaché à l'outil, et du repère de l'outil vers le repère attaché au capteur de force, respectivement. Le terme  $(m_t {}^{sf}R_{ee} {}^{ee}R_0 V_g)$  présente le module de soustraction de la

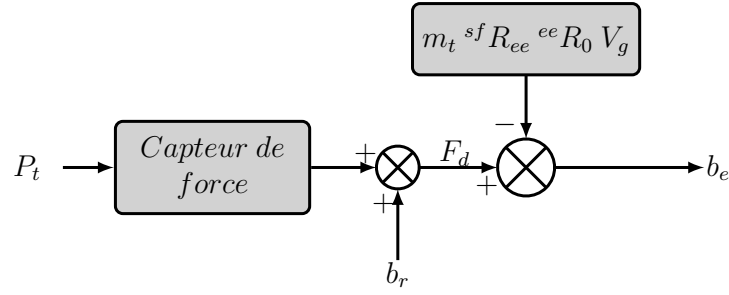


FIGURE 4.10 – Estimation du biais pendant les premières secondes de la mise en marche du robot.

force de gravité appliquée par l'outil, utilisé pour estimer le biais. Ce module reste installé pendant la durée de la manipulation, ce qui permet de récupérer un signal sans l'effet dû au poids de la masse de l'outil  $m_t$ . Ensuite, nous remplaçons  $F_d$  par son expression de (4.15) dans (4.17) :

$$F_{sf} = \sum F_{app} + b_r - b_e - m_t {}^{sf}R_{ee} {}^{ee}R_0 V_g. \quad (4.18)$$

Nous supposons que la valeur estimée du biais est proche de sa valeur réelle (équation (4.16)). D'où :

$$F_{sf} \simeq \sum F_{app} - m_t {}^{sf}R_{ee} {}^{ee}R_0 V_g. \quad (4.19)$$

La figure 4.11 illustre la méthode de récupération du signal de la force mesurée  $F_{sf}$  exprimé dans le repère attaché au capteur.

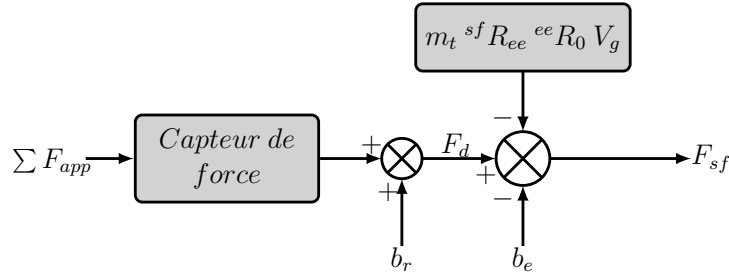


FIGURE 4.11 – Récupération du signal de la force mesurée exprimé dans le repère attaché au capteur.

Dans le cas où le capteur de force n'est pas soumis à des forces extérieures, donc  $\sum F_{app} = P_t$  :

$$\begin{aligned} F_{sf} &= P_t + b_r - b_e - m_t {}^{sf}R_{ee} {}^{ee}R_0 V_g \\ &= \simeq 0. \end{aligned} \quad (4.20)$$

Dans une étape suivante, le signal récupéré  $F_{sf}$  est filtré.

### 4.3.3 Estimation de la force de l'opérateur

Pendant un essai expérimental de co-manipulation homme-robot, l'outil est en mouvement dans l'espace. L'ensemble {outil , capteur de force} devient donc un

système dynamique soumis à la force appliquée par l'opérateur et à la force inertielle due de cet ensemble. Notons que cette dernière est nulle en statique. Dans la figure 4.12, nous proposons un schéma du capteur de force inséré entre le bras d'un robot et un outil représenté par une masse ponctuelle. Le schéma illustre le cas unidimensionnel. L'extension au cas tridimensionnel est obtenu aisément à partir du cas unidimensionnel.

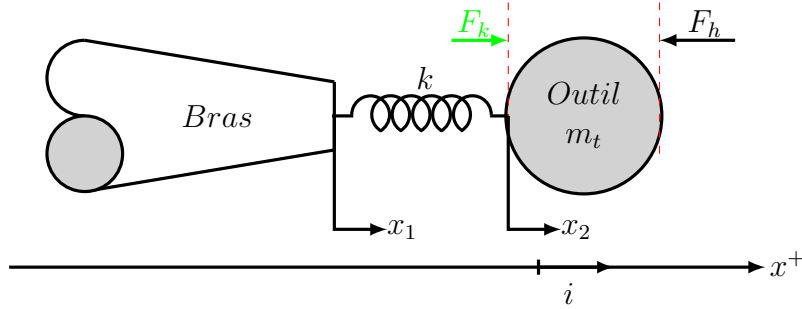


FIGURE 4.12 – Capteur de force installé entre un bras et un outil.

Le capteur de force est caractérisé par un ressort de raideur  $k$  et une masse supposée être incluse dans celle de l'outil dans le modèle que nous considérons. Soient  $x_1$  et  $x_2$  les déplacements des deux extrémités du ressort dans le repère absolu. La raideur  $k$  qui caractérise le capteur de force est très élevée. Dans le cas du capteur *Gamma* de la société *ATI Industrial Automation*, les valeurs qui sont données pour ces raideurs sont  $9.1 \times 10^6 \text{ N/m}$  selon les axes  $x$  et  $y$ , et  $1.8 \times 10^7 \text{ N/m}$  selon l'axe  $z$  [1]. Compte tenu de ces ordres de grandeurs, nous pouvons considérer que la différence entre les déplacements  $x_1$  et  $x_2$  est négligeable devant le déplacement de corps rigide de l'ensemble :

$$x_2 \simeq x_1. \quad (4.21)$$

On désigne par  $\ddot{x}_1$  l'accélération de la masse  $m_t$  calculée à la base des différentes variables articulaires  $q$ ,  $\dot{q}$  et  $\ddot{q}$  :

$$\ddot{x}_1 = \dot{J}_v(q, \dot{q}) \dot{q} + J_v(q) \ddot{q}, \quad (4.22)$$

où  $J_v$  est la matrice jacobienne de la vitesse linéaire donnée par l'équation (3.15). La méthode d'obtention de l'équation (4.22) sera présentée dans la section 5.2. L'équation du mouvement de la masse  $m_t$  est la suivante :

$$F_k + F_h = m_t \ddot{x}_1, \quad (4.23)$$

où  $F_h$  est la force extérieure appliquée par l'environnement sur le robot. Dans notre étude,  $F_h$  correspond à la force appliquée par un opérateur humain.  $F_k$  correspond au signal mesuré ( $F_{mes}$ ) par le capteur de force. Il vient :

$$F_k = F_{mes} = {}^0R_{ee} {}^{ee}R_{sf} F_{sf}. \quad (4.24)$$

A travers les équations (4.23) et (4.24) :

$$F_h = -{}^0R_{ee}{}^{ee}R_{sf} F_{sf} + m_t \ddot{x}_1. \quad (4.25)$$

Cette dernière équation représente l'expression de la force de l'opérateur estimée en fonction de l'accélération du manipulateur et de la force mesurée. Elle est illustrée aussi par le schéma suivant :

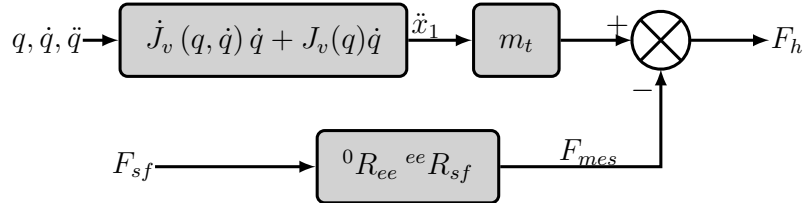


FIGURE 4.13 – Estimation de la force de l'opérateur.

L'équation (4.25) reste valable pour le cas tridimensionnel car la force de gravité due à la masse  $m_t$  est compensée avant l'acquisition du signal  $F_{sf}$ .

#### 4.3.4 Pénalisation de la force de l'opérateur

L'objectif de l'ajout du bloc *Pénalisation de Force* est de garantir que le robot ne dépasse pas les limites de la zone de travail. La première idée pour la conception de ce bloc s'est basée sur la théorie du champ potentiel [51], utilisée pour la génération de trajectoire. Cette théorie consiste à créer un champ de forces attractives autour de la position désirée (cible) du robot, et des champs de forces répulsives autour des obstacles, afin d'éviter la collision du robot avec ces derniers. Ainsi, le robot est considéré comme une particule qui fait un mouvement dans un champ de potentiel généré par le point désiré, et les obstacles qui se présentent dans son environnement. Le champ de potentiel peut être vu comme un champ d'énergie. D'où, son gradient en un point correspond à une force. Le robot dans ce champ est soumis aux actions des forces correspondant aux gradients des champs de potentiel attractifs et répulsifs. Ces forces permettent de le guider vers le point objectif, et d'éviter les obstacles au cours du mouvement.

Pour des tâches de co-manipulation dans un espace à limites virtuelles, la considération de ces limites comme des obstacles à champs répulsifs implique qu'ils vont pousser le robot à s'éloigner s'il est proche d'une limite virtuelle. Par conséquent, la propriété de transparence de la co-manipulation diminue à mesure que le point d'interaction s'approche des limites virtuelles.

Afin de satisfaire la condition de transparence sur tout l'espace de travail limité, nous proposons une pénalisation de la composante normale de la force de l'opérateur. Cette pénalisation permet une adaptation de la force à la contrainte des limites de l'espace de travail (figure 4.9).

Nous désignons par espace (ou bien zone ou région) de travail d'un robot, un volume limité où le mouvement du point d'interaction est permis. Il est à noter que la



notion d'espace de travail utilisée ici est différente de celle qui correspond à « l'Espace atteignable », qui correspondra, lui, à l'ensemble des points atteignables par l'organe terminal du robot. La restriction de mouvement du robot dans un espace inclus dans l'espace atteignable permet d'éviter les singularités aux limites dues à l'extension complète du bras.

Une co-manipulation homme-robot ne doit pas permettre au robot de sortir de sa zone de travail. Pour cela, le générateur de trajectoire doit générer des trajectoires de référence incluses dans cette région. Si la force  $F_h$  de l'opérateur guide le robot vers l'extérieur de son espace de travail, la composante normale de  $F_h$  est alors pénalisée jusqu'à ce qu'elle devienne nulle à la limite de l'espace. Dans ce cas, les composantes binormale et tangentielle de  $F_h$ , portées par le plan tangentiel, guident le robot. Dans le cas où la direction de la composante normale de la force est inversée, cette dernière devient effective à nouveau. Par conséquent, le robot s'éloignera des limites de l'espace de travail.

### Étude générale

Soit un espace de travail en tridimensionnel, noté  $(\Xi)$ , défini par un système de  $m$  équations :

$$g_i(x, y, z) \leq 0, \quad i = 1, \dots, m. \quad (4.26)$$

Le système (4.26) définit un espace de travail **fermé**. Les équations dans (4.26) définissent la surface limitante  $(\xi)$  de l'espace  $(\Xi)$ . A un instant  $t$ , la force de l'opérateur  $F_h$  a pour composantes  $F_h = [F_x \ F_y \ F_z]_{R_0}^T$  dans le repère absolu  $R_0$ , et l'organe terminal a pour position de référence  $X_d = [x_d \ y_d \ z_d]_{R_0}^T$ . La position initiale  $X_0 = [X_{x_0} \ X_{y_0} \ X_{z_0}]_{R_0}^T$  de l'organe terminal peut être à l'intérieur de l'espace de travail ( $g(X_{x_0}, X_{y_0}, X_{z_0}) \leq 0$ ), ou hors cet espace ( $g(X_{x_0}, X_{y_0}, X_{z_0}) > 0$ ). Soit la droite  $(\Delta)$  dans l'espace passant par le point  $X_d$  et dirigée par le vecteur  $F_h$ . Cette dernière est caractérisée par les équations suivantes :

$$(\Delta) : \begin{cases} x = F_x \sigma + x_d, \\ y = F_y \sigma + y_d, \\ z = F_z \sigma + z_d. \end{cases} \quad \sigma \in R, \quad (4.27)$$

Afin de trouver les points d'intersection entre la droite  $(\Delta)$  et la surface  $(\xi)$ , nous résolvons le système d'équations suivant :

$$\begin{cases} g_i(x, y, z) = 0 \\ x = F_x \sigma + x_d, \\ y = F_y \sigma + y_d, \\ z = F_z \sigma + z_d. \end{cases} \quad , \quad i = 1, \dots, m \quad (4.28)$$

Notons  $P$  l'ensemble des points  $P_i(x_i, y_i, z_i)$  solutions du système (4.28). Nous pouvons alors distinguer les deux cas suivants :

1. **Cas 1** :  $P = \emptyset$

Si l'ensemble  $P$  est vide, cela signifie qu'il n'y a pas de butée le long de la direction de la force  $F_h$  (figure 4.14). Le robot est donc à l'extérieur de la région de travail  $(\Xi)$ .

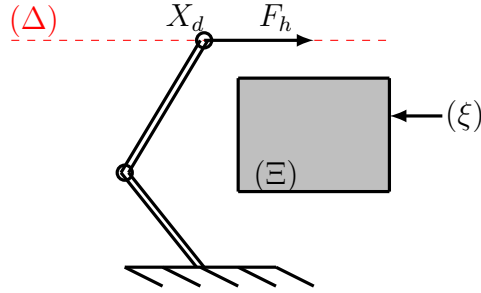


FIGURE 4.14 – Illustration du cas :  $P = \emptyset$ .

Ce cas peut se présenter à l'instant initial d'un essai de co-manipulation (robot initialement à l'extérieur de la zone). Notons par  $F_h'$  l'image de  $F_h$  qui est utilisée en entrée du générateur pour calculer la vitesse désirée. Dans le cas d'un ensemble  $P$  vide,  $F_h'$  est nulle :

$$\begin{cases} P = \emptyset \Rightarrow \exists i, g_i(X_d) > 0 \\ F_h' = 0. \end{cases} \quad (4.29)$$

Le cas d'un ensemble  $P$  vide signifie que le robot est à l'extérieur de l'espace de travail virtuel, et la force  $F_h$  est, de plus, ne tend pas à faire pénétrer le robot dans l'espace  $(\Xi)$  (figure 4.14).

## 2. Cas 2 : $P \neq \emptyset$

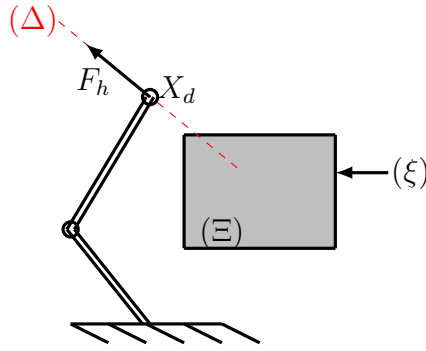
Si l'ensemble  $P$  n'est pas vide, nous calculons les rapports, notés  $k_i$ , entre le vecteur  $F_h$  et chaque vecteur  $X_d P_i$ , avec  $P_i(x_i, y_i, z_i) \in P$ . Pour cela, il suffit de calculer le rapport entre une composante  $j$  non nulle de  $F_h$ , et la même composante  $j$  de chaque vecteur  $X_d P_i$  :

$$k_i = \frac{P_i(j) - X_d(j)}{F_h(j)}, \quad F_h(j) \neq 0. \quad (4.30)$$

Nous regroupons alors les rapports  $k_i$  dans le vecteur  $K$ , et les rapports  $k_i$  positifs dans le vecteur  $K'$ . Nous distinguons alors les cas suivants :

### (a) Cas 2.1 : $K' = \emptyset$

Si tous les  $k_i$  sont négatifs ou nuls, l'organe terminal est donc à l'extérieur ou à la limite de l'espace  $(\Xi)$ , et la force  $F_h$  est dirigée dans un sens qui guide le robot à s'éloigner de plus en plus de l'espace  $(\Xi)$  (figure 4.15).

FIGURE 4.15 – Illustration du cas :  $P \neq \emptyset$  &  $K' = \emptyset$ .

Dans ce cas,  $F_h'$  est nulle :

$$\begin{cases} P \neq \emptyset, K' = \emptyset \Rightarrow \exists i, g_i(X_d) \geq 0 \\ F_h' = 0. \end{cases} \quad (4.31)$$

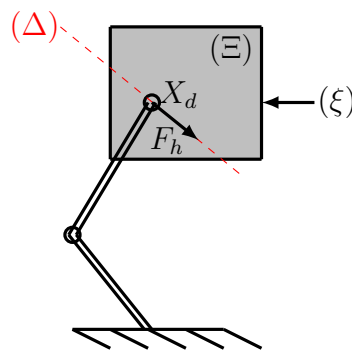
(b) **Cas 2.2** :  $K' \neq \emptyset$

Si  $\exists k_i \in K'$ , nous distinguons deux cas possibles :

- **Le robot est à l'intérieur de l'espace  $(\Xi)$**  :  $\forall i, g_i(X_d) \leq 0$  :

Dans ce cas (figure 4.16), la force  $F_h'$  soit la force  $F_h$  pénalisée par rapport au point  $P_i$  correspond au plus petit  $k_i$  positif. Elle est notée  $F_{P_1}$  :

$$\begin{cases} P \neq \emptyset, K \neq \emptyset, \forall i, g_i(X_d) \leq 0 \\ F_h' = F_{P_1}. \end{cases} \quad (4.32)$$

FIGURE 4.16 – Illustration du cas :  $P \neq \emptyset, K' \neq \emptyset$  &  $\forall i, g_i(X_d) \leq 0$ 

- **Le robot est à l'extérieur de l'espace  $(\Xi)$**  :  $\exists i, g_i(X_d) > 0$  :

Dans ce cas (figure 4.17), la force  $F_h'$ , soit la force  $F_h$  pénalisée par rapport au point  $P_i$ , correspond au deuxième plus petit  $k_i$  positif. Si le robot est en dehors de la région de travail, le plus petit  $k_i$  correspond au point de pénétration du robot vers la zone de travail. La force pénalisée

est notée  $F_{P_2}$  :

$$\begin{cases} P \neq \emptyset, K \neq \emptyset, \exists i, g_i(X_d) > 0 \\ F_h' = F_{P_2}. \end{cases} \quad (4.33)$$

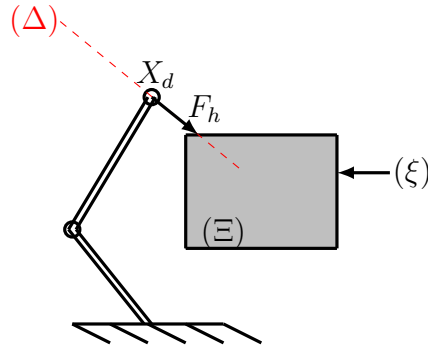


FIGURE 4.17 – Illustration du cas :  $P \neq \emptyset, K' \neq \emptyset$  &  $\exists i, g_i(X_d) > 0$

L'organigramme de la figure 4.18 résume les différents cas qui peuvent être rencontrés pour une co-manipulation dans un espace de travail fermé :

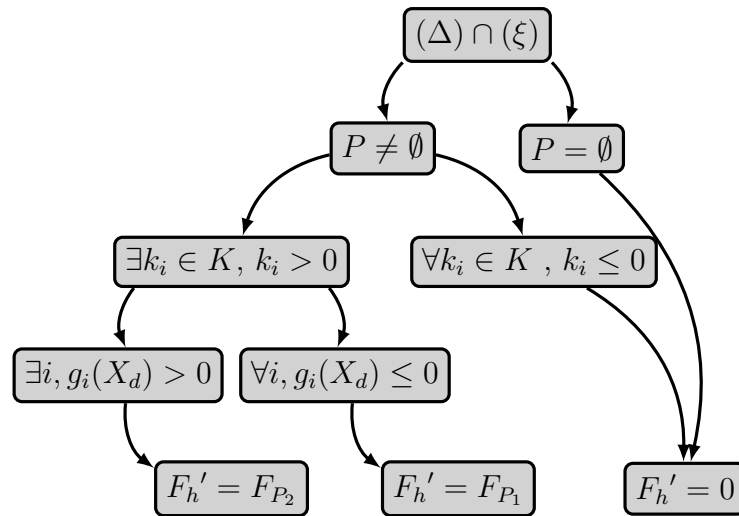
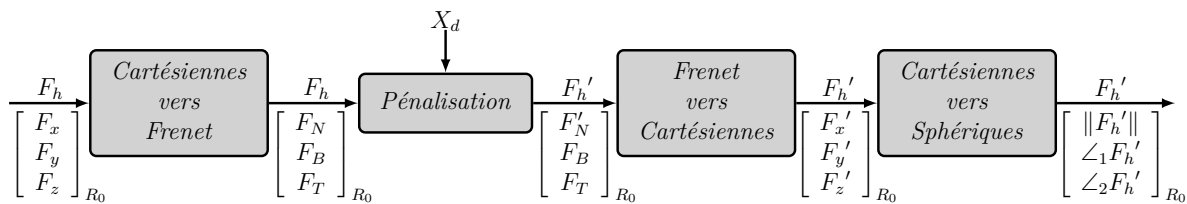


FIGURE 4.18 – Les différents cas possibles

### Calcul des composantes de la force pénalisée

La figure 4.19 présente le contenu du bloc *Pénalisation de Force* de la figure 4.9, qui résume la procédure de pénalisation de la force  $F_h$ , pour les cas où la force  $F_h'$  n'est pas nulle.

FIGURE 4.19 – Procédure de pénalisation de  $F_h$ .

Dans ce qui suit, nous présentons les étapes de calcul des composantes de  $F_h'$ , illustrées dans la figure 4.19.

### 1. Composantes du vecteur normal

Soit  $(\rho)$  le plan tangentiel à  $(\xi)$  au point  $P_s = [x_s \ y_s \ z_s]^T$  correspondant au  $k_i$  pris en compte pour la pénalisation de  $F_h$  (plus petit ou deuxième plus petit). Le plan  $(\rho)$  est défini par l'équation suivante :

$$(\rho) : (x - x_s) \frac{\partial g_i}{\partial x} \Big|_{P_s} + (y - y_s) \frac{\partial g_i}{\partial y} \Big|_{P_s} + (z - z_s) \frac{\partial g_i}{\partial z} \Big|_{P_s} = 0, \quad (4.34)$$

tel que :  $g_i(P_s) = 0$ . Le développement de l'équation (4.34) permet de la réécrire sous la forme :

$$(\rho) : ax + by + cz + d = 0. \quad (4.35)$$

Notons par  $N = [a \ b \ c]^T$  le vecteur normal au plan  $(\rho)$ .  $N$  est aussi normal à la surface  $(\xi)$  au point  $P_s$ .

### 2. Passage de la base cartésienne vers la base de Frenet

Le passage de la base cartésienne vers la base de Frenet est nécessaire afin de calculer la composante normale de  $F_h$ , notée  $F_N$  au point  $P_s$ . Pour cela, nous procédons aux transformations suivantes :

#### (a) Rotation autour $z_0$ de $R_0$

On fait une rotation du repère  $R_0$  autour de l'axe  $z_0$ , d'un angle  $\varphi$  défini comme suit :

$$\varphi = \arctan\left(\frac{b}{a}\right). \quad (4.36)$$

La base résultante de cette transformation est notée  $R_\varphi$ . La relation entre les deux bases  $R_0$  et  $R_\varphi$  est donnée par l'équation suivante :

$$\begin{bmatrix} x_\varphi \\ y_\varphi \\ z_\varphi \end{bmatrix}_{R_\varphi} = \underbrace{\begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\varphi R_0 = \text{Rot}(z_0, \varphi)} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_{R_0}. \quad (4.37)$$

La matrice  ${}^\varphi R_0$  est souvent appelée matrice de rotation de la base  $R_0$  vers la base  $R_\varphi$ .

(b) **Rotation autour  $y_\varphi$  de  $R_\varphi$** 

On fait une rotation du repère  $R_\varphi$  autour de l'axe  $y_\varphi$ , d'un angle  $\theta$  défini comme suit :

$$\theta = \arctan\left(\frac{c}{\sqrt{a^2 + b^2}}\right). \quad (4.38)$$

La base résultante de cette transformation est notée  $R_{Frm}$ . La relation entre les deux bases  $R_\varphi$  et  $R_{Frm}$  est donnée par l'équation suivante :

$$\begin{bmatrix} x_\theta \\ y_\theta \\ z_\theta \end{bmatrix}_{R_{Frm}} = \underbrace{\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}}_{{}^\theta R_\varphi = \text{Rot}(y_\varphi, \theta)} \begin{bmatrix} x_\varphi \\ y_\varphi \\ z_\varphi \end{bmatrix}_{R_\varphi}. \quad (4.39)$$

**3. Projection du vecteur force dans la base de Frenet**

Le vecteur  $F_h$  est projeté sur la base de Frenet à l'aide des matrices de rotation  ${}^\varphi R_0$  et  ${}^\theta R_\varphi$  :

$$\begin{bmatrix} F_N \\ F_B \\ F_T \end{bmatrix}_{R_{Frm}} = {}^\theta R_\varphi {}^\varphi R_0 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{R_0}. \quad (4.40)$$

$F_B$  et  $F_T$  désignent les composantes binormale et tangentielle du vecteur  $F_h$ , respectivement.

**4. Fonction de pénalisation**

La fonction de pénalisation permet de pénaliser la composante  $F_N$ , de telle sorte que  $F_N$  diminue si la position de l'organe terminal s'approche de la surface qui limite l'espace de travail, jusqu'à s'annuler lorsque l'organe terminal se trouve sur la surface. Pour cela, nous utilisons la fonction  $f$  suivante :

$$f(x) = H(H(x) - \gamma_1) \left[ 4 \frac{\arctan(x+1)^\eta}{\pi} - 1 \right], \quad (4.41)$$

$$\gamma_1 \in \left] \frac{1}{2}, 1 \right[ ,$$

où  $H(\cdot)$  est la fonction de Heaviside, et  $\eta$  correspond au degré de courbure. La figure 4.20 illustre l'allure de cette fonction en fonction de la distance  $x$ .

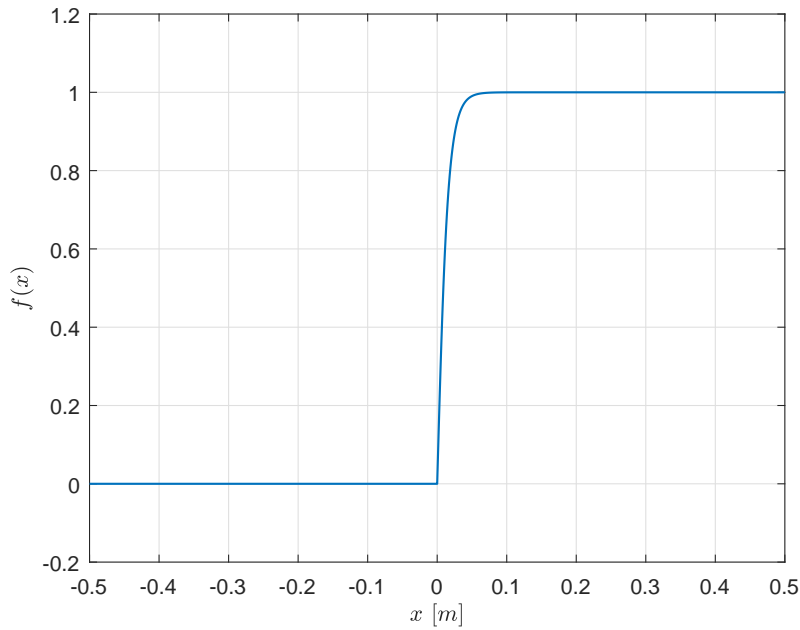
Nous pouvons remarquer que  $f(x)$  s'annule pour  $x \leq 0$ .

**5. Expression de la nouvelle composante normale de la force**

Notons  $R$  la distance entre la position de l'organe terminal  $X_d$  et le point  $P_s$  de la surface  $\xi$  :

$$R = \|X_d P_s\| = \sqrt{(x_s - x_d)^2 + (y_s - y_d)^2 + (z_s - z_d)^2}. \quad (4.42)$$

La nouvelle composante normale  $F_N'$ , après pénalisation, est alors une fonction

FIGURE 4.20 – Profil de  $f(x)$  pour  $\eta = 100$ , et  $\gamma_1 = 0.75$ .

de la distance  $R$  :

$$F_N' = f(R) F_N. \quad (4.43)$$

### 6. Composantes cartésiennes de $F_h'$

Le vecteur de la force pénalisée est exprimé dans la base cartésienne à l'aide des matrices de rotation  ${}^\varphi R_0$  et  ${}^\theta R_\varphi$  :

$$F_h' = ({}^\theta R_\varphi {}^\varphi R_0)^{-1} \begin{bmatrix} F_N' \\ F_B \\ F_T \end{bmatrix}_{R_{F_{rn}}}. \quad (4.44)$$

### Projection du nouveau vecteur force dans la base sphérique

Le vecteur de la force  $F_h'$ , exprimé initialement dans la base cartésienne, est exprimé dans la base sphérique afin d'avoir accès au module et aux angles d'orientation de  $F_h'$ . Cette projection est faite à l'aide des équations suivantes :

$$\begin{cases} \|F_h'\| = \sqrt{F_x'^2 + F_y'^2 + F_z'^2}, \\ \angle_1 F_h' = \arctan\left(\frac{F_z'}{\sqrt{F_x'^2 + F_y'^2}}\right), \\ \angle_2 F_h' = \arctan\left(\frac{F_y'}{F_x'}\right). \end{cases} \quad (4.45)$$

### 4.3.5 Générateur de la vitesse désirée

Le bloc *Générateur de Vitesse Désirée* de la figure 4.9 permet de générer la vitesse désirée  $\dot{X}_d$  de l'organe terminal. La figure 4.21 montre la structure de ce bloc :

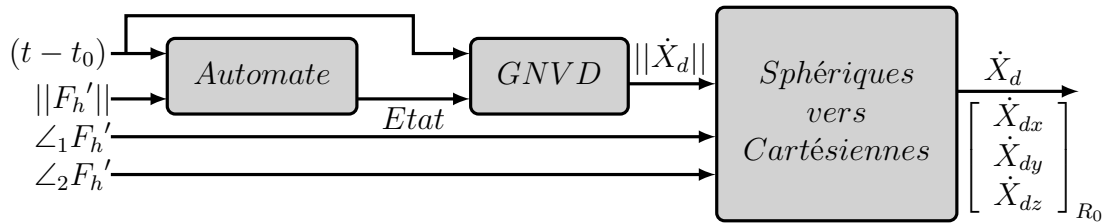


FIGURE 4.21 – Générateur de la vitesse désirée.

Le générateur de la vitesse est constitué des blocs suivants :

#### L'automate

L'automate possède quatre états. Son fonctionnement est illustré sur la figure 4.22. La vitesse générée dépend de l'état en cours de l'automate.

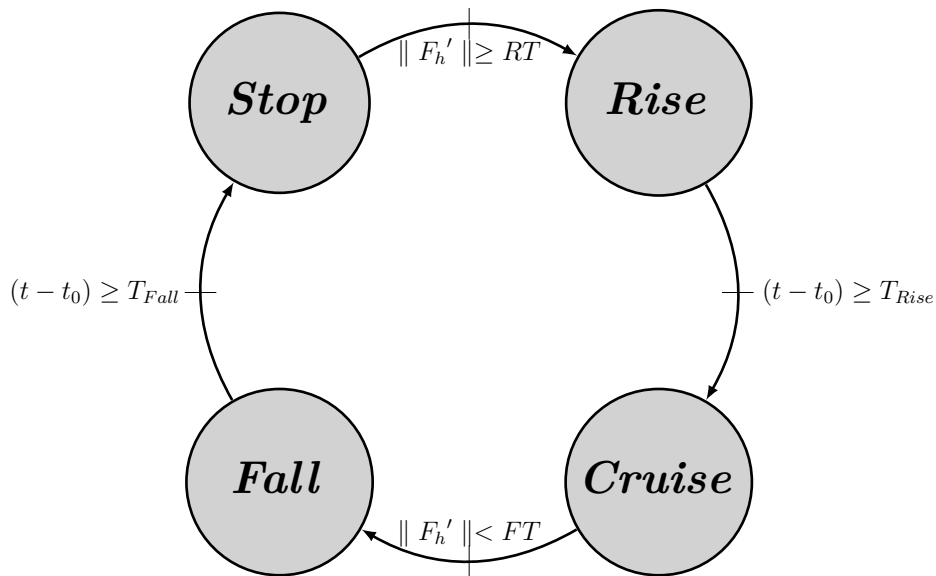


FIGURE 4.22 – L'automate gérant les états désirés du robot.

Pour le mouvement et l'arrêt du robot, deux seuils de la norme  $\|F_h'\|$  sont définis :

- $RT$  : signifie *Rise Threshold*. Il correspond à la valeur minimale de  $\|F_h'\|$  pour laquelle le robot bascule de l'état *Stop* ( $\|\dot{X}_d\| = 0$ ), vers l'état *Rise* ( $\|\dot{X}_d\| > 0$ );
- $FT$  : signifie *Fall Threshold*. Il correspond à la valeur maximale de  $\|F_h'\|$  pour laquelle le robot bascule de l'état *Cruise* ( $\|\dot{X}_d\| = V_C$ ), vers l'état *Fall* ( $\|\dot{X}_d\| < V_C$ ), où  $V_C$  est la vitesse de croisière.

A partir de la figure 4.22, nous pouvons distinguer les états suivants :

- L'état *Rise* : lorsque  $\|F_h'\|$  dépasse le seuil  $RT$ , l'état *Rise* est activé. Cela signifie que  $\|\dot{X}_d\|$  est en train d'augmenter. L'automate maintient cet état actif pendant



une durée égale à  $T_{Rise}$ . Même si  $\|F_h'\|$  devient inférieure à  $FT$ , l'automate ne prend pas en compte cette information tant que la durée d'activation de l'état *Rise* est inférieure à  $T_{Rise}$ . Cette durée est nécessaire pour que la norme de la vitesse désirée atteigne une vitesse de croisière prédéfinie ( $\|\dot{X}_d\| = V_C$ ) à partir de l'arrêt ;

- L'état *Cruise* : une fois la durée  $T_{Rise}$  est écoulée, cela signifie que la norme de la vitesse a atteint son maximum ( $\|\dot{X}_d\| = V_C$ ). L'état *Cruise* est maintenant activé.  $\|\dot{X}_d\|$  reste constante à la vitesse  $V_C$ , tant que  $\|F_h'\|$  est toujours supérieure à  $FT$ . Sinon, l'automate bascule vers l'état *Fall* ;
- L'état *Fall* : si  $\|F_h'\|$  descend moins que  $FT$ , l'automate active l'état *Fall*, cela signifie que  $\|\dot{X}_d\|$  commence à diminuer. Si  $\|F_h'\|$  dépasse  $RT$  avant la fin de la durée de cet état, égal à  $T_{Fall}$ , l'automate ne prend pas en compte cette information, et il maintient l'état *Fall* actif.
- L'état *Stop* : à la fin de la durée  $T_{Fall}$ ,  $\|\dot{X}_d\|$  devient nulle, c'est l'état *Stop* qui est maintenant actif. L'automate reste dans l'état *Stop* tant que  $\|F_h'\|$  est toujours inférieure à  $RT$ . Sinon, il bascule vers l'état *Rise*.

La non prise en compte des éventuels changements (augmentations/diminutions) de la norme de la force par rapport à  $RT$  et  $FT$ , durant les états *Rise* et *Fall*, permet d'avoir un profil de vitesse continu et lisse. Cela permet de protéger les actionneurs du robot des éventuels dysfonctionnements. Les changements brusques des vitesses comme par exemple, un passage de  $+100^\circ/s$  à  $-100^\circ/s$  instantanément, peut affecter le réducteur de vitesse, et même l'arbre. Aussi, pour la sécurité de l'opérateur, la norme de la vitesse de l'état *Cruise* (mouvement établi) est choisie constante, et respecte la norme sur la vitesse d'un organe terminal d'un robot, en interaction avec un être humain qui est de  $0.25 m/s$ .

### Générateur de la Norme de la Vitesse Désirée (GNVD)

A chaque instant  $t$ , le bloc GNVD génère la norme de la vitesse désirée pour le robot  $\|\dot{X}_d(t)\|$  en fonction de la durée  $(t - t_0)$ , et de l'état du robot :

$$\|\dot{X}_d(t)\| = fct((t - t_0), etat). \quad (4.46)$$

Le profil de  $\|\dot{X}_d\|$  est choisi afin d'avoir une continuité et une dérivabilité des profils de vitesse et d'accélération, lors d'un basculement d'un état à un autre. Comme l'accélération des états *Cruise* et *Stop* est égal à zéro (vitesses constante et nulle, respectivement), il est nécessaire que la dérivée de l'accélération au début et à la fin des états *Rise* et *Fall* soit nulle. L'expression de  $\|\dot{X}_d(t)\|$  est alors écrite comme un polynôme de la spline quintique donnée par (4.8) et (4.12). Le terme  $\|\dot{X}_d(t)\|$  joue alors le rôle de  $x(t)$  dans (4.8). Le polynôme est donné par l'expression suivante :

$$\|\dot{X}_d(t)\| = \|\dot{X}_{d0}\| + (\|\dot{X}_{df}\| - \|\dot{X}_{d0}\|) \cdot \left( 6 \left( \frac{t - t_0}{t_f - t_0} \right)^5 - 15 \left( \frac{t - t_0}{t_f - t_0} \right)^4 + 10 \left( \frac{t - t_0}{t_f - t_0} \right)^3 \right), \quad (4.47)$$

où  $t_0$  et  $t_f$  sont respectivement, les instants initial et final de l'état considéré,  $t_f - t_0$  est la durée de l'état (*Rise* ou *Fall*).  $\|\dot{X}_{d0}\|$  et  $\|\dot{X}_{df}\|$  sont les normes de la vitesse désirée aux instants  $t_0$  et  $t_f$ , respectivement. Le tableau suivant montre les valeurs des paramètres de l'équation (4.47) pour chaque état :

TABLE 4.1 – Valeurs des paramètres du profil de la vitesse pour chaque état [45].

État	Réglage des paramètres		
<i>Rise</i>	$t_f - t_0 = T_{Rise}$	$\ \dot{X}_{d0}\  = 0$	$\ \dot{X}_{df}\  = V_C$
<i>Cruise</i>	--	$\ \dot{X}_{d0}\  = V_C$	$\ \dot{X}_{df}\  = V_C$
<i>Fall</i>	$t_f - t_0 = T_{Fall}$	$\ \dot{X}_{d0}\  = V_C$	$\ \dot{X}_{df}\  = 0$
<i>Stop</i>	--	$\ \dot{X}_{d0}\  = 0$	$\ \dot{X}_{df}\  = 0$

La figure 4.23 montre la génération de la norme de  $\|\dot{X}_d\|$  en fonction d'un profil de  $\|F_h(t)\|$  donné.

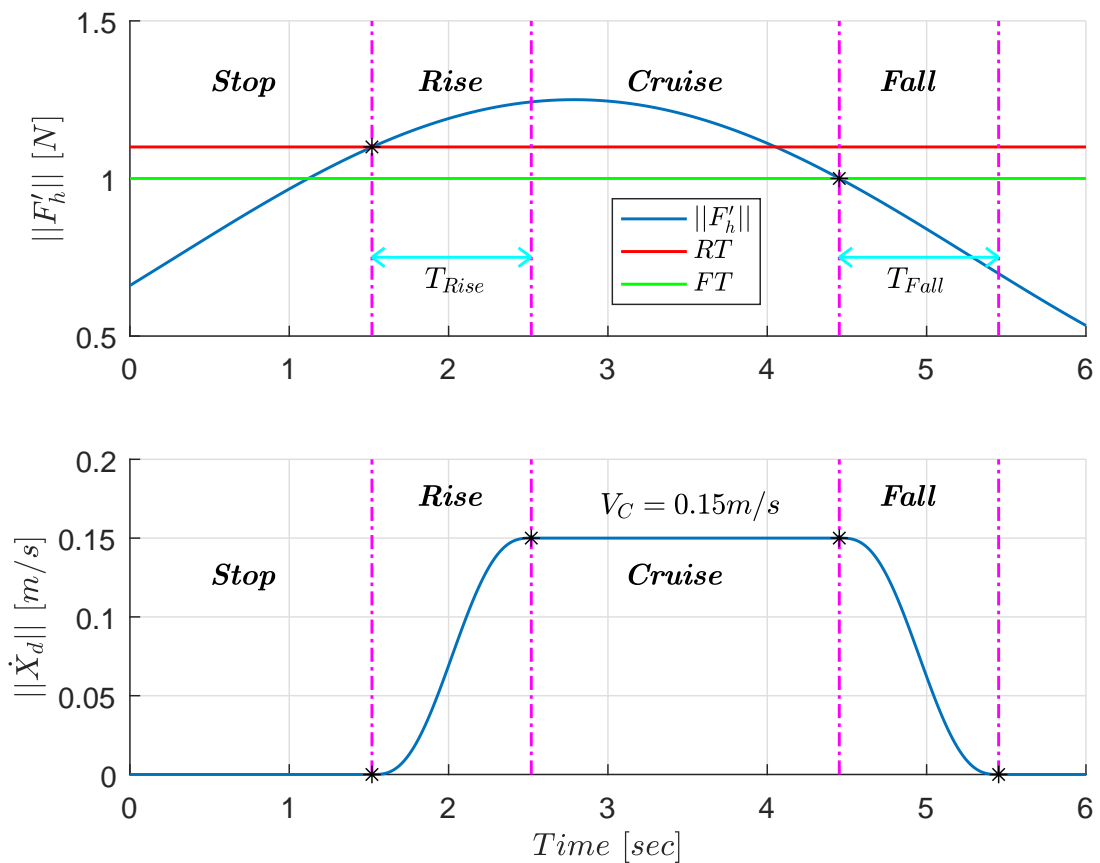


FIGURE 4.23 – Génération de  $\|\dot{X}_d\|$  en fonction de  $\|F_h'\|$ .

Le profil de  $\|\dot{X}_d(t)\|$  est généré par l'automate et le générateur de la norme de la vitesse présentés ci-dessus. Nous pouvons remarquer, sur la figure 4.23, une continuité de la norme de la vitesse, lors du basculement d'un état à un autre.

### Arguments de la vitesse désirée

Afin d'assurer la condition de transparence définie par (4.13a) et (4.13b), la vitesse désirée  $\dot{X}_d(t)$  doit être dirigée selon la direction de la force  $F_h'$  :

$$\begin{cases} \angle_1 \dot{X}_d = \angle_1 F_h', \\ \angle_2 \dot{X}_d = \angle_2 F_h'. \end{cases} \quad (4.48)$$

Si la force  $F_h'$  s'annule durant une phase de mouvement, la vitesse  $\dot{X}_d$  garde comme arguments les derniers arguments de  $F_h'$  avant qu'elle s'annule.

### Vitesse désirée dans la base cartésienne

La vitesse désirée déjà générée dans la base sphérique est convertie vers la base cartésienne absolue à travers les équations suivantes :

$$\begin{cases} \dot{x}_d = \|\dot{X}_d\| \cos(\angle_1 \dot{X}_d) \cos(\angle_2 \dot{X}_d), \\ \dot{y}_d = \|\dot{X}_d\| \cos(\angle_1 \dot{X}_d) \sin(\angle_2 \dot{X}_d), \\ \dot{z}_d = \|\dot{X}_d\| \sin(\angle_1 \dot{X}_d). \end{cases} \quad (4.49)$$

En utilisant l'équation (4.49), la position et l'accélération désirées de l'organe terminal du robot peuvent alors être calculées.

### 4.3.6 Pénalisation de la force de l'opérateur (suite)

A partir de la définition des états de l'automate décrits précédemment, quelques considérations sont à prendre en compte pour la pénalisation de la force.

#### Expression de la nouvelle composante normale de la force

Notons  $D_R$  et  $D_F$  les distances parcourues durant les phases *Rise* et *Fall*, respectivement. Ces distances peuvent être calculées par intégration numérique de la norme de la vitesse désirée. La zone de sécurité est caractérisée par  $D_F$ . Par conséquent, l'organe terminal doit évoluer dans un espace caractérisé par :

$$d = R - D_F \geq 0 \quad (4.50)$$

La nouvelle composante normale  $F_N'$  est donc la suivante :

$$F_N' = f(d) F_N \quad (4.51)$$

La distance de sécurité est nécessaire pour garantir que le robot ne sort pas de son espace de travail. Cela peut se produire, par exemple, dans le cas où l'état *Fall* est actif, lorsque  $X_d$  est presque à une distance  $D_F$  de la surface limite ( $\xi$ ) de l'espace de travail.

### Cas particulier

Afin d'améliorer l'algorithme de pénalisation de la force, nous traitons le cas particulier suivant :

- la force appliquée  $F_h$  est purement normale à la surface ( $\xi$ ) au point  $P_s$  ;
- $X_d$  est à une distance inférieure à  $(D_R + D_F)$  de ( $\xi$ ) ;
- l'état *Stop* est actif ;
- la norme de la force appliquée est supérieure  $RT$ .

Dans ce cas, si l'automate bascule vers l'état *Rise*, le point repéré par  $X_d(t)$  va parcourir une distance  $D_R$  et va se trouver à une distance inférieure à  $D_F$  de la surface ( $\xi$ ). En parallèle,  $F_N$  est en train d'être pénalisée et elle s'annule car  $d < 0 \Rightarrow f(d) = 0$ . Donc, juste à la fin de l'état *Rise*, l'automate va basculer vers l'état *Cruise*. Puisque la condition  $\|F_h\| < FT$  sera vérifiée, il basculera alors vers l'état *Fall*. Le point repéré par  $X_d$  va parcourir une distance  $D_F$ . Donc, le point repéré par  $X_d$  aura dépassé la surface ( $\xi$ ), ce qui ne convient pas à la logique de butée virtuelle que nous souhaitons instaurer. Pour remédier à ce cas, nous commençons d'abord par définir la distance  $s$  suivante :

$$s = R - (D_R + D_F). \quad (4.52)$$

L'état *Stop* est caractérisé par une norme de vitesse nulle  $\|\dot{X}_d\| = 0$ . Si la force  $F_h$  est purement normale, alors ses composantes tangentielle et binormale sont nulles :

$$\|F_h\| = F_N \Rightarrow F_T = 0 \ \& \ F_B = 0. \quad (4.53)$$

Soit les fonctions  $h_1$ ,  $h_2$  et  $h_3$  définies comme suit :

$$\begin{cases} h_1(s) = H(\gamma_2 - H(s)) = \begin{cases} 0 & \text{si } s \geq 0, \ \gamma_2 \in ]0, \frac{1}{2}[ \\ 1 & \text{si } s < 0 \end{cases} \\ h_2(\|\dot{X}_d\|) = 2H(-\|\dot{X}_d\|) = \begin{cases} 1 & \text{si } \|\dot{X}_d\| = 0 \\ 0 & \text{si } \|\dot{X}_d\| > 0 \end{cases} \\ h_3(F_T, F_B) = 4H(-|F_T|)H(-|F_B|) = \begin{cases} 1 & \text{si } F_T = 0 \ \& \ F_B = 0 \\ 0 & \text{sinon} \end{cases} \end{cases} \quad (4.54)$$

Le cas particulier décrit précédemment peut être exprimé à l'aide des fonctions  $h_1$ ,  $h_2$  et  $h_3$ . On considère la fonction  $h_4$  définie de la manière suivante :

$$\begin{aligned} h_4(s, \|\dot{X}_d\|, F_T, F_B) &= 1 - h_1(s) h_2(\|\dot{X}_d\|) h_3(F_T, F_B) \\ &= \begin{cases} 0 & \text{si } \|\dot{X}_d\| = 0 \ \& \ s < 0 \ \& \ F_T = 0 \ \& \ F_B = 0 \\ 1 & \text{sinon} \end{cases} \end{aligned} \quad (4.55)$$

La composante normale de  $F_h'$  dans la base cartésienne est donnée par la relation suivante :

$$F_N' = h_4(s, \|\dot{X}_d\|, F_T, F_B) f(d) F_N. \quad (4.56)$$

L'utilisation de la fonction  $h_4$  permet d'assurer que, dans le cas particuliers décrit précédemment, l'automate reste à l'état *Stop*, jusqu'à ce que le vecteur  $F_h$  change de

direction afin de garantir que  $X_d$  ne dépasse pas la surface limite ( $\xi$ ).

### Point d'intersection multiple

Le point d'intersection multiple résulte de l'intersection des surfaces limites. Dans ce cas, plusieurs composantes normales peuvent se présenter simultanément, chacune correspondant à une surface. Pour simplifier l'étude, nous pénalisons toutes les composantes de la force  $F_h$  dans ce cas :

$$F_h' = f(d)F_h. \quad (4.57)$$

### 4.3.7 Position et accélération désirées

La position et l'accélération désirées,  $X_d$  et  $\ddot{X}_d$  respectivement, de l'organe terminal du robot à l'instant  $t_k$  sont calculées par intégration et par dérivation numériques de  $\dot{X}_d(t_k)$ , de la manière suivante :

$$\begin{cases} X_d(t_k) = \Delta t \dot{X}_d(t_k) + X_d(t_{k-1}), \\ \ddot{X}_d(t_k) = \frac{\dot{X}_d(t_k) - \dot{X}_d(t_{k-1})}{\Delta t}, \end{cases} \quad (4.58)$$

où  $\Delta t$  est le pas d'échantillonnage,  $X_d(t_{k-1})$  et  $\dot{X}_d(t_{k-1})$  sont la position et la vitesse désirées à l'instant  $t_{k-1}$ .

## 4.4 Tests de validation

Dans cette section, nous nous intéressons à la validation du générateur de trajectoire développée. Pour cela, nous allons simuler, pour quelques configurations possibles, le fonctionnement du générateur pour générer la trajectoire d'un point  $p$ , initialement à une position  $X_0$ . Nous avons choisi un espace de travail ( $\Xi$ ) délimité par une surface ( $\xi$ ) composée de deux surfaces sphériques ( $\xi_{R_M}$ ) et ( $\xi_{R_m}$ ), de rayons  $R_M = 1 \text{ m}$  et  $R_m = 0.2 \text{ m}$  respectivement, et d'une surface plane ( $\xi_z$ ) appartient au plan  $z_0 = 0$  dans le repère fixe  $R_0$ . Il en résulte un espace ( $\Xi$ ) sous forme d'une demi-sphère creusée positive ( $z_0 > 0$ ). L'espace ( $\Xi$ ) est décrit par le système d'inégalités suivant :

$$\begin{cases} x^2 + y^2 + z^2 \leq R_M^2 \\ x^2 + y^2 + z^2 \geq R_m^2 \\ z \geq 0 \end{cases} . \quad (4.59)$$

$x$ ,  $y$  et  $z$  sont exprimés dans le repère  $R_0$ . Le système (4.59) peut être ré-écrit sous la forme des inégalités (4.26), comme suit :

$$\begin{cases} g_1 = x^2 + y^2 + z^2 - R_M^2 \leq 0 \\ g_2 = -(x^2 + y^2 + z^2) + R_m^2 \leq 0 \\ g_3 = -z \leq 0 \end{cases} . \quad (4.60)$$

En utilisant les équations (4.60), le système (4.28) s'écrit comme suit :

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} x^2 + y^2 + z^2 - R_M^2 = 0 \\ x = F_x\sigma + x_d \\ y = F_y\sigma + y_d \\ z = F_z\sigma + z_d \end{array} \right. \\ \left\{ \begin{array}{l} - (x^2 + y^2 + z^2) + R_m^2 = 0 \\ x = F_x\sigma + x_d \\ y = F_y\sigma + y_d \\ z = F_z\sigma + z_d \end{array} \right. \end{array} \right. \left\{ \begin{array}{l} -z = 0 \\ x = F_x\sigma + x_d \\ y = F_y\sigma + y_d \\ z = F_z\sigma + z_d \end{array} \right. . \quad (4.61)$$

Le système (4.61) admet au plus 5 solutions. Il y a donc au maximum 5 points d'intersection (dont les points rejetés) entre la droite ( $\Delta$ ) qui porte le vecteur  $F_h$ , et les surfaces limites ( $\xi$ ) ( $(\xi_{R_M})$ ,  $(\xi_{R_m})$ ,  $(\xi_z)$ ). Selon le point  $P_s$  pris en compte (s'il existe) pour la pénalisation de la composante  $F_N$ , l'équation du plan tangentiel ( $\rho$ ) à ( $\xi$ ) au point  $P_s$  peut être :

$$(\rho) : x_s x + y_s y + z_s z - (x_s^2 + y_s^2 + z_s^2) = 0, \quad (4.62)$$

si  $P_s$  résulte de l'interaction de ( $\Delta$ ) avec l'une des deux surfaces ( $\xi_{R_M}$ ) ou ( $\xi_{R_m}$ ). Le vecteur normal  $N$  au plan ( $\rho$ ), et au même temps à ( $\xi_{R_M}$ ) (ou ( $\xi_{R_m}$ )) au point  $P_s$  est :

$$N = [x_s \quad y_s \quad z_s]^T. \quad (4.63)$$

Si  $P_s$  résulte de l'intersection de ( $\Delta$ ) avec ( $\xi_z$ ), l'équation de ( $\rho$ ) soit :

$$(\rho) : z - z_s = 0, \quad (4.64)$$

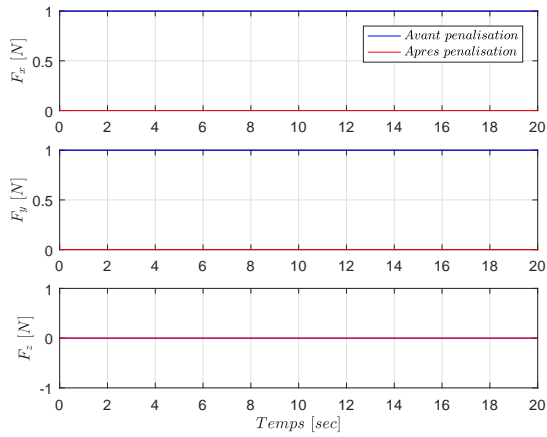
et le vecteur  $N$  est donc :

$$N = [0 \quad 0 \quad 1]^T. \quad (4.65)$$

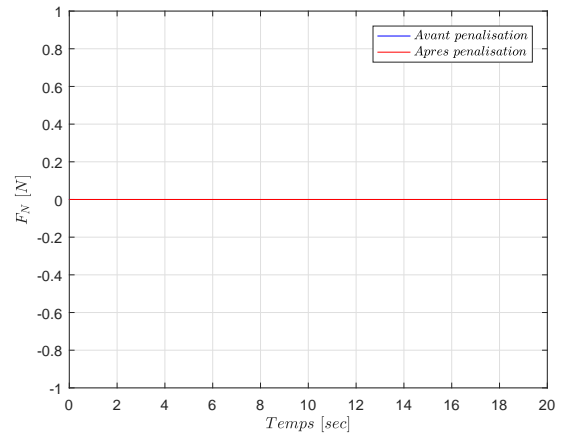
Pour les simulations faites pour tester le générateur, nous avons pris les seuils de mise en mouvement (*Rise Threshold*)  $RT$ , et celui d'arrêt du mouvement (*Fall Threshold*)  $FT$  égal à 1.1 [ $N$ ] et 1 [ $N$ ], respectivement. Les durées des états *Rise* et *Fall* sont  $T_{Rise} = 1$  [ $s$ ],  $T_{Fall} = 1$  [ $s$ ]. La vitesse de croisière est  $V_C = 0.1$  [ $m/s$ ]. La simulation démarre dans le cas où l'état initial du robot est à *Stop*. Les distances  $D_R$  et  $D_F$  sont calculées par intégration numérique du profil de la norme de la vitesse (équation (4.47) et tableau 4.1) sur les durées  $T_{Rise}$  et  $T_{Fall}$ , respectivement :  $D_R = D_F = 0.05$  [ $m$ ]. Dans ce qui suit, nous allons présenter les trajectoires produites par le générateur à la base des paramètres mentionnés ci-dessus, pour quelques positions initiales de l'organe terminal  $X_0$ , associées aux quelques directions de la force  $F_h$ .

Il est à noter que dans cette section, nous nous intéressons à la validation du générateur au niveau de l'exécution correct des états de l'automate, et le respect des surfaces limites, selon la direction de  $F_h$ . La validation de la relation entre la vitesse désirée générée  $\dot{X}_d$  et la force  $F_h'$  sera constatée dans le chapitre suivant.

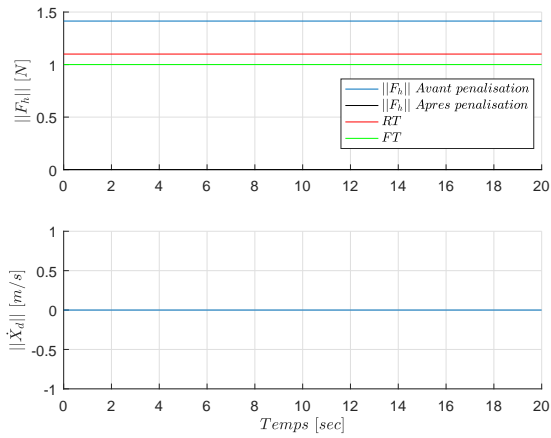




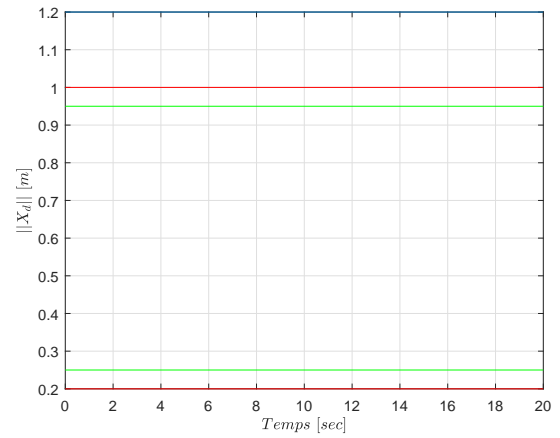
(a)  $F_h$ .



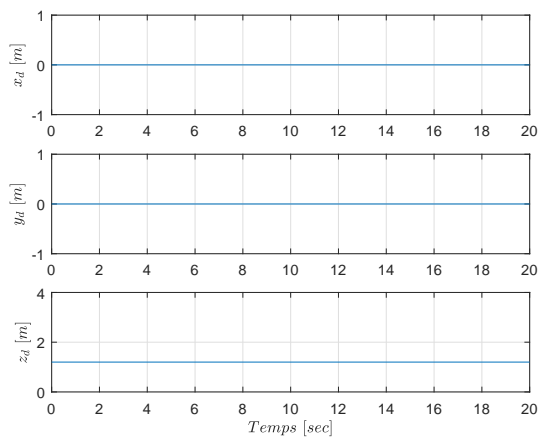
(b)  $F_N$ .



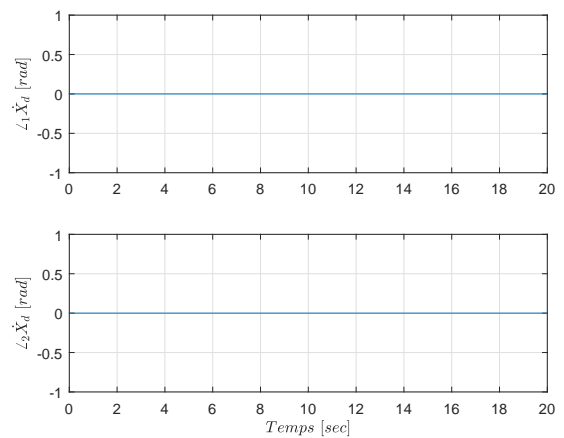
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d$ ,  $y_d$  &  $z_d$ .



(f) Angles de  $\dot{X}_d$ .

FIGURE 4.25 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 1.2]^T$  et  $F_h = [1 \ 1 \ 0]^T$ .



**Cas 2 :**  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ 2]^T$

Pour ce cas, la position initiale du point  $p$  est aussi hors l'espace de travail, et la force  $F_h$  est dirigée selon la direction positive de l'axe  $z_0$  (figure 4.26).

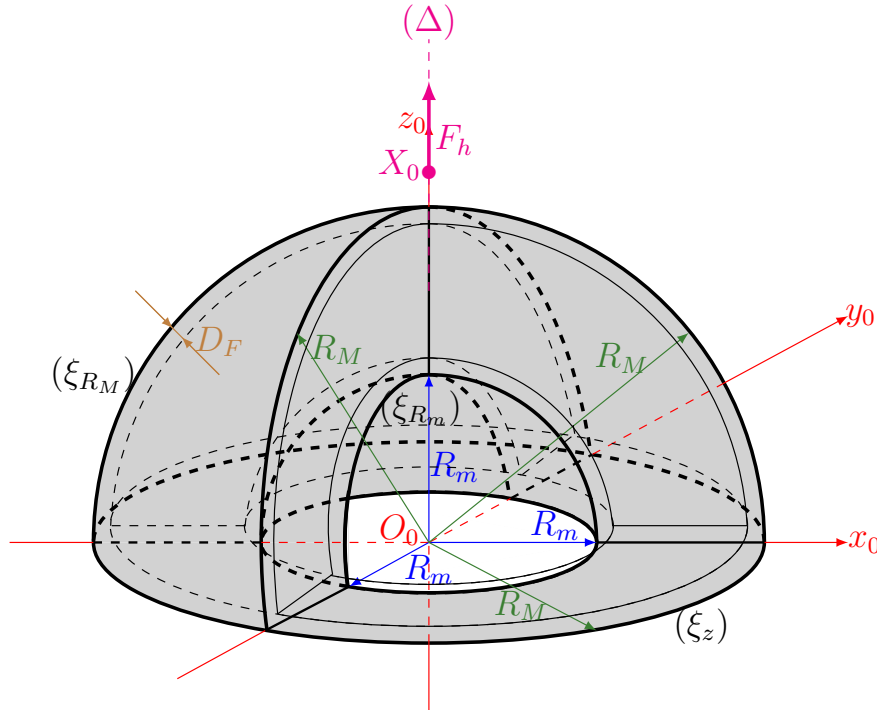
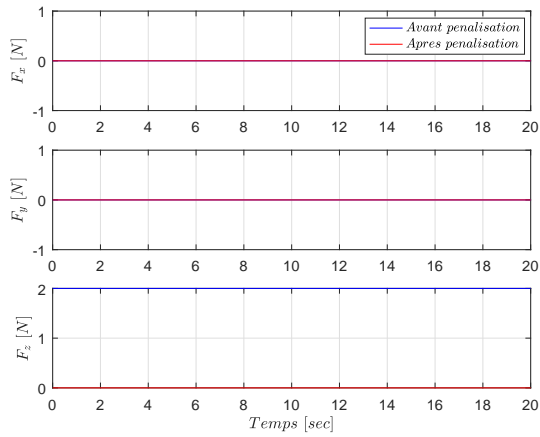
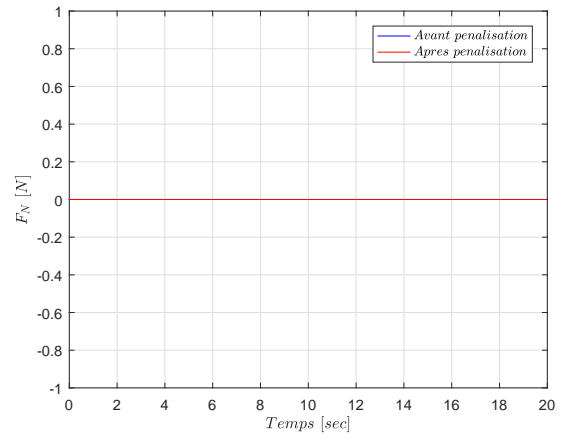


FIGURE 4.26 – Illustration du cas :  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

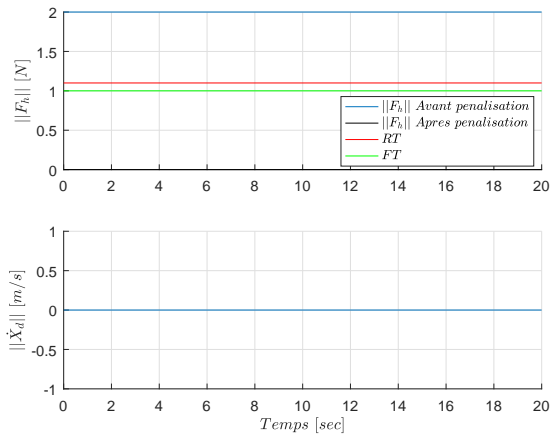
D'après la figure 4.26, la droite  $(\Delta)$  intersecte avec les surfaces  $(\xi_{R_M})$  et  $(\xi_{R_m})$ , mais les rapports  $k_i$  (équation (4.30)) sont tous négatifs. La force  $F_h$  ne guide pas le point  $p$  vers la zone de travail. La pénalisation de la force dans ce cas donne un vecteur  $F_h'$  nul (figures 4.27a et 4.27b). L'automate reste à l'état *Stop*, et le générateur produit une vitesse désirée nulle (figures 4.27c et 4.27f). La position désirée de  $p$  reste immobile à  $X_0$  (figures 4.27e et 4.27d).



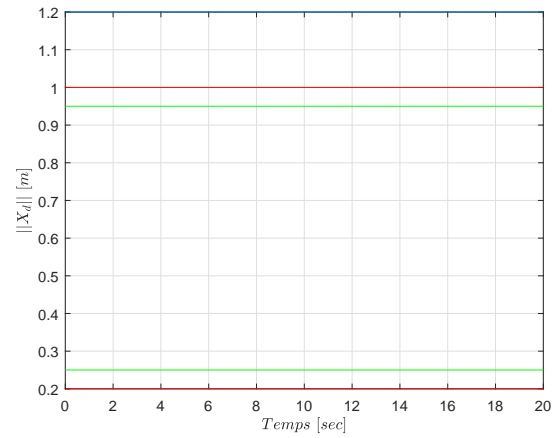
(a)  $F_h$ .



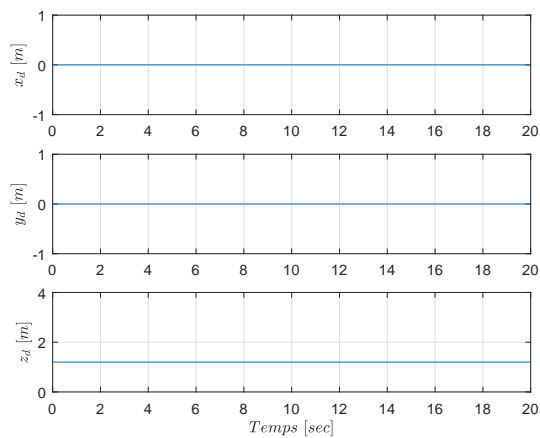
(b)  $F_N$ .



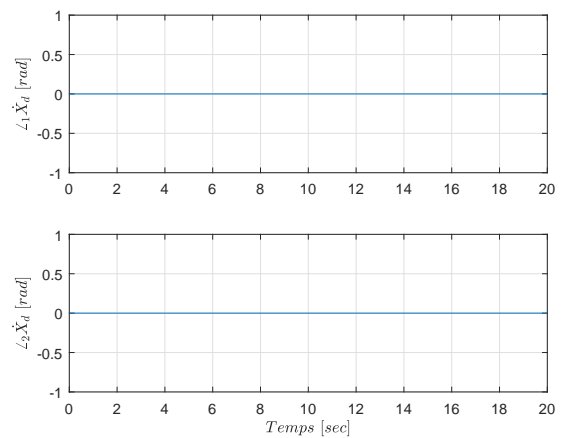
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d, y_d$  &  $z_d$ .



(f) Angles de  $\dot{X}_d$ .

FIGURE 4.27 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

**Cas 3 :**  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ -2]^T$

Pour ce troisième cas, la position initiale du point  $p$  est toujours en dehors la région de travail ( $\Xi$ ), mais la force  $F_h$  cette fois ci, est dirigée vers ce dernier (figure 4.28).

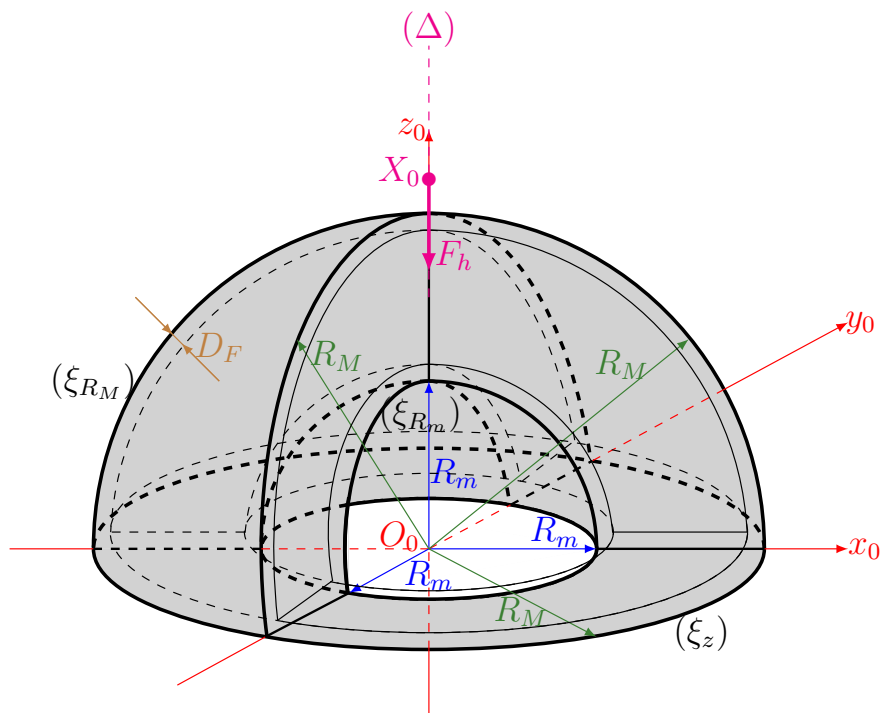
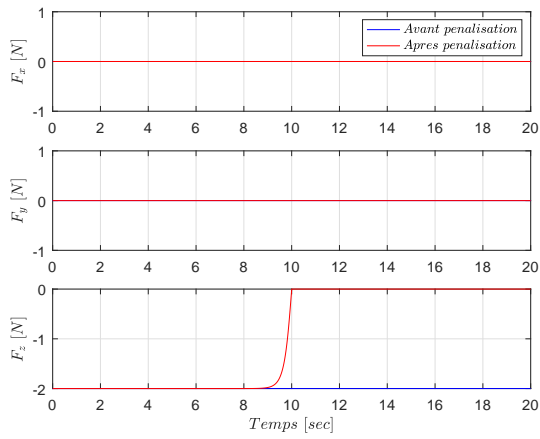
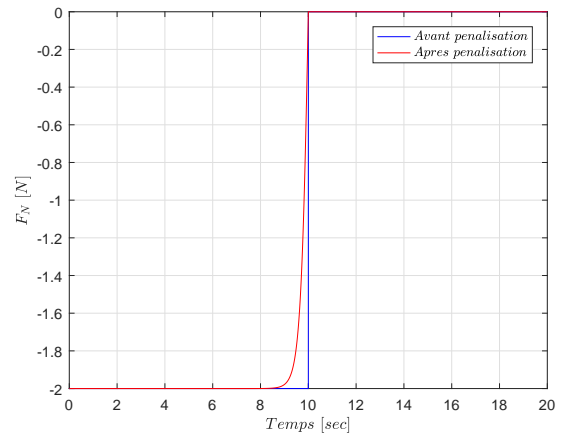


FIGURE 4.28 – Illustration du cas :  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ -2]^T$ .

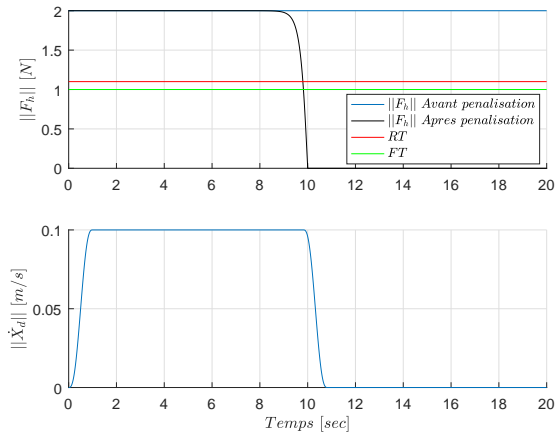
La figure 4.28 illustre que  $F_h$  tend à faire pénétrer le point  $p$  dans l'espace de travail. Par conséquent, l'image  $F_h'$  de  $F_h$  n'est pas nulle dans ce cas (figure 4.29a). Comme  $F_h$  est perpendiculaire à la surface limite ( $\xi_{R_m}$ ), la composante normale  $F_N$  est égal à  $F_z$ . Sur les figures 4.29a et 4.29b, on observe que  $F_h'$  est égal à  $F_h$  entre les instants 0 et 9 sec, car le point  $p$  est suffisamment loin de la surface ( $\xi_{R_m}$ ). La norme  $\|F_h'\|$  pendant cette durée est supérieure à  $RT$ , d'où le fait que l'automate bascule vers l'état *Rise* pour une durée de  $T_{Rise}$ , puis vers l'état *Cruise*. La vitesse désirée est par conséquent, sur cet intervalle de temps n'est pas nulle (figures 4.29c et 4.29f). Après l'instant 9 sec, le point  $X_0$  devient proche de ( $\xi_{R_m}$ ) (figure 4.29d). La pénalisation commence donc à générer une image  $F_h'$  qui tend à devenir nulle lorsque  $p$  est à une distance  $D_F$  de ( $\xi_{R_m}$ ). Après la descente de  $\|F_h'\|$  inférieure à  $FT$ , l'automate bascule vers l'état *Fall* pour une durée  $T_{Fall}$ , puis vers l'état d'arrêt *Stop*. Suite à l'activation du dernier état, la vitesse de  $p$  s'annule, et sa position devient fixe (figures 4.29d et 4.29e). Aussi, à travers les figures 4.29e, 4.29f et 4.30, on constate une continuité sur les profils de position, vitesse et accélération sur les trois axes, pendant toute la durée de simulation.



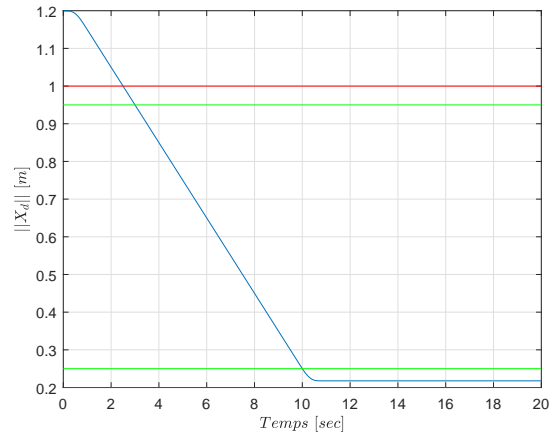
(a)  $F_h$ .



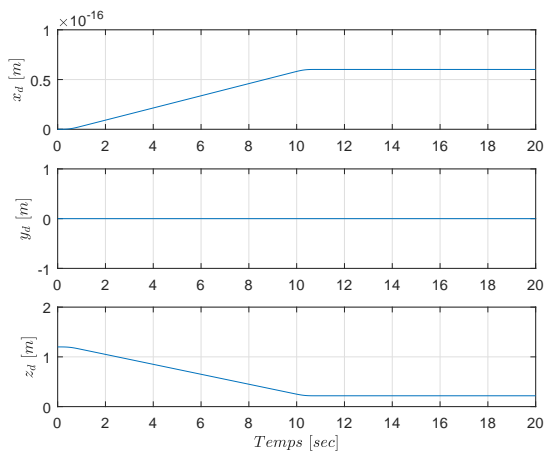
(b)  $F_N$ .



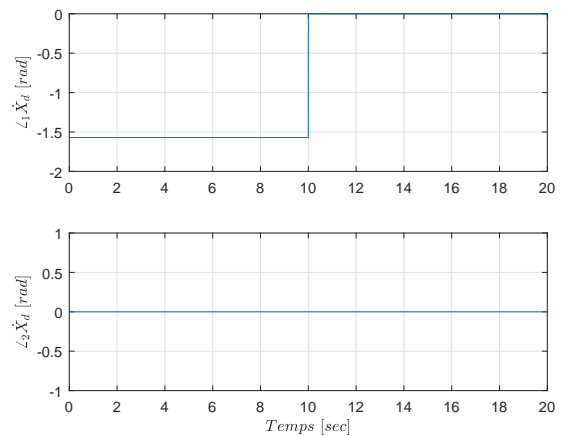
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d$ ,  $y_d$  &  $z_d$ .



(f) Angles de  $\dot{X}_d$ .

FIGURE 4.29 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ -2]^T$ .

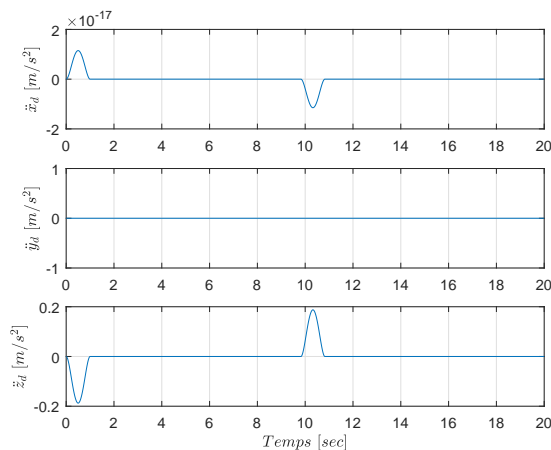


FIGURE 4.30 –  $\ddot{x}_d$ ,  $\ddot{y}_d$  &  $\ddot{z}_d$  pour le cas :  $X_0 = [0 \ 0 \ 1.2]^T$ ,  $F_h = [0 \ 0 \ -2]^T$ .

#### Cas 4 : $X_0 = [0 \ 0 \ 0.5]^T$ , $F_h = [0 \ 0 \ 2]^T$

Pour ce cas, la position initiale  $X_0$  du point  $p$  est sur l'axe  $z_0$  à l'intérieur de l'espace de travail.  $F_h$  est dirigée dans le sens positif de  $z_0$  (figure 4.31).

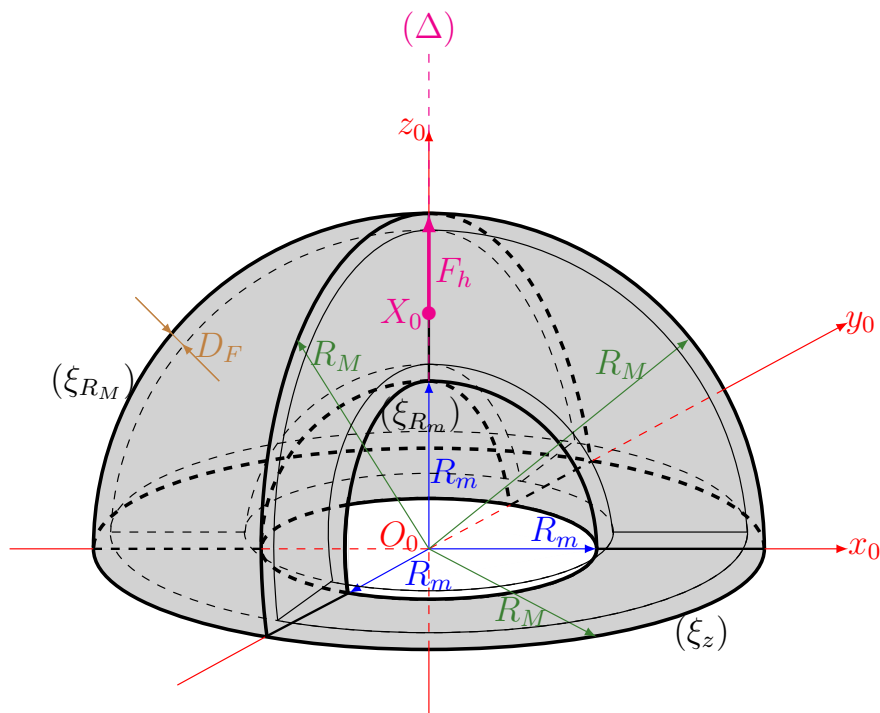


FIGURE 4.31 – Illustration du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

La figure 4.31 illustre que  $F_h$  guide le point  $p$  vers la surface  $(\xi_{R_M})$ . Au début du mouvement, comme le point  $p$  est suffisamment loin de  $(\xi_{R_M})$  (figure 4.32d),  $F_h'$  est égal donc à  $F_h$ , et la norme  $\|F_h'\|$  est supérieure au seuil  $RT$ . Comme pour le cas décrit précédemment, l'automate bascule vers l'état *Rise*, puis *Cruise*. Une vitesse non nulle est par conséquent, générée (figures 4.32c et 4.32f). A partir de l'instant 9

sec, la position de  $p$  est devenue proche de  $(\xi_{R_M})$ . Le fonctionnement du générateur sera donc identique à ce qu'était décrit pour le cas précédent lorsque  $p$  est devenu proche de  $(\xi_{R_m})$ .

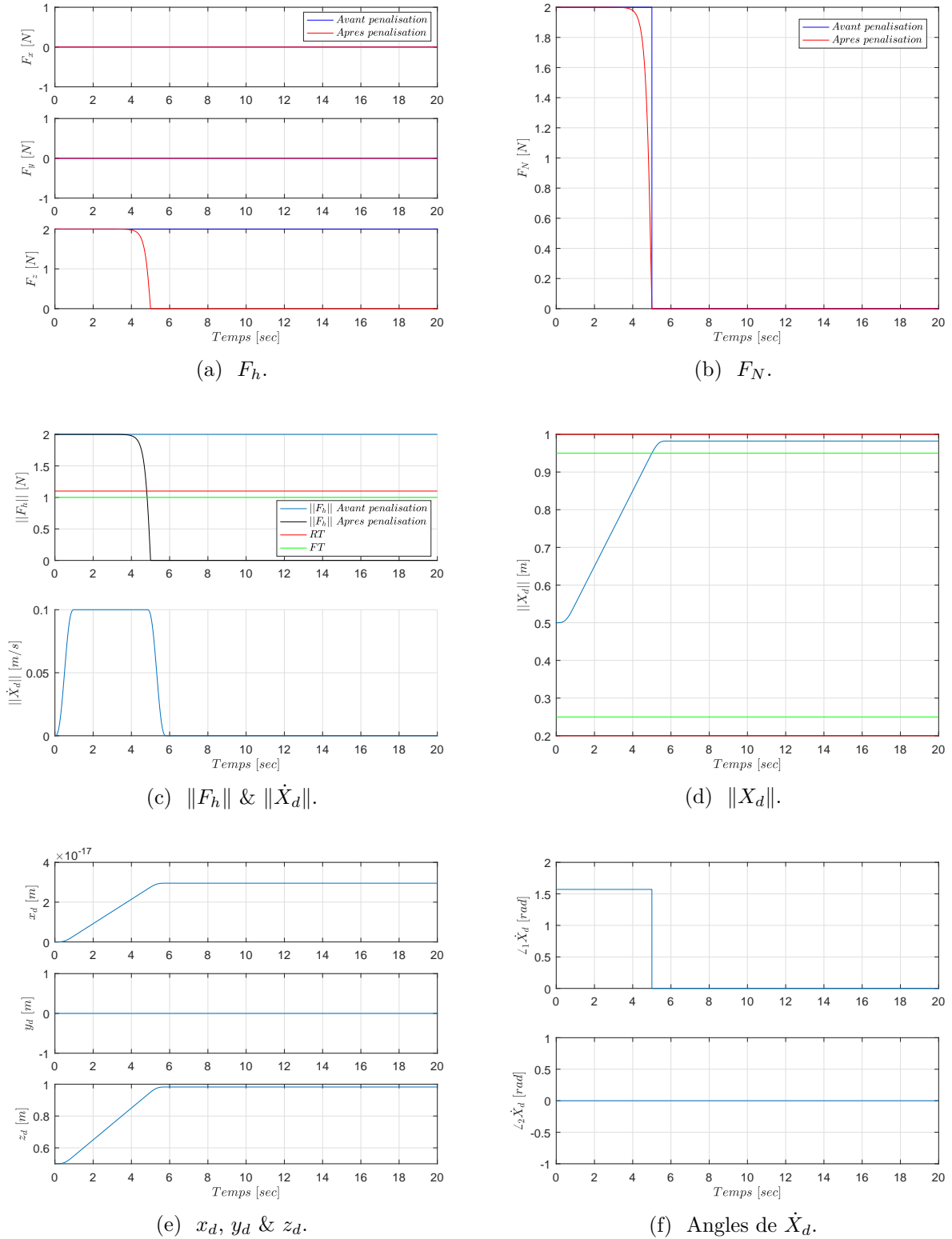


FIGURE 4.32 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

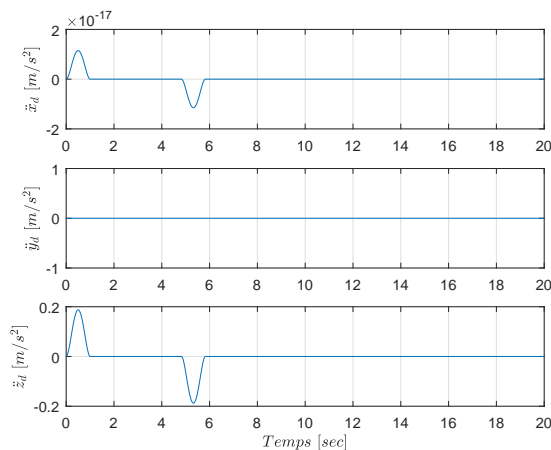


FIGURE 4.33 –  $\ddot{x}_d$ ,  $\ddot{y}_d$  &  $\ddot{z}_d$  pour le cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

**Cas 5 :**  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [2 \ 0 \ 0]^T$

Pour ce cas, la position initiale  $X_0$  de  $p$  est à l'intérieur de  $(\Xi)$ , et  $F_h$  est dirigée selon l'axe  $x_0$  (figure 4.34).

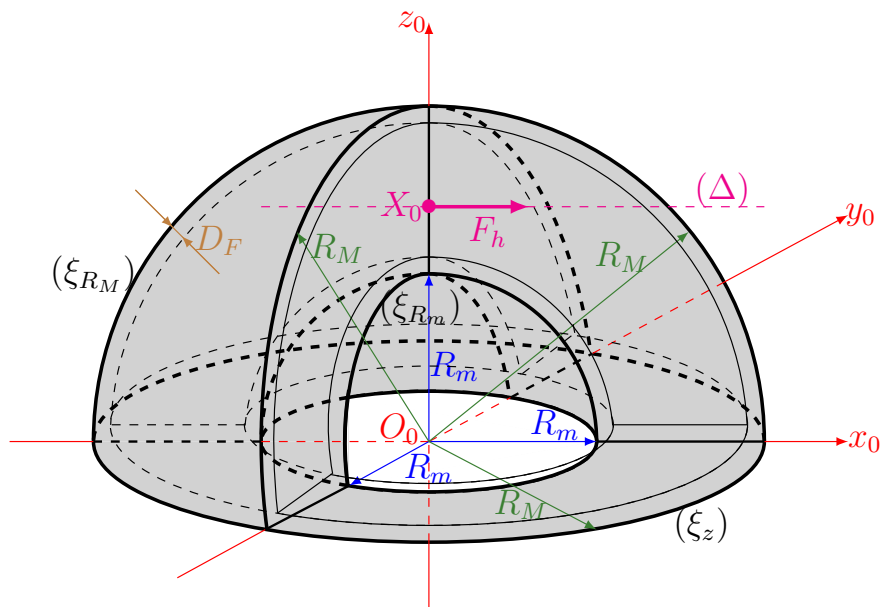
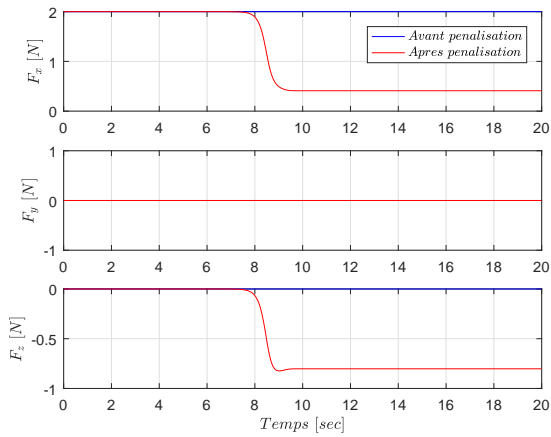
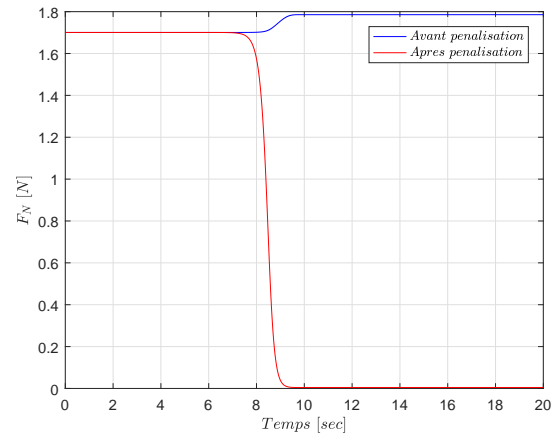


FIGURE 4.34 – Illustration du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [2 \ 0 \ 0]^T$ .

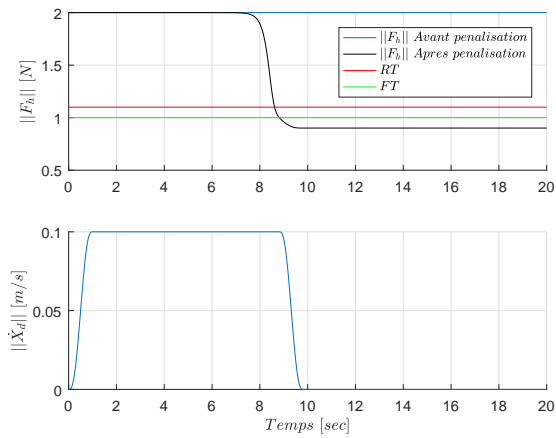
Dans ce cas, la force  $F_h$  guide le point  $p$  vers la surface  $(\xi_{R_M})$ . Après une durée de mouvement du point  $p$  correspondant aux basculements vers les états *Rise* puis *Cruise*,  $p$  devient proche de  $(\xi_{R_M})$ . Par conséquent,  $F_N$  tend vers zéro (figure 4.35b), et il ne reste que la composante tangentielle qui n'est pas nulle. Sa valeur est inférieure à  $FT$  (figure 4.35c), d'où le fait que l'automate bascule vers l'état *Fall*. Pendant la durée de pénalisation de  $F_N$ , le point  $p$  tend vers une surface à  $D_F$  de  $(\xi_{R_M})$  (figure 4.35d). Le mouvement est dirigé vers le bas, car l'influence de  $F'_N$  sur la direction de mouvement est en train de diminuer, jusqu'au l'arrêt (activation de l'état *Stop*).



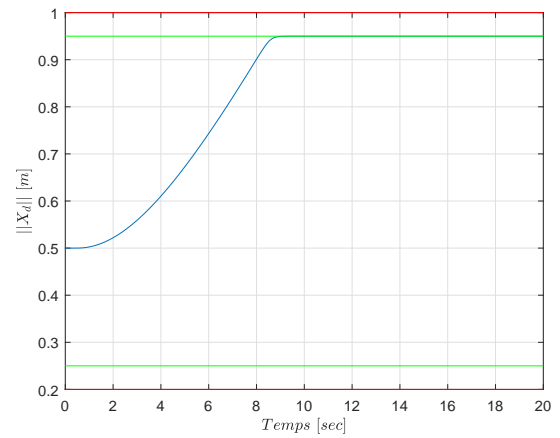
(a)  $F_h$ .



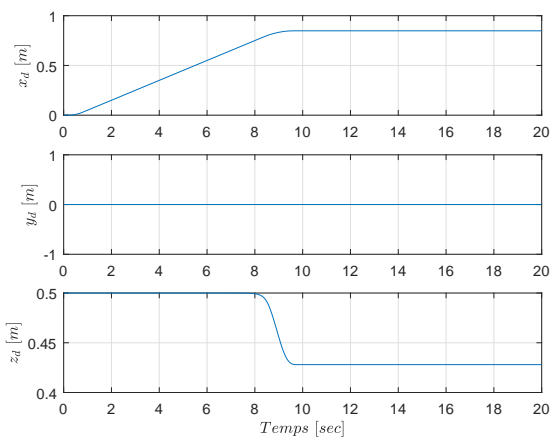
(b)  $F_N$ .



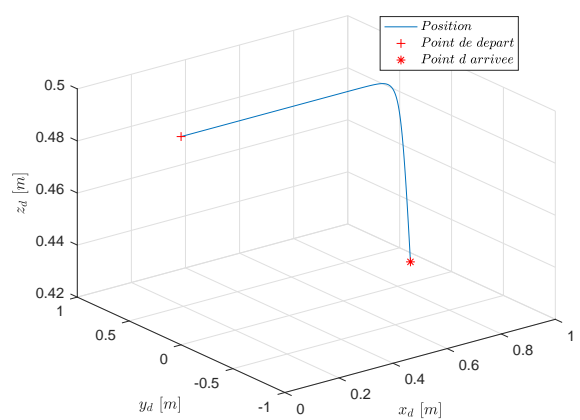
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d$ ,  $y_d$  &  $z_d$ .



(f)  $x_d$ ,  $y_d$  &  $z_d$  en 3D.

FIGURE 4.35 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [2 \ 0 \ 0]^T$ .



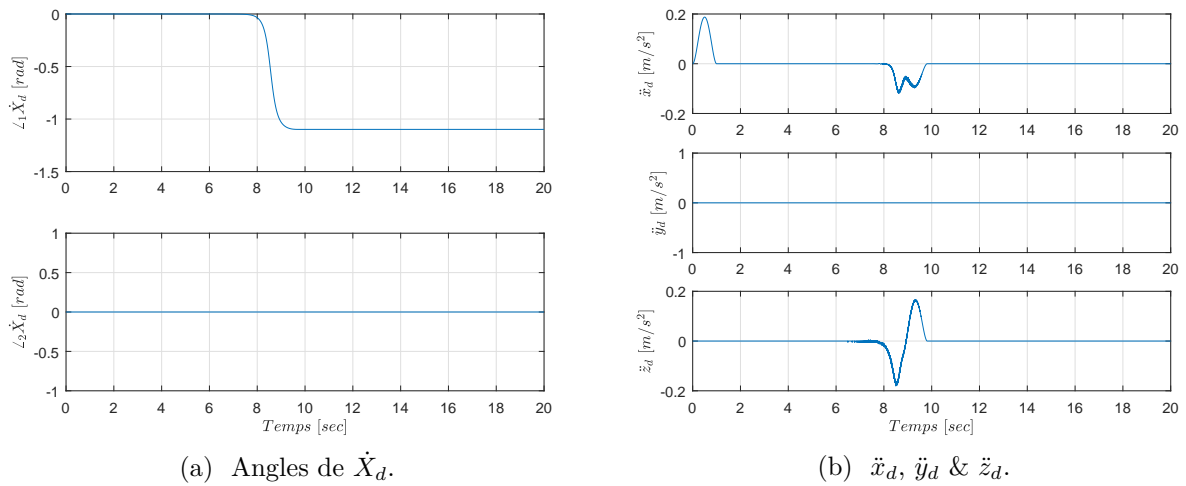


FIGURE 4.36 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [2 \ 0 \ 0]^T$ .

### Cas 6 : $X_0 = [0 \ 0 \ 0.92]^T$ , $F_h = [0 \ 0 \ 2]^T$

Ce cas présente le cas particulier discuté précédemment, où la position initiale  $X_0$  du point  $p$  est à une distance inférieure à  $(D_F + D_R)$  de la surface  $(\xi_{R_M})$ , et la force  $F_h$  est purement normale, et tend à sortir  $p$  de l'espace de travail (figure 4.37).

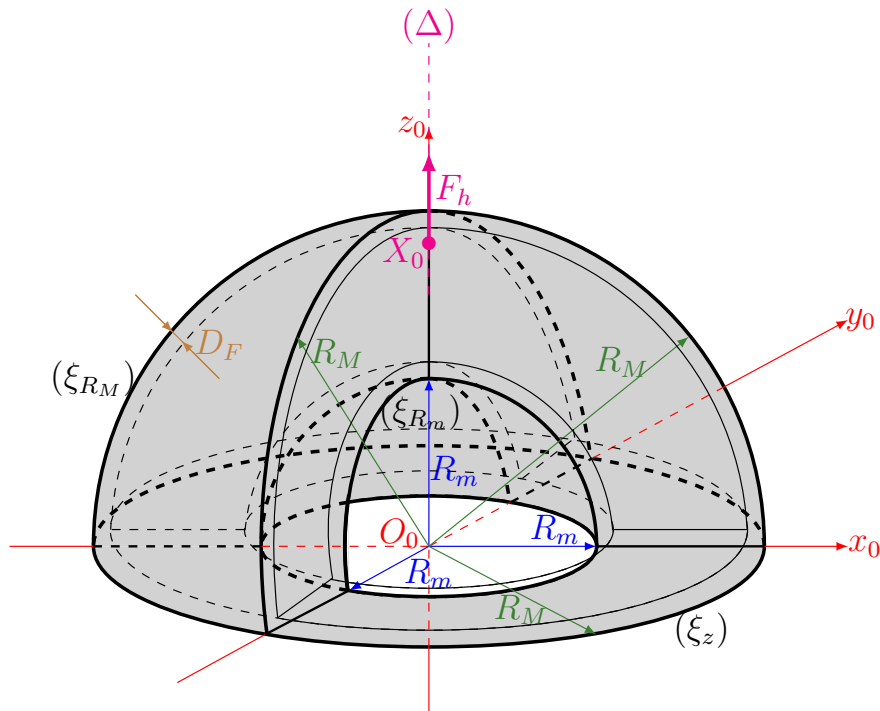
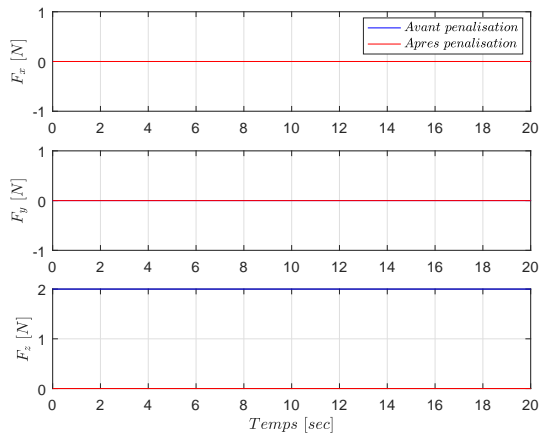
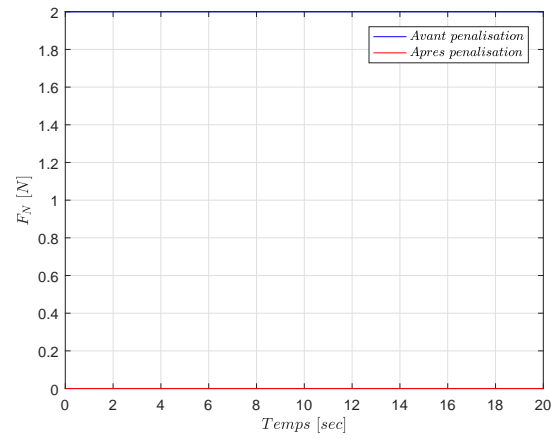


FIGURE 4.37 – Illustration du cas :  $X_0 = [0 \ 0 \ 0.92]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

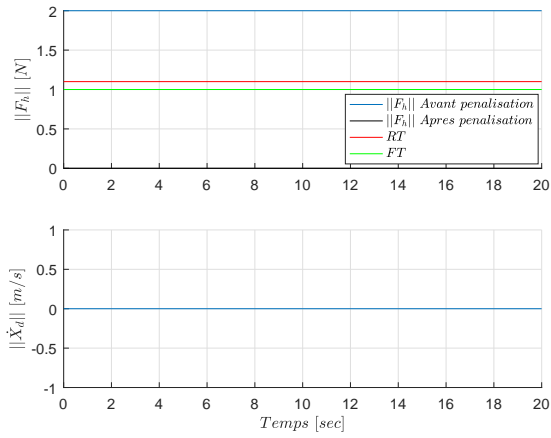
A travers les figures 4.38, on constate que pour ce cas particuliers, l'automate reste à l'état d'arrêt. Donc, la vitesse est nulle, et le point  $p$  reste à la position  $X_0$ .



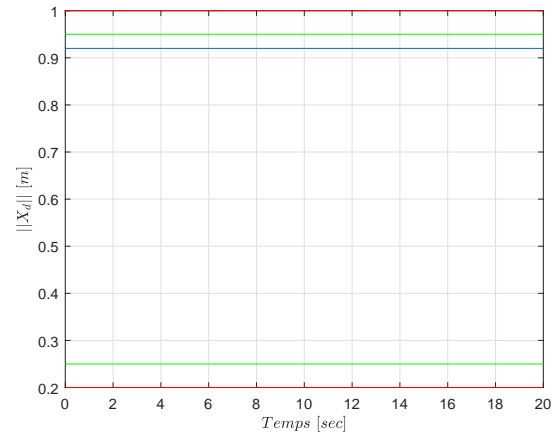
(a)  $F_h$ .



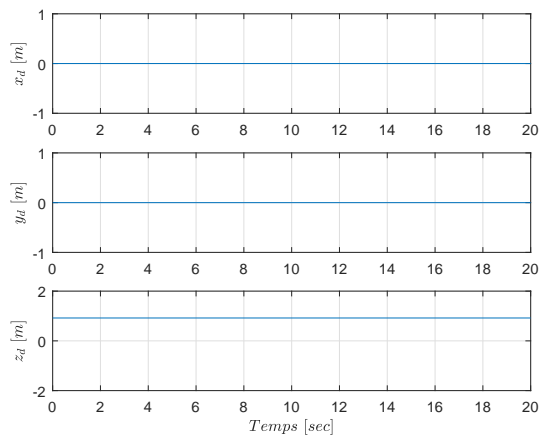
(b)  $F_N$ .



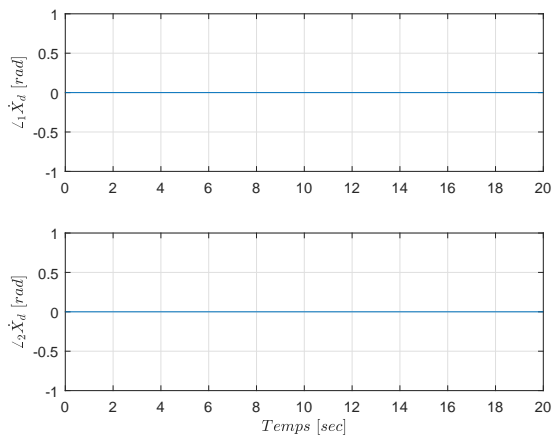
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d$ ,  $y_d$  &  $z_d$ .



(f) Angles de  $\dot{X}_d$ .

FIGURE 4.38 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.92]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

**Cas 7 :**  $X_0 = [0 \ 0 \ 0.95]^T$ ,  $F_h = [0.9487 \ 0 \ -0.9]^T$

Ce cas présente le cas de point d'intersection multiple. A travers la figure 4.39, la droite  $(\Delta)$  portant la force  $F_h$  est en intersection avec les surfaces  $(\xi_{R_M})$  et  $(\xi_z)$  au même temps.

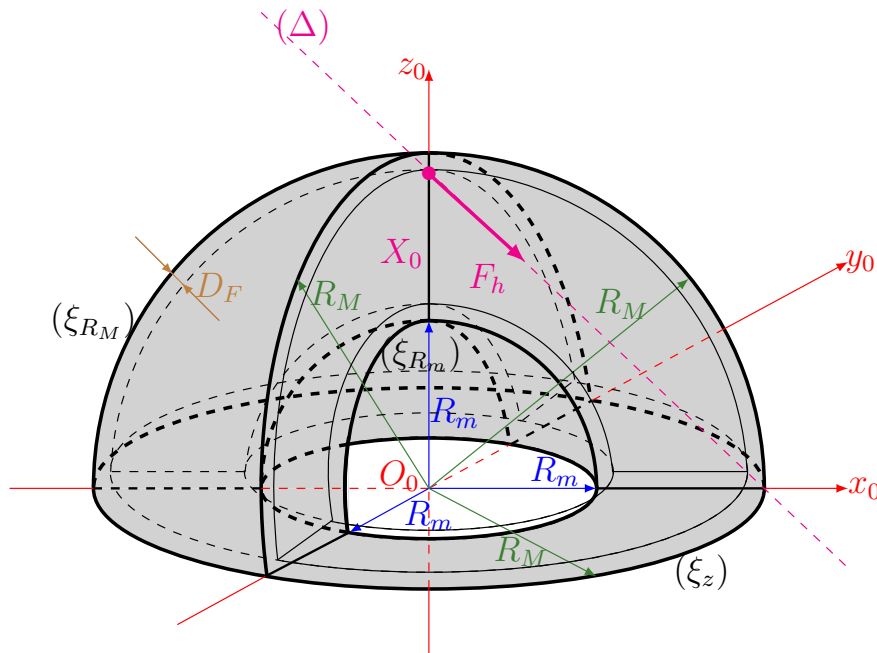
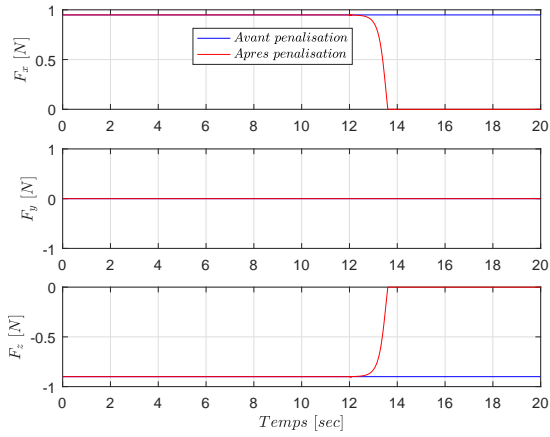
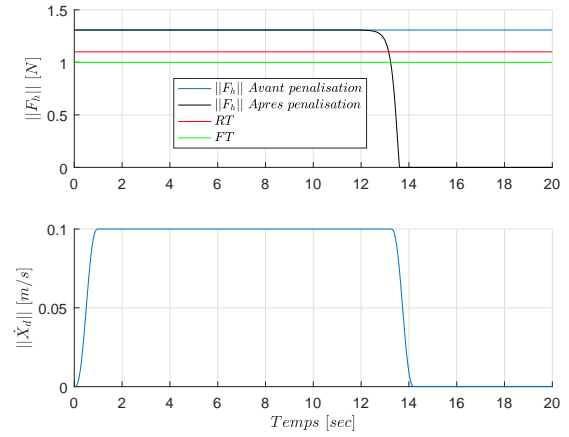


FIGURE 4.39 – Illustration du cas :  $X_0 = [0 \ 0 \ 0.95]^T$ ,  $F_h = [0.9487 \ 0 \ -0.9]^T$ .

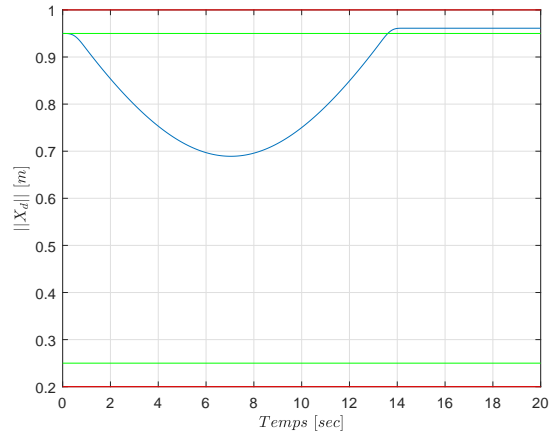
Dans ce cas, toutes les composantes de  $F_h$  sont pénalisées (figure 4.40a). Le point  $p$  est en mouvement suite à l'activation des états *Rise* puis *Cruise* (figure 4.40b). Les composantes de  $F_h$  commencent à être pénalisées quand  $p$  se rapproche du point d'intersection, et  $\|F_h'\|$  commence à diminuer et devenir inférieure à  $FT$ . L'automate bascule donc vers l'état *Fall* puis *Stop*, et le mouvement s'arrête sans dépasser les surfaces  $(\xi_{R_M})$  (figure 4.40c) ni  $(\xi_z)$  (figure 4.40e).



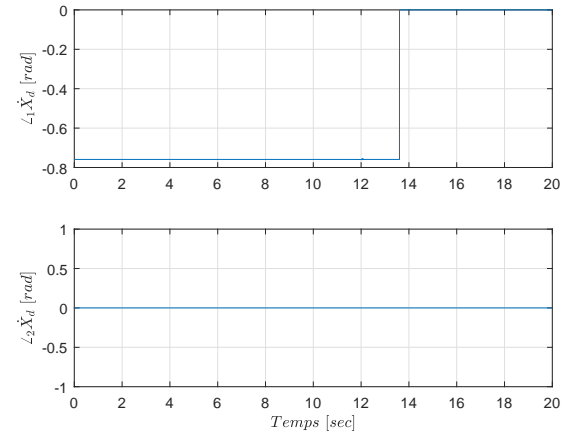
(a)  $F_h$ .



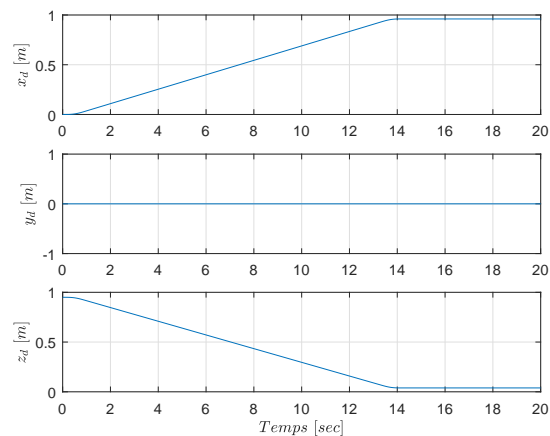
(b)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



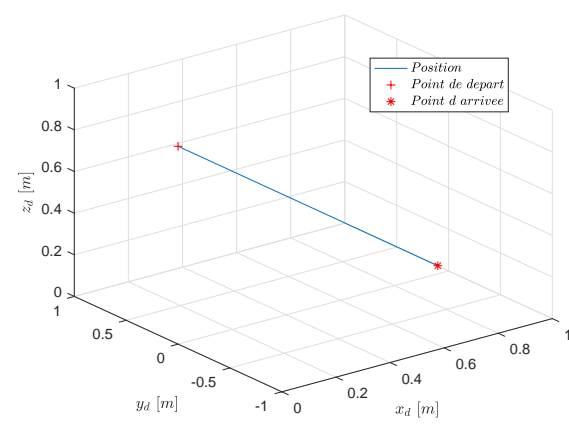
(c)  $\|X_d\|$ .



(d) Angles de  $\dot{X}_d$ .



(e)  $x_d$ ,  $y_d$  &  $z_d$ .



(f)  $x_d$ ,  $y_d$  &  $z_d$  en 3D.

FIGURE 4.40 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.95]^T$ ,  $F_h = [0.9487 \ 0 \ -0.9]^T$ .

**Cas 8 :**  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$

Ce dernier cas est similaire au cas 4, avec un changement rapide au niveau de la norme  $\|F_h\|$  (figure 4.42b) au cours d'activation d'un état de transition (*Rise* dans ce cas).

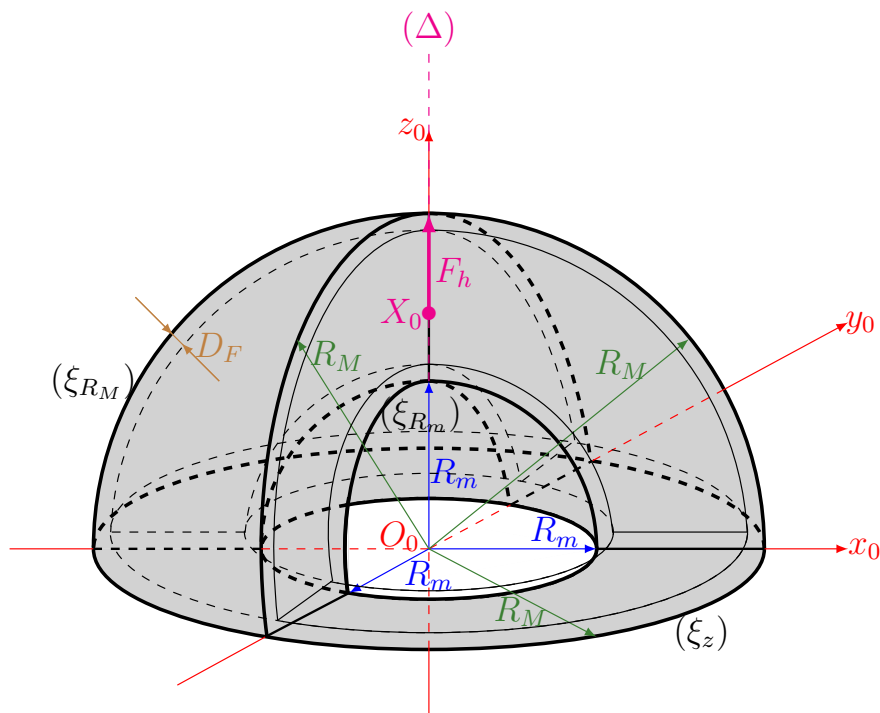
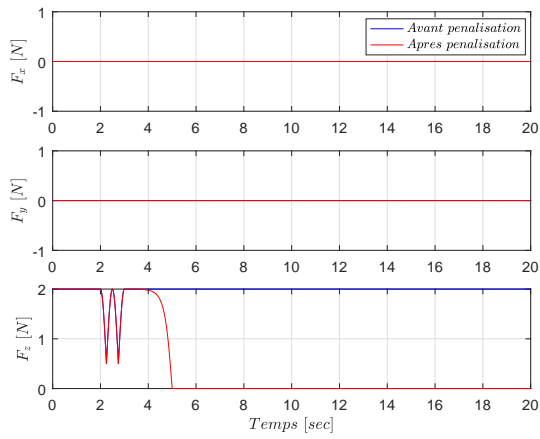
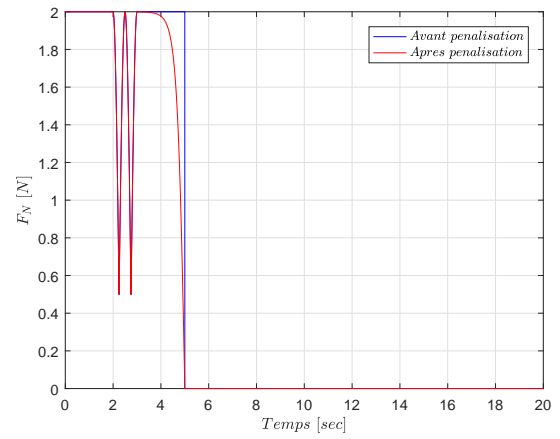


FIGURE 4.41 – Illustration du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

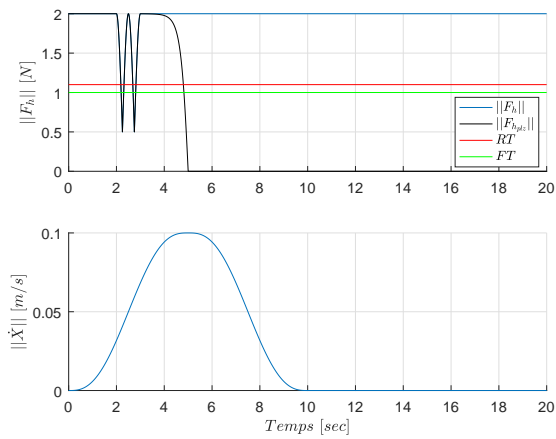
Pour ce cas, nous avons choisi les durées  $T_{Rise}$  et  $T_{Fall}$  égal à 5 sec. Par conséquent, les distances  $D_R$  et  $D_F$  sont plus importantes qui valent 0.25 m (figure 4.42d), tout en gardant la même vitesse de croisière  $V_C$  à 0.1 m/s. Nous remarquons qu'au cours de l'activation de l'état *Rise* suite au franchissement de  $\|F_h'\|$  à  $RT$ , il y a une diminution de cette dernière inférieure à  $FT$ , ce qui normalement déclenche l'état *Fall*. Mais, comme l'état *Rise* n'est pas encore terminé, l'automate garde cet état en activation. Après une durée  $T_{Rise}$ , l'automate bascule vers l'état *Cruise*, tant que  $\|F_h'\|$  est supérieure à  $RT$  maintenant. Pendant un petit moment,  $p$  arrive déjà à une distance  $D_F$  de la surface ( $\xi_{R_M}$ ) (figure 4.42d), d'où l'automate bascule vers l'état *Fall*.



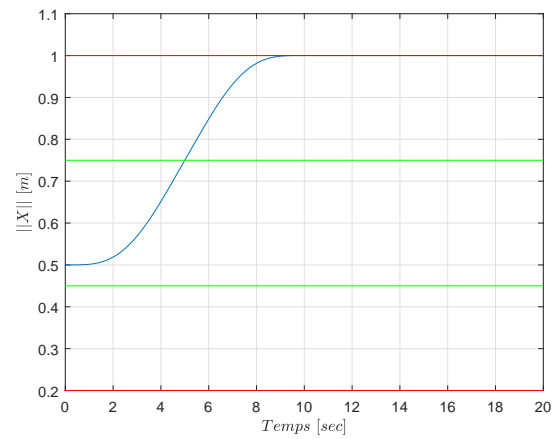
(a)  $F_h$ .



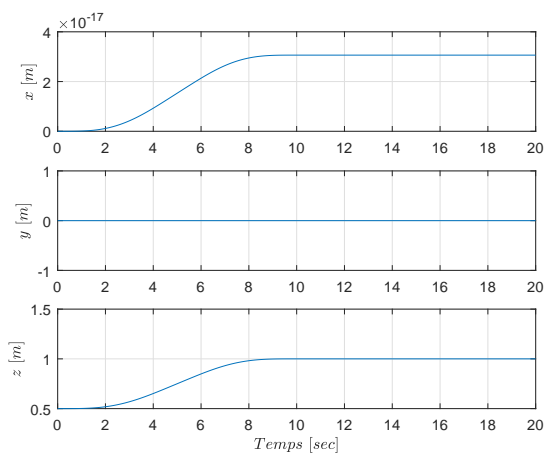
(b)  $F_N$ .



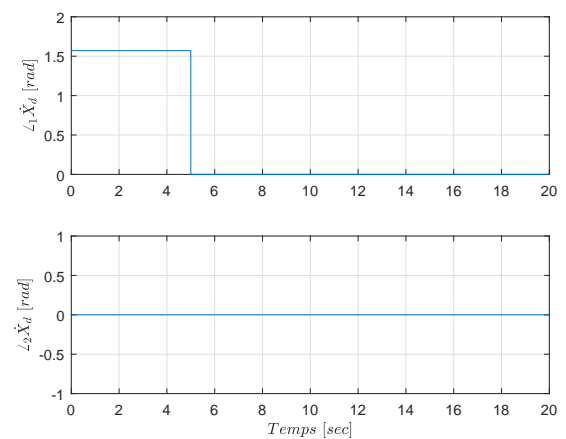
(c)  $\|F_h\|$  &  $\|\dot{X}_d\|$ .



(d)  $\|X_d\|$ .



(e)  $x_d, y_d$  &  $z_d$ .



(f) Angles de  $\dot{X}_d$ .

FIGURE 4.42 – Résultats de simulation du cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

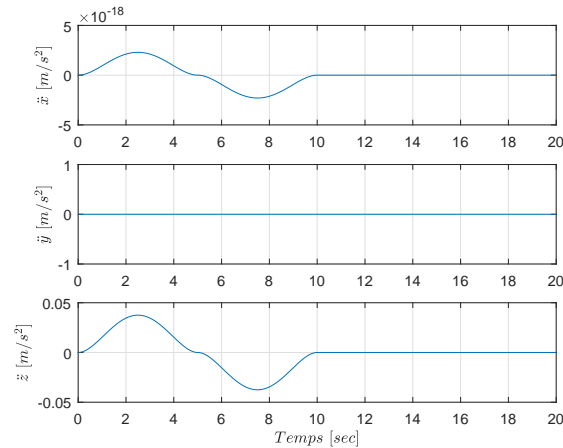


FIGURE 4.43 –  $\ddot{x}_d$ ,  $\ddot{y}_d$  &  $\ddot{z}_d$  pour le cas :  $X_0 = [0 \ 0 \ 0.5]^T$ ,  $F_h = [0 \ 0 \ 2]^T$ .

## 4.5 Conclusion

Dans ce chapitre, nous avons présenté en détail le générateur de trajectoire en temps réel conçu, destiné pour des tâches de co-manipulation homme-robot. Ce dernier reçoit, entre autre, comme entrée la force de l'opérateur estimée à partir de la mesure de la force d'interaction entre l'opérateur et le robot, et l'équation du mouvement de l'organe terminal. Le générateur assure que la co-manipulation soit à l'intérieur de l'espace de travail prédéfini, à travers la pénalisation de la force de l'opérateur estimée. Le générateur contient un automate à quatre états qui permet de gérer l'état du robot, et suite auquel, la trajectoire de la vitesse est calculée. Les transitions entre les états sont fonction soit de la force de l'opérateur, soit de la durée de l'état en cours. Le générateur garantit aussi une vitesse désirée dirigée selon la force de l'opérateur, ce qui confère une transparence à la co-manipulation. L'opérateur par conséquent, ne ressent pas une résistance de la part du robot dans le sens du mouvement, sauf aux limites de la zone opérationnelle. Les trajectoires générées sont essentielles pour le calcul de la commande en impédance, présentée dans le chapitre suivant.

# Commande d'un robot en interaction

## Sommaire

---

<b>5.1 Introduction</b>	<b>127</b>
<b>5.2 Contrôle d'impédance par une commande basée sur l'inversion du modèle dynamique</b>	<b>128</b>
<b>5.3 Simulations &amp; analyse des résultats</b>	<b>131</b>
<b>5.4 Conclusion</b>	<b>142</b>

---

## 5.1 Introduction

Dans ce chapitre, nous allons nous intéresser à l'implémentation en simulation du schéma de commande de la figure 5.1, appliquée à un modèle du robot Denso VP-6242G issu de l'identification proposée au chapitre 3. Pour cela, nous envisageons d'utiliser, pour le suivi de trajectoire, la loi de commande par bouclage la plus répandue, basée sur l'inversion du modèle dynamique du robot [63, 74]. Cette loi de commande présente l'intérêt de donner lieu, en boucle fermée, à une équation de mouvement reliant la force appliquée par l'opérateur, à l'erreur de suivi de trajectoire du point d'interaction dans l'espace opérationnel du robot, qui correspond à celle d'une liaison mécanique de type visco-élastique entre deux masses ponctuelles soumises pour l'une à l'action de l'opérateur, pour l'autre à l'action du robot [44, 45]. Par conséquent, les paramètres de cette relation d'impédance ont une signification physique très claire, et peuvent être ajustés sur la base de considérations pratiques. En contrepartie, cette loi de commande suppose de pouvoir disposer d'un modèle dynamique qui soit précis de manière à ce que les couples produits par l'inversion d'un tel modèle compensent correctement les actions mécaniques correspondantes du robot au cours de son mouvement.

Il existe d'autres lois de commande par bouclage permettant de réaliser un suivi de trajectoire. Cela va du simple Proportionnel-Dérivée sur les coordonnées articulaires, à la commande basée sur la passivité des équations de mouvement d'un robot rigide, voire à la commande adaptative. Nous pourrions nous référer à [63, 74] pour une description des principales lois de commande envisageables. Quelle que soit la loi de commande envisagée, elles nécessitent toutes de connaître, avec une bonne précision,



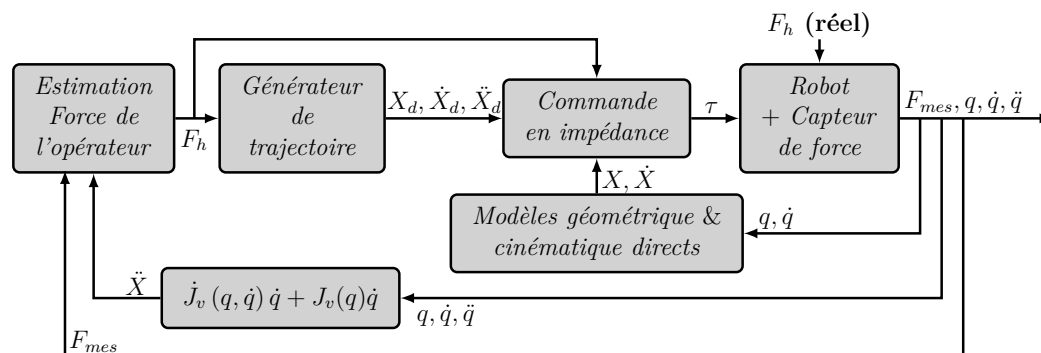


FIGURE 5.1 – La stratégie de commande en impédance étudiée.

le terme dû à la gravité dans les équations de mouvement du robot données par (3.53), voire les termes dûs aux actions des forces centrifuges et de Coriolis. Ces termes correspondent respectivement à  $G(q)$  et  $N(q, \dot{q})\dot{q}$  dans les équations d'Euler-Lagrange données par (3.53).

Comme nous allons le voir dans ce chapitre à la section 5.3, le MDD issu du MDI correspondant, provenant de l'identification du chapitre 3, permet de présenter des résultats relativement concluants dans les simulations de mise en œuvre du schéma de commande de la figure 5.1, lorsque nous exploitons ce modèle pour construire et analyser la loi de commande par bouclage basée sur l'inversion du modèle dynamique. Mais une analyse par simulation ne peut se substituer à une validation expérimentale sur le robot réel, car rien ne prouve que la commande basée sur l'inversion du modèle dynamique identifié pourra, lors d'essais expérimentaux, réaliser « l'inversion attendue » du comportement dynamique du robot réel.

Le reste du chapitre est organisé de la manière suivante. La section 5.2 présentera le principe du contrôle de l'impédance mécanique et la manière dont cela est réalisé en robotique, dans la perspective d'un suivi de trajectoire. Après avoir présenté l'application du schéma de commande de la figure 5.1 utilisant cette loi de commande dans la section 5.3, nous proposerons quelques éléments de conclusion pour ce chapitre.

## 5.2 Contrôle d'impédance par une commande basée sur l'inversion du modèle dynamique

Le contrôle d'impédance est un principe de contrôle qui permet de caractériser l'erreur de poursuite en position, et même en vitesse et en accélération, qui est cause de l'exercice d'une force extérieure sur le robot. Selon les objectifs du contrôle, le contrôle d'impédance peut être en espace opérationnel ou bien articulaire. Afin de réaliser la poursuite des trajectoires désirées de l'organe terminal, élaborées par le générateur de trajectoire conçu pour des tâches de co-manipulation, nous proposons de concevoir une stratégie de contrôle d'impédance par une commande basée sur l'inversion du modèle dynamique, dans l'espace opérationnel, illustrée par le schéma de la figure 5.2.

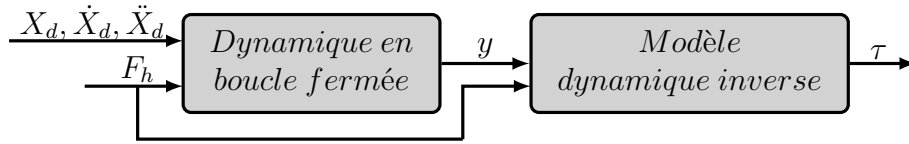


FIGURE 5.2 – Stratégie du contrôle d'impédance.

### 5.2.1 Loi de commande basée sur l'inversion du modèle dynamique

Notons par  $X$ ,  $\dot{X}$  et  $\ddot{X}$  les variables de position, de vitesse et d'accélération du point d'interaction exprimés dans le repère de base. Notons par  $X_d$ ,  $\dot{X}_d$  et  $\ddot{X}_d$  les variables de position, de vitesse et d'accélération désirées pour ce point. Rappelons que les variables  $X_d$ ,  $\dot{X}_d$  et  $\ddot{X}_d$  sont générées en temps réel par le générateur de trajectoire en ligne. Soient :

$$\begin{aligned}\tilde{X} &:= X_d - X, \\ \dot{\tilde{X}} &:= \dot{X}_d - \dot{X}, \\ \ddot{\tilde{X}} &:= \ddot{X}_d - \ddot{X},\end{aligned}$$

les erreurs de poursuite en position, vitesse et accélération du point d'interaction respectivement. La dynamique désirée dans l'espace opérationnel s'écrit comme une fonction des erreurs de poursuite et de la relation d'impédance caractérisant l'interaction [27], exprimée comme suit :

$$M_d \ddot{\tilde{X}} + K_D \dot{\tilde{X}} + K_P \tilde{X} = F_h, \quad (5.1)$$

où  $M_d$ ,  $K_D$  et  $K_P \in \mathbb{R}^{3 \times 3}$  sont des matrices définies positives et diagonales [74], qui représentent des coefficients de masse, d'amortissement visqueux et de raideur, respectivement. Ces matrices permettent d'ajuster la relation d'impédance désirée pour caractériser l'interaction homme-robot.

Le vecteur  ${}^0t_n$  du modèle géométrique direct (équations (3.7) et (3.9)) permet d'exprimer la position de l'organe terminal dans le repère fixe  $R_0$  en fonction des positions articulaires  $q$  :

$$X = {}^0t_n = f(q). \quad (5.2)$$

Rappelons que l'équation (3.15) permet d'exprimer la vitesse de l'organe terminal  $\dot{X}$  dans  $R_0$  :

$$\dot{X} = {}^0V_{O_n} = J_v(q)\dot{q}. \quad (5.3)$$

En dérivant l'expression (5.3) par rapport au temps, on trouve l'accélération de l'organe terminal dans  $R_0$  :

$$\ddot{X} = J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q} \quad (5.4)$$

En remplaçant la relation (5.4) dans l'équation (5.1), après développement, la dynamique en boucle fermée est donnée par l'équation suivante :

$$y = \ddot{q} = J_v^+(q) \left[ \ddot{X}_d - \dot{J}_v(q, \dot{q}) \dot{q} + M_d^{-1} (K_D \dot{X} + K_P \tilde{X} - F_h) \right]. \quad (5.5)$$

La matrice jacobienne  $J_v(q)$  n'étant pas une matrice carrée puisqu'elle est ici de dimension  $(3 \times 6)$ , le calcul dans l'équation (5.5) fait alors intervenir la pseudo-inverse de Moore-Penrose de  $J_v(q)$ . Cela permet ainsi d'éliminer tout problème de singularité de la matrice  $J_v(q)$ .

Rappelons par ailleurs la relation (3.53) qui donne le modèle dynamique du robot :

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = \tau + \tau_f + J_v^T(q)F_{ext}. \quad (5.6)$$

Pour une tâche de co-manipulation homme-robot, nous supposons que la seule force extérieure au robot, appliquée à l'organe terminal est la force de l'opérateur :

$$F_{ext} = F_h.$$

A partir de l'équation (5.5), et en se basant sur le modèle dynamique du robot (5.6), la loi de commande basée sur l'inversion du modèle dynamique est donnée par la relation :

$$\tau = M(q)y + N(q, \dot{q})\dot{q} + G(q) - \tau_f - J_v^T(q)F_h. \quad (5.7)$$

## 5.2.2 Modulation des paramètres de l'opérateur d'impédance

En plus de l'implémentation de la contrainte portant sur la « limite de l'espace de travail » dans le générateur de trajectoire, nous proposons de moduler les paramètres de l'opérateur d'impédance de l'interaction homme-robot en fonction de la distance minimale  $d$  définie dans l'équation (4.50). Cette modulation consiste essentiellement à pondérer les paramètres de raideur de l'impédance  $K_P$ , dans la loi de commande (5.5), par une fonction  $f_{K_P}(d)$  décroissante en  $d$ , définie pour  $d \geq 0$ . Cette pondération est donnée par la fonction suivante :

$$f_{K_P}(d) = H(H(d) - \gamma_2) \left[ 2(\Lambda - 1) \left( 1 - 2 \frac{\arctg((d+1)^\eta)}{\pi} \right) + 1 \right], \quad (5.8)$$

où  $f_{K_P}(d=0) = \Lambda > 1$  et  $\gamma_2 \in \left] 0, \frac{1}{2} \right[$ . Cela permet de rendre très rigide l'interaction homme-robot lorsque le point d'interaction se situe sur la surface limitante ( $\xi$ ). Le paramètre matriciel  $K_P$  de la loi de commande (5.5) est alors remplacé par la matrice diagonale semi-définie positive  $K_P'$ , donnée par :

$$K_p' = f_{K_P}(d)K_P. \quad (5.9)$$

La figure 5.3 montre le profil de  $f_{K_P}$  en fonction de  $d$ .

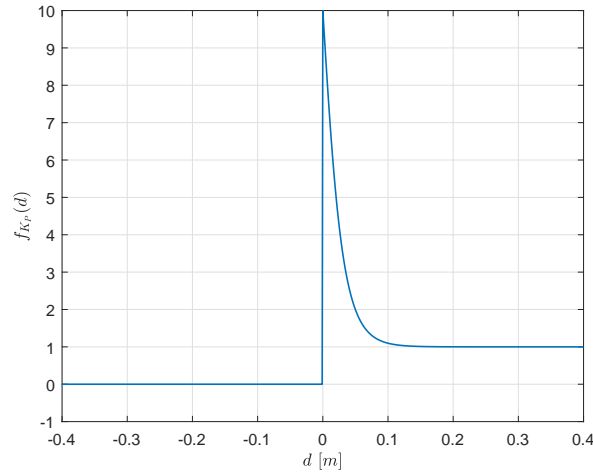


FIGURE 5.3 – Profil de  $f_{K_P}(d)$  pour  $\Lambda = 10$  et  $\eta = 50$ .

Cette figure 5.3 illustre le fait que la fonction  $f_{K_P}$  tend vers  $\Lambda > 1$  lorsque  $X_d$  se rapproche de la surface limite, c'est-à-dire lorsque  $d \rightarrow 0$ . L'augmentation de la raideur sur la surface ( $\xi$ ) permet de diminuer l'erreur de position, et donc de diminuer des éventuels dépassements de la position réelle du robot,  $X$ , vis-à-vis de la surface limite.

La dépendance de la nouvelle matrice de raideur  $K'_P$  au paramètre  $d$  se traduit par une dynamique désirée vérifiant l'équation de mouvement actualisée donnée par

$$M_d \ddot{\tilde{X}} + K_D \dot{\tilde{X}} + K'_P \tilde{X} = F_h. \quad (5.10)$$

Or,  $d$  étant dépendant de  $F_h$ , comme indiqué dans le chapitre précédent, le paramètre  $K'_P$  évolue alors au cours du temps. L'équation de mouvement en l'erreur  $\tilde{X}$  qui en résulte, n'est plus une équation linéaire stationnaire pour laquelle l'analyse de la stabilité au sens entrée-bornée-sortie-bornée serait conduite par l'examen des racines de l'équation caractéristique associée. L'analyse de la stabilité de l'équation de mouvement actualisée (5.10) doit être examinée avec attention, avec les outils d'étude de la stabilité des systèmes non-linéaires, pour pouvoir conclure sur la stabilité du suivi de trajectoire, par exemple au sens de Lyapunov.

## 5.3 Simulations & analyse des résultats

Nous avons fait une simulation du schéma de commande détaillé précédemment, avec le modèle identifié le plus abouti du robot Denso VP-6242G, obtenu et présenté dans le chapitre 3. La simulation a été faite sur un intervalle de temps de 20 secondes. A l'instant initial, le vecteur des positions articulaires correspondait à  $q_0 = [0, 22.5, -65, 0, -90, 0]^T$  [°], tandis que les vitesses articulaires étaient toutes nulles. La position et la vitesse initiales de l'organe terminal (modèles géométrique et cinématique directs) étaient donc  $X_0 = [0.0578, 0, 0.4772]^T$  [m], et  $\dot{X}_0 = [0, 0, 0]^T$  [m/sec]. Les paramètres de la loi de commande (équation (5.5)) ont été choisis de la manière suivante :  $M_d = 1 E_3$ ,  $K_D = 10 E_3$  et  $K_P = 1000 f_{K_P} E_3$ , où  $E_3 \in \mathbb{R}^{3 \times 3}$  est la matrice identité de dimension 3. Les paramètres de la fonction  $f_{K_P}$  sont  $\Lambda = 10$  et

$\eta = 50$ . Les paramètres cinématiques et dynamiques du robot ont déjà été présentés dans la section 3.6. Le reste des paramètres de la simulation sont les suivants. La constante de gravité vaut  $g = 9.81 [m/s^2]$ . Le pas d'échantillonnage est imposé par le système de contrôle du robot et vaut  $\Delta t = 10^{-3} s$ . Le seuil de mise en mouvement (Rise Threshold) et celui d'arrêt du mouvement (Fall Threshold) sont respectivement  $RT = 1.1 [N]$  et  $FT = 1 [N]$ . Les durées des états *Rise* et *Fall* sont  $T_{Rise} = 1 [s]$ ,  $T_{Fall} = 1 [s]$ . La vitesse de croisière  $V_C = 0.1 [m/s]$ . La simulation démarre dans le cas où l'état initial du robot est à *Stop*.

Nous avons pris un espace de travail du robot limité extérieurement par une demi-sphère ( $z > 0$  dans  $R_0$ ) de rayon  $R_M = 0.55 [m]$ , et intérieurement par une demi-sphère ( $z > 0$  dans  $R_0$ ) de rayon  $R_m = 0.38 [m]$ . La position initiale  $q_0$  est choisie de manière à être à l'intérieur de l'espace de travail. Les distances  $D_R$  et  $D_F$  sont calculées par intégration numérique du profil de la norme de la vitesse (équation (4.47) et tableau 4.1) sur les durées  $T_{Rise}$  et  $T_{Fall}$ , respectivement :  $D_R = D_F = 0.05 [m]$ . Les durées  $T_{Rise}$  et  $T_{Fall}$  dépendent de l'environnement de la co-manipulation, ainsi que des capacités de l'opérateur et de celles du robot [44]. Pour nos simulations, ces durées sont choisies à titre indicatif.

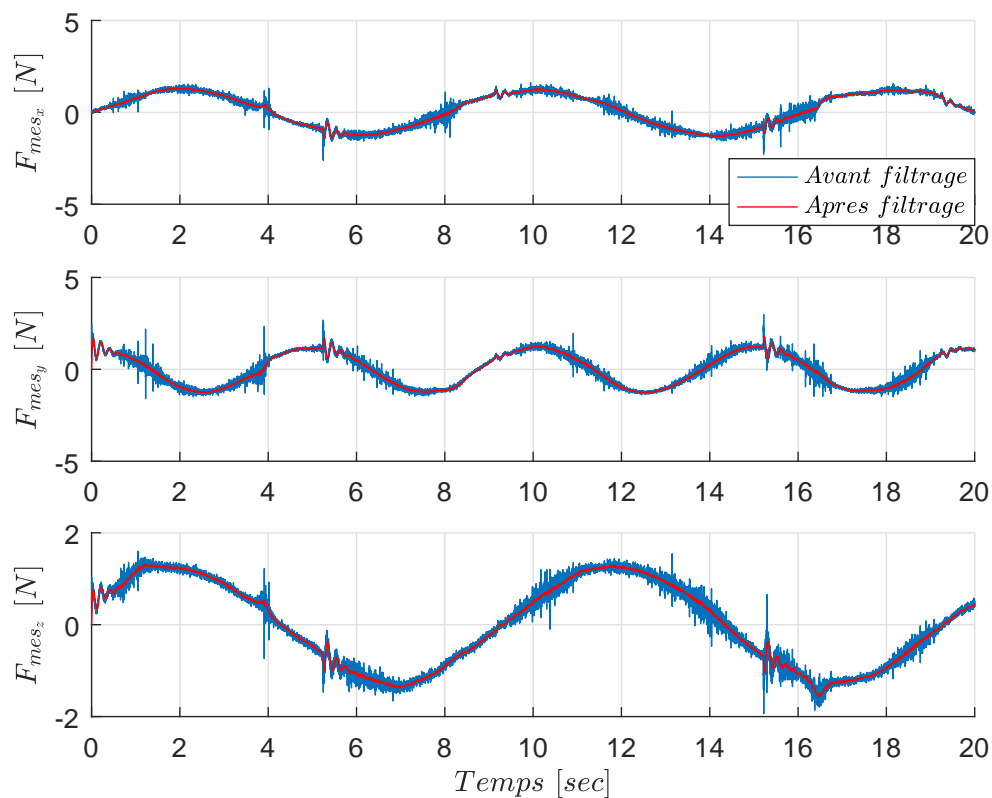
Les seuils  $RT$  et  $FT$  de la force de l'opérateur sont pris suffisamment petits par rapport à l'effort appliqué par l'opérateur. Ce choix doit permettre d'assurer que l'opérateur n'exerce pas un effort trop important. Cependant, il est nécessaire que les seuils soient détectables par le capteur de force. Les valeurs  $RT$  et  $FT$  sont différentes afin de créer un hystérésis dans l'interprétation de mise en mouvement et d'arrêt du mouvement. En effet, dans le cas où  $\|F'_h\| = RT = FT$  l'effort de l'opérateur peut enclencher de manière indésirée les états de mise en mouvement et d'arrêt du mouvement. Pour cela,  $RT$  est pris supérieur à  $FT$  car l'inverse introduirait un intervalle d'effort non défini lorsque l'amplitude de la force de l'opérateur se situerait entre ces deux seuils.

Les matrices  $M_d$ ,  $K_D$  et  $K_P$  permettent de régler l'impédance ressentie par l'opérateur lorsque le robot poursuit la trajectoire générée [44]. Les valeurs de ces matrices sont prises afin d'avoir une bonne qualité de poursuite des trajectoire de référence. L'implémentation expérimentale devrait nous permettre de régler ces valeurs selon l'impédance ressentie et la qualité de poursuite en pratique.

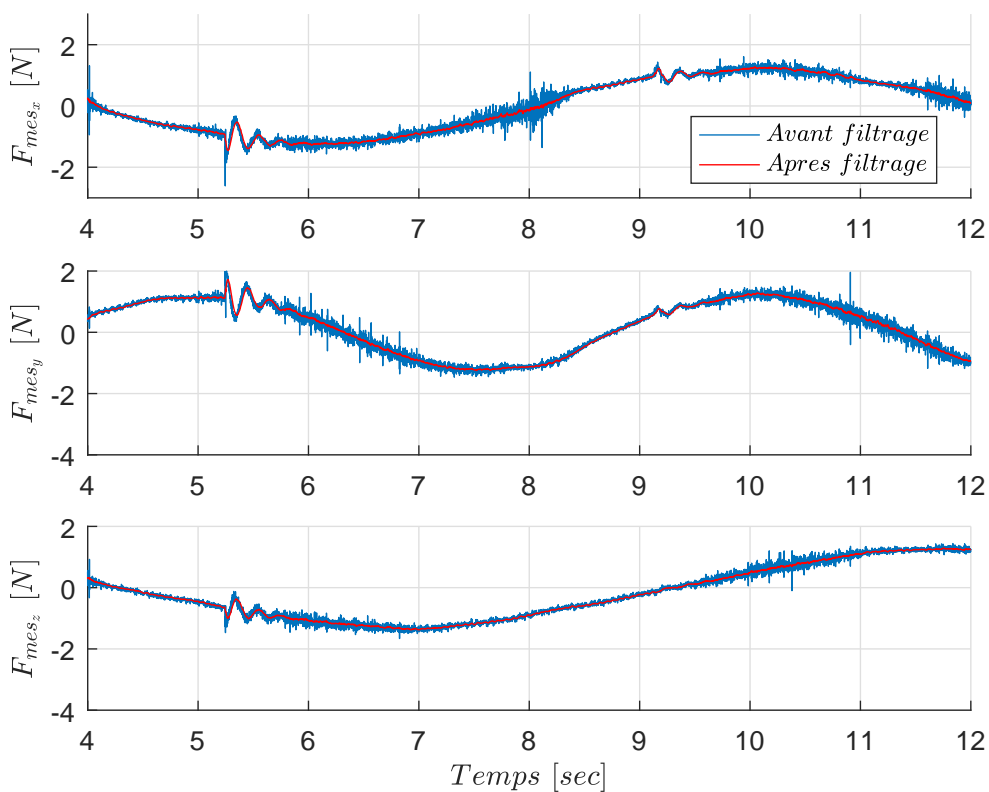
La valeur de la vitesse de croisière  $V_C$  est choisie en respectant la norme ISO-10218 qui spécifie les exigences de sécurité pour les robots industriels [14]. Selon cette norme, la vitesse de l'organe terminal du robot pour une interaction physique homme-robot est limitée à  $0,25 [m/s]$ .

Pour cette simulation, nous supposons que la force mesurée est soumise à un bruit gaussien de moyenne nulle et de variance suffisamment grande pour être plausible et correspondre au bruit observer sur le capteur de force de notre dispositif expérimental.

La figure 5.4a montre le profil de la force mesurée par le capteur de force, avant (en bleu) et après filtrage (en rouge). La figure 5.4b correspond à un zoom entre les instants 4 et 12 *sec* sur le même signal, toujours avant et après filtrage.



(a) Intervalle temporel complet de simulation.



(b) Zoom entre les instants 4 et 12 sec.

FIGURE 5.4 – Force mesurée avant (bleu) et après (rouge) filtrage.

Les trois composantes de  $F_{mes}$  ont été filtrées par un filtre passe-bas du premier ordre, avec un gain unitaire et une fréquence de coupure de 10 [Hz]. La fréquence de coupure est considérée largement plus grande par rapport aux fréquences contenues dans  $F_{mes_x}$  (0.125 [Hz]),  $F_{mes_y}$  (0.2 [Hz]) et dans  $F_{mes_z}$  (0.1 [Hz]). Nous pouvons constater que  $F_{mes}$  est correctement filtrée.

La figure 5.5 montre le profil de la force  $F_h$  estimée avant (en bleu) et après pénalisation (en rouge). La force  $F_h$  a été estimée à partir de la mesure filtrée de  $F_{mes}$  et du modèle de capteur de force indiqué dans (4.25).

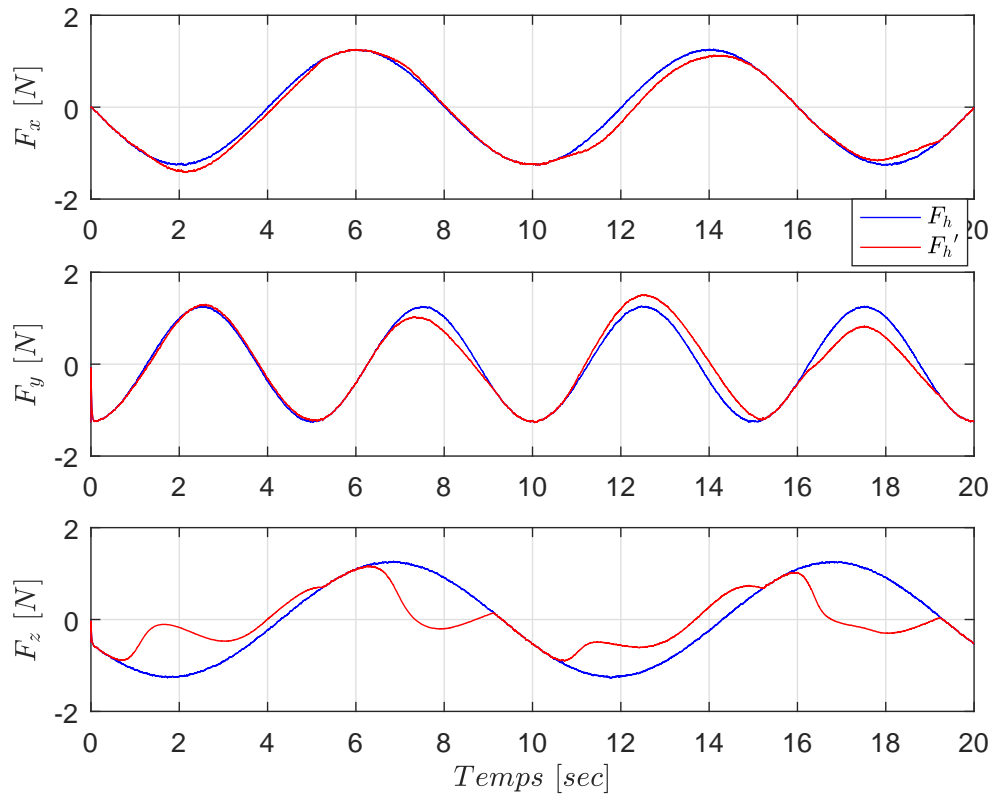
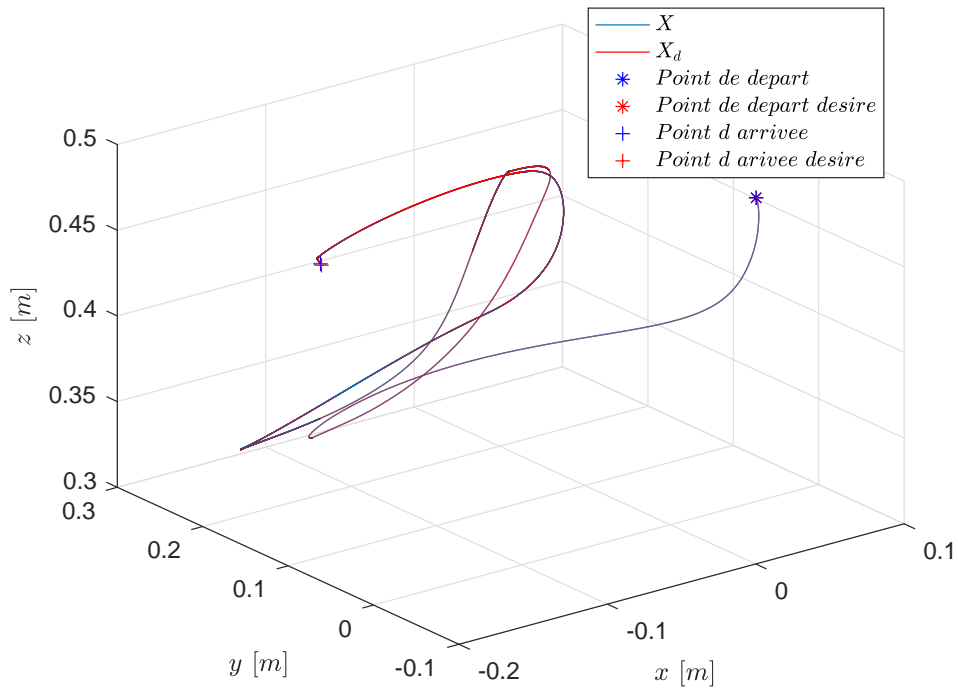


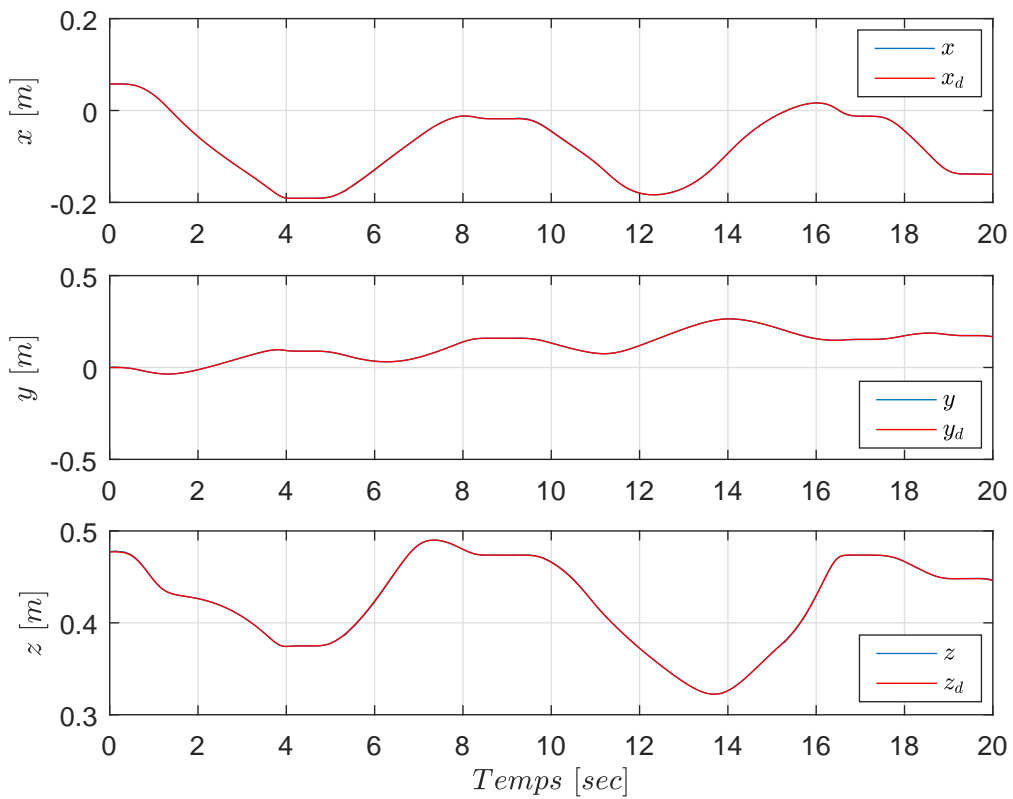
FIGURE 5.5 – Force  $F_h$  estimée, avant et après pénalisation.

Les parties non confondues des signaux de  $F_h$  avant et après pénalisation signifient que le robot pendant ces intervalles est proche d'une butée virtuelle, ce qui a impliqué une pénalisation de la force  $F_h$ .

La figure 5.6a représente la trajectoire de position de l'organe terminal dans l'espace et sa trajectoire de référence, dans le repère fixe  $R_0$ . La figure 5.6b représente la position de l'organe terminal selon les trois axes du repère  $R_0$ .



(a) Évolution dans l'espace tridimensionnel.



(b) Évolution temporelle suivant chacun des trois axes.

FIGURE 5.6 – Position de l'organe terminal.



Les figures 5.6a et 5.6b illustrent une bonne qualité de poursuite de la trajectoire de position, selon les trois directions  $x$ ,  $y$  et  $z$ . Cette qualité dépend essentiellement du choix des coefficients de la matrice  $K_P$ . Cependant, des valeurs trop élevées de  $K_P$  augmentent la rigidité de l'interaction. La mise en pratique nous permettra de mieux réaliser le compromis qualité de poursuite-souplesse de la co-manipulation.

La figure 5.7 représente la norme de la position de l'organe terminal (bleu) avec sa référence (rouge).

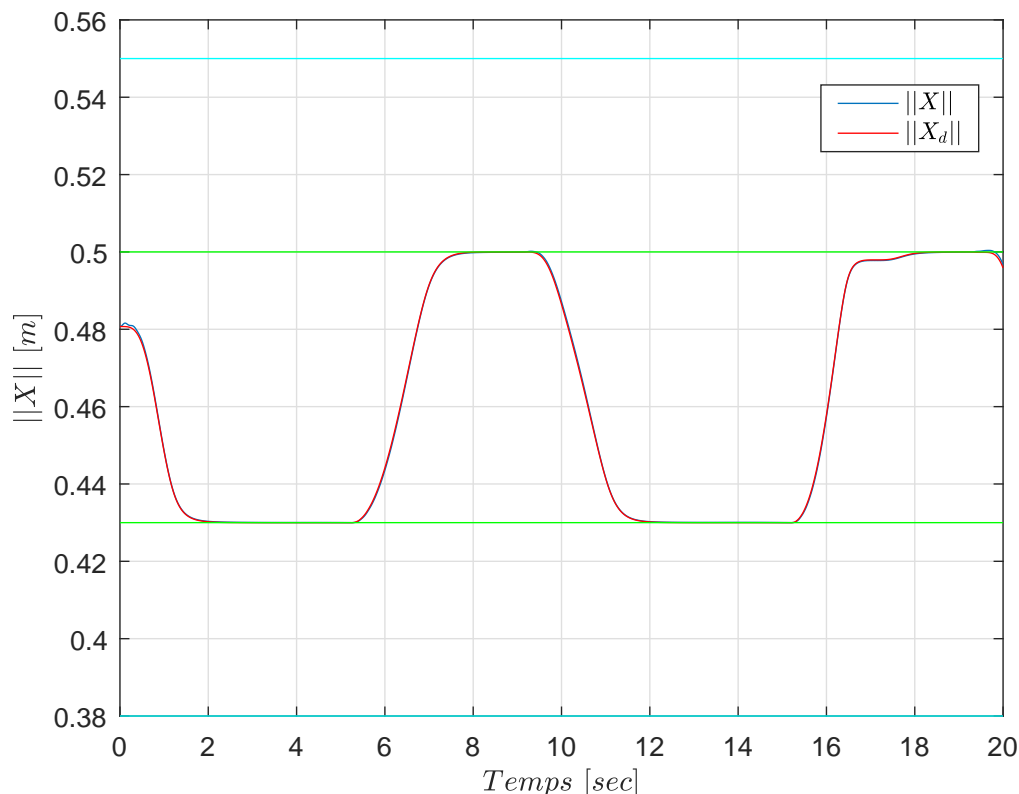


FIGURE 5.7 – Norme de la position de l'organe terminal.

Les rayons  $R_M$  et  $R_m$  de la demi-sphère virtuelle sont représentés par les deux lignes en cyan, tandis que les rayons  $R_M - D_F$  et  $R_m + D_F$  sont représentés par les lignes vertes. La distance entre les deux lignes correspond bien à la distance de sécurité.

La figure 5.7 montre l'efficacité du générateur de trajectoire sous la contrainte de l'espace de travail à limites virtuelles. Notons que la norme de la position est limitée aux bornes de la zone de travail prédéfinie. Cette restriction est due à la pénalisation de la composante normale  $F_N$  de la force  $F_h$  (Figure 5.8). Nous pouvons aussi constater qu'il n'y a pas de dépassement de la zone bornée, grâce à l'adaptation de la matrice des coefficients de raideurs.

La figure 5.8 présente la composante normale de  $F_h$  avant (bleu) et après (rouge) pénalisation.

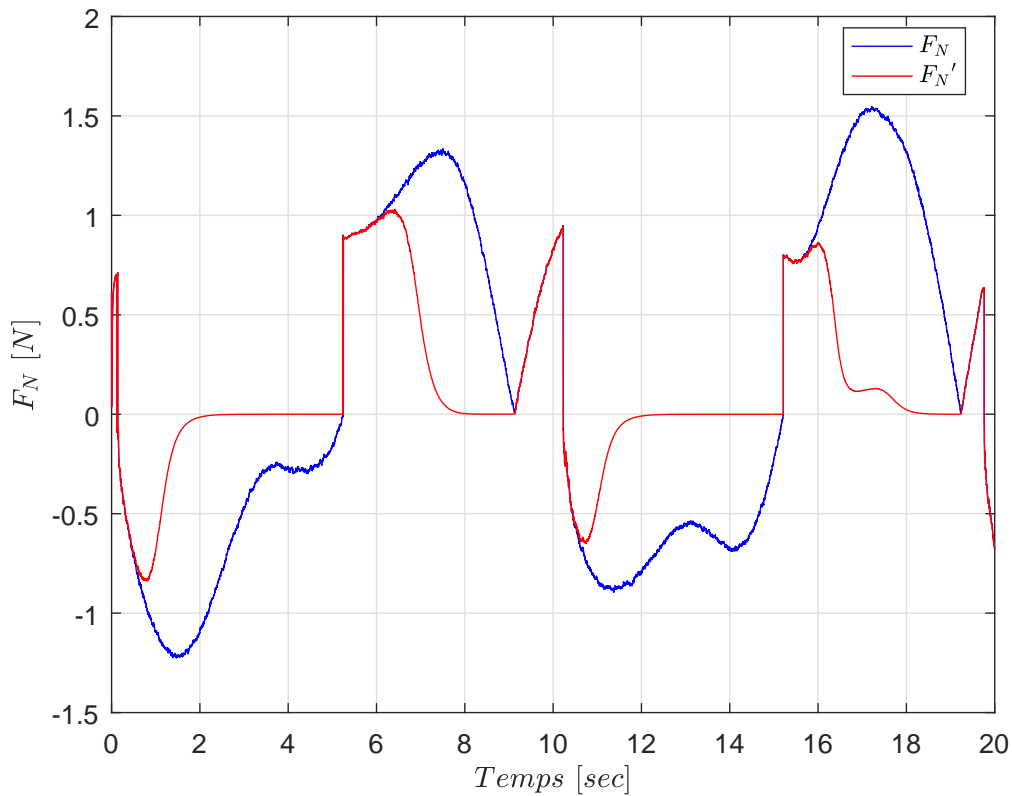
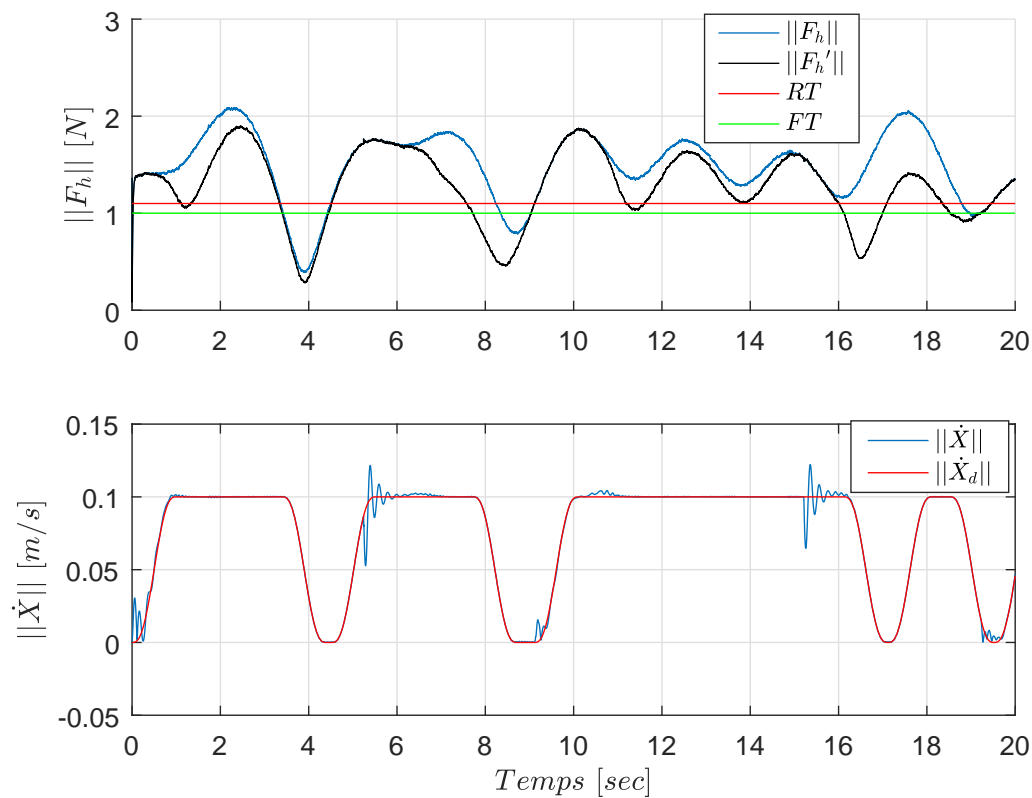


FIGURE 5.8 – Composante normale de la force avant et après pénalisation.

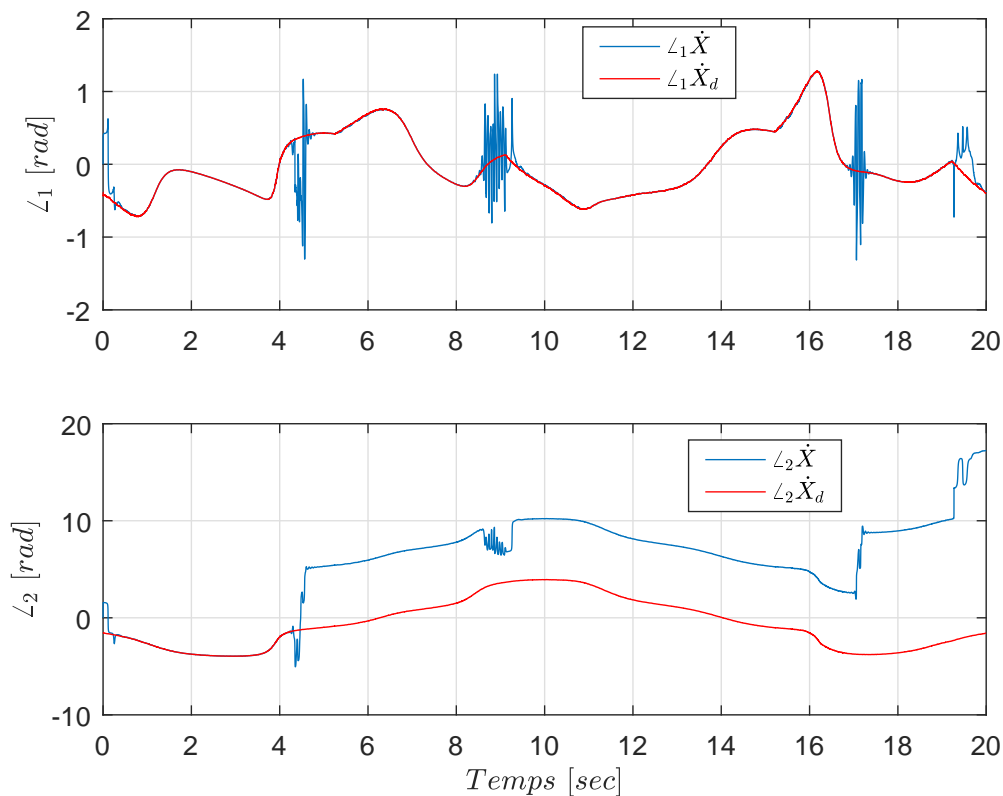
Nous pouvons remarquer que cette composante tend vers zéro progressivement (courbe en rouge) lorsque l'organe terminal se rapproche de la limite de l'espace de travail.

La figure 5.9a en haut représente la norme de la force de l'opérateur avant (bleu) et après (noir) pénalisation. Les seuils d'activation des états *Rise* et *Fall* sont représentés par les lignes rouges et vertes, respectivement. La figure 5.9a en bas représente la norme de la vitesse de l'organe terminal (bleu) avec sa référence (rouge). La figure 5.9b représente les angles du vecteur vitesse linéaire de l'organe terminal et les références correspondantes.

A partir des figures 5.9b et 5.9a, la poursuite de la norme et des angles de la vitesse par rapport à leurs références est considérée comme étant vérifiée. Par conséquent, la condition de transparence entre la vitesse et la force de l'opérateur est vérifiée aussi. Pour quelques intervalles de temps, la vitesse désirée est nulle, ce qui explique qu'il y a des parties sur les graphes des angles (figure 5.9b) où ils ne sont pas confondus avec leurs références. Dans ces parties, il n'y a pas de condition de transparence à vérifier.



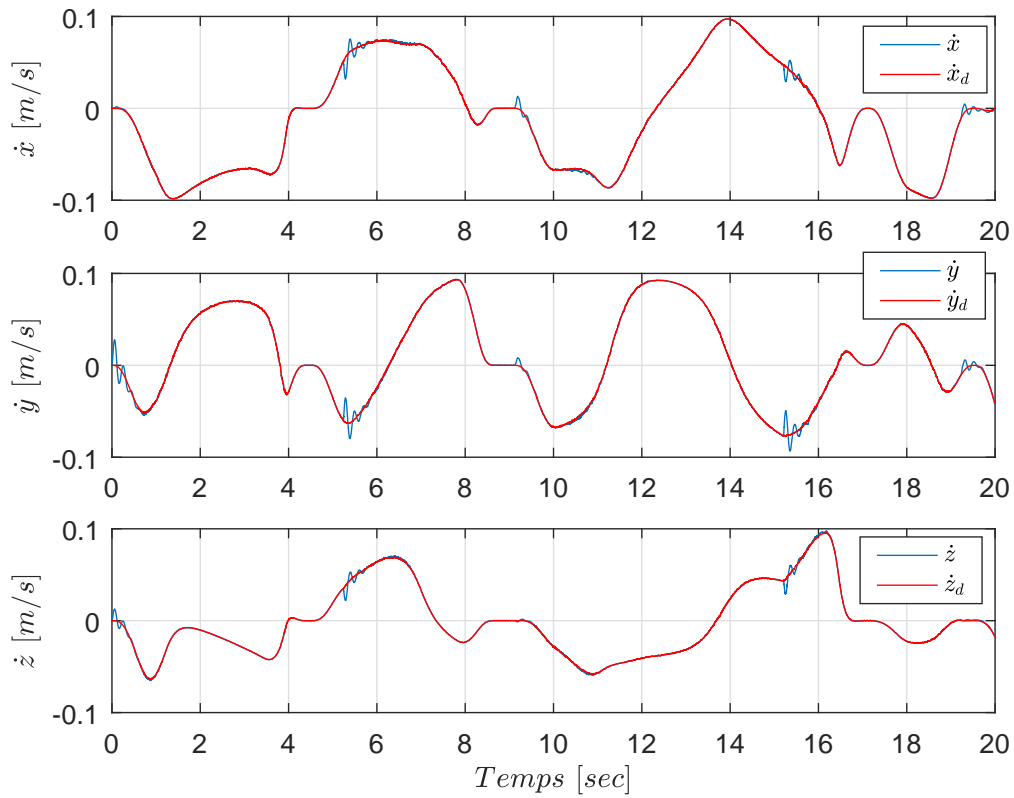
(a) Norme de la force de l'opérateur (haut), et celle de la vitesse de l'organe terminal (bas).



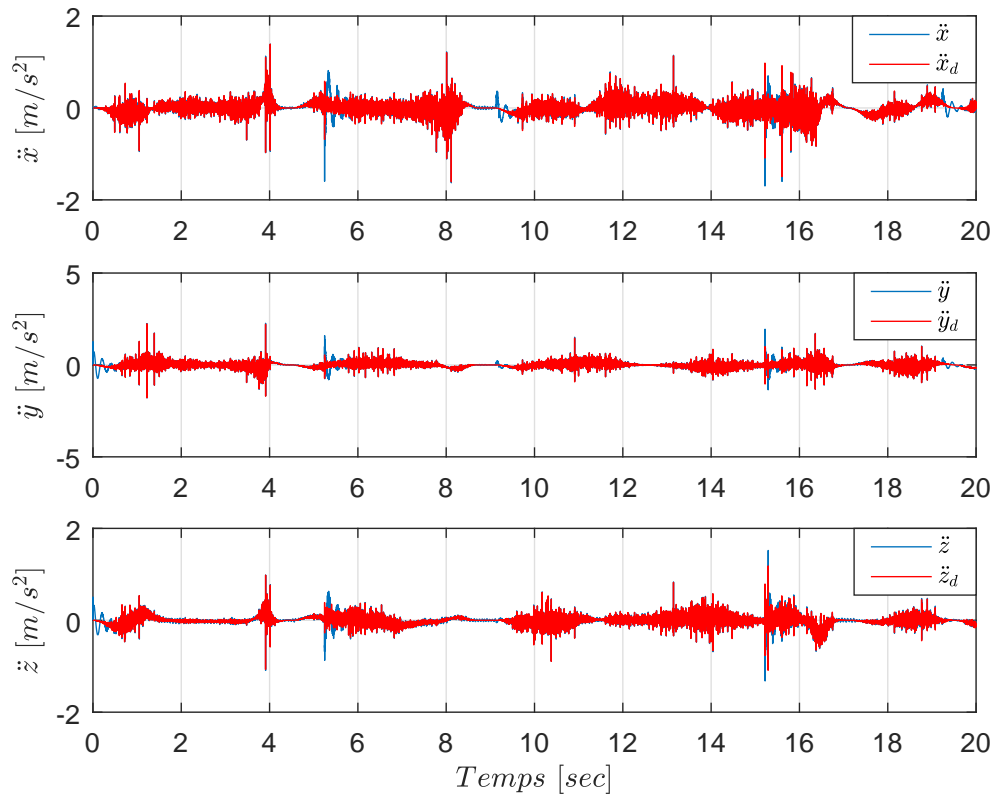
(b) Angles du vecteur vitesse linéaire de l'organe terminal.

FIGURE 5.9 – Caractéristiques du vecteur vitesse linéaire du point d'interaction homme-robot.

La figure 5.10a représente la vitesse de l'organe terminal selon les trois axes du repère  $R_0$ . La figure 5.10b représente l'accélération de l'organe terminal selon les trois axes du repère  $R_0$ . Les figures 5.10a et 5.10b prouvent que les composantes de la vitesse et de l'accélération suivent les trajectoires de référence correspondantes. Les figures 5.11a et 5.11b montrent les positions et vitesses articulaires avec leurs limites définies dans le tableau 3.2. Nous pouvons remarquer que les vitesses articulaires respectent les limites sur les vitesses, ce qui n'est pas le cas pour les positions ( $q_2$ ,  $q_3$ ,  $q_4$ ,  $q_5$ ). Afin d'éviter les collisions du robot avec son environnement et avec ses propres organes, il est indispensable de garantir que les  $q_i$  respectent les limites sur les positions articulaires, avant de passer à l'implémentation sur le robot réel. La figure 5.12 présente les couples articulaires.

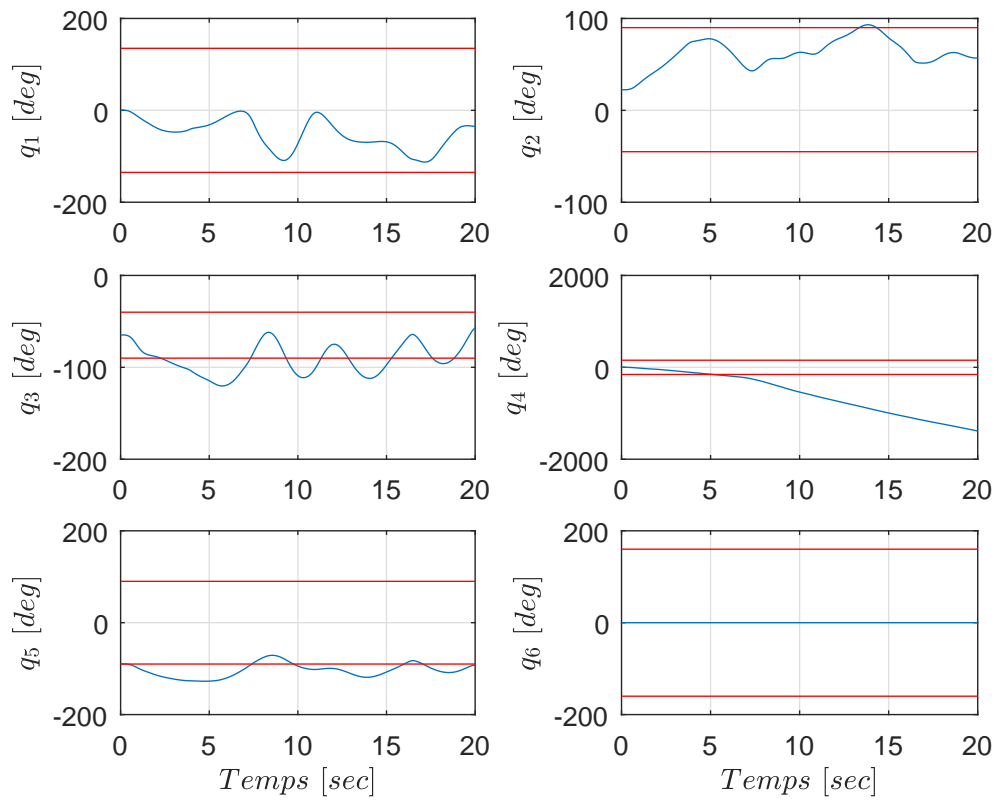


(a) Vitesse de l'organe terminal selon chacun des trois axes.

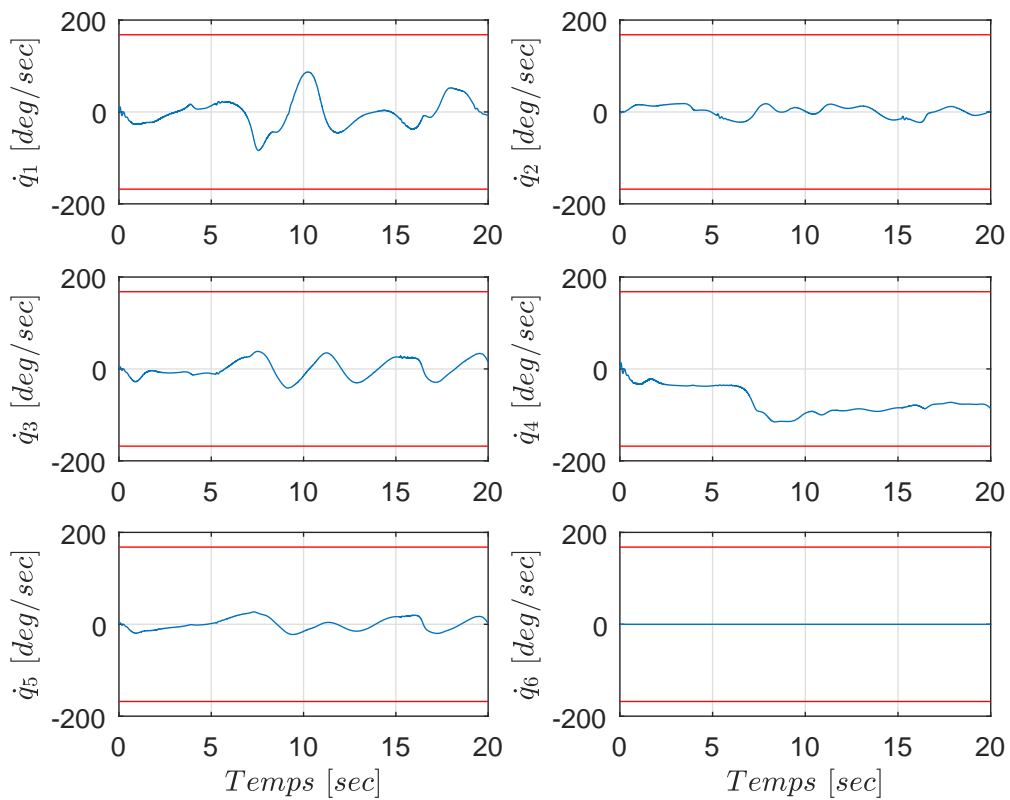


(b) Accélération de l'organe terminal selon chacun des trois axes.

FIGURE 5.10



(a) Positions articulaires.



(b) Vitesses articulaires.

FIGURE 5.11 – Positions et vitesses articulaires, avec leurs limites.

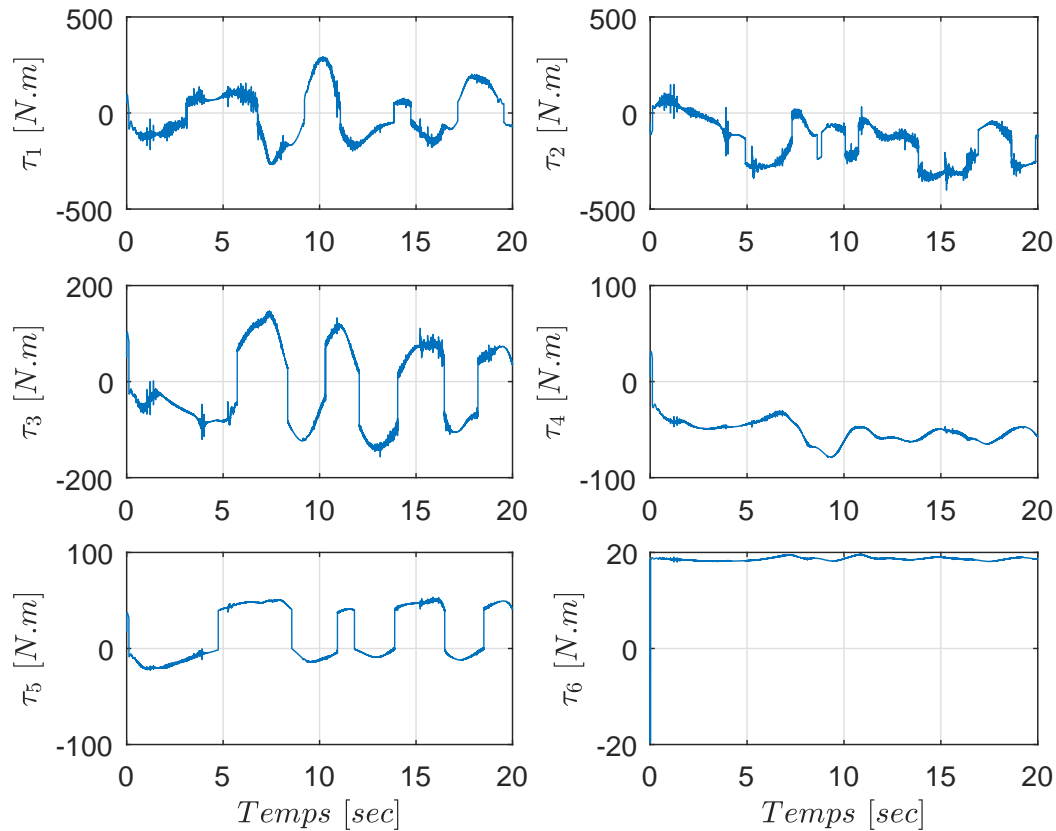


FIGURE 5.12 – Couples articulaires.

## 5.4 Conclusion

Dans ce chapitre, nous avons illustré en simulation la validité du schéma de commande du robot Denso VP-6242G en interaction avec un opérateur humain virtuel. Pour cela, nous avons proposé d'utiliser une loi de commande par inversion du modèle dynamique, basée sur le meilleur modèle identifié pour ce robot. Nous avons cherché à examiner le fonctionnement du générateur de trajectoire en ligne, présenté au chapitre 4 pour vérifier s'il permettait d'assurer une co-manipulation homme-robot conforme aux exigences de la problématique considérée dans ce mémoire, rappelées en Introduction et dans le chapitre 4. Dans l'ensemble, le schéma de commande donne de bons résultats en matière de qualité de la poursuite des trajectoires de référence produites par le générateur de trajectoire. Cependant, ces bons résultats sont tributaires du modèle identifié au chapitre 3 utilisé pour l'analyse. Il est évident que seule une validation expérimentale du schéma de commande proposé sur le robot Denso VP-6242G permettrait de tirer des conclusions valables. Les simulations de la figure 5.11a montre qu'il est risqué d'envisager en l'état une mise en œuvre expérimentale car, bien que l'espace d'évolution autorisé pour une co-manipulation soit limité par des butées virtuelles, le générateur de trajectoires actuel ne permet pas d'éviter des configurations géométriques hors des limites articulaires qui empêchent

---

le robot de passer par des configurations singulières, voire, de franchir les butées articulaires qui servent à empêcher des auto-collisions de certains corps du robot avec lui-même, et avec l'environnement.





# Conclusion et perspectives

## Bilan

Durant cette étude, nous nous sommes intéressés à la commande des robots manipulateurs industriels en mode de co-manipulation avec un opérateur humain pour la manutention de charges lourdes. Dans un premier temps, nous avons présenté l'ensemble des études qui ont été menées en co-manipulation homme-robot, que nous pouvons rencontrer dans plusieurs domaines notamment en industrie et en chirurgie. Ensuite, nous avons abordé la modélisation et l'identification des paramètres dynamiques du robot *Denso VP-6242G*. Nous avons utilisé le logiciel *OpenSYMORO* pour calculer son modèle dynamique. Le logiciel nous a permis de réduire le temps dédié à la partie modélisation, grâce à sa puissance de calcul symbolique. Après une présentation détaillée de la méthode d'identification des paramètres de robots manipulateurs, nous l'avons appliqué au cas de notre robot. Cela nous a permis d'obtenir un vecteur des paramètres qui garantit une matrice d'inertie définie positive pour n'importe quelle configuration du robot, et aussi, une bonne qualité de reconstruction des couples pour des vitesses articulaires constantes, ou variables au cours du temps. Pour poursuivre, nous avons détaillé les nouvelles fonctionnalités proposées pour le générateur de trajectoire en temps réel. Nous avons présenté la méthode d'estimation de la force de l'opérateur à partir des mesures de la force d'interaction entre le robot et l'opérateur, la pénalisation de la force de l'opérateur pour avoir une image de cette dernière qui permet de générer une trajectoire qui respecte les limites de l'espace de travail, et l'automate qui permet de gérer les différents états du robot (marche, arrêt, transition). Des tests sur le générateur pour différents cas de figure possibles nous ont permis de vérifier l'efficacité des nouvelles fonctionnalités proposées. Le générateur permet de générer une trajectoire dans l'espace de travail tridimensionnel selon la direction de l'effort appliqué par l'opérateur, ce qui contribue à l'exigence de transparence requise durant un essai de co-manipulation. Le générateur assure aussi des vitesses désirées sans danger pour l'opérateur. Dans le dernier chapitre, nous avons présenté et validé en simulation une commande en impédance dont les trajectoires de référence sont issues du générateur développé. Les résultats obtenus ont donné lieu à une bonne qualité de poursuite des trajectoires désirées. D'autre part, le respect des limites virtuelles de l'espace de travail a également bien été pris en

compte. Cependant, les trajectoires articulaires correspondantes peuvent franchir les limites définies pour préserver l'intégrité du robot. En effet, un tel dépassement peut conduire à des auto-collisions entre les différents corps du robot, ou entre ces derniers et l'environnement du robot. Il est donc impératif de régler cette problématique des limites articulaires avant de procéder à la mise en pratique de la stratégie étudiée.

## Perspectives

Les travaux présentés dans cette étude ouvrent plusieurs perspectives de développement, notamment :

- **La prise en compte des butées articulaires** : la commande en impédance basée sur le générateur de trajectoire ne doit pas faire sortir le robot de son espace articulaire prédéfini, afin d'éviter le problème des auto-collisions et les collisions avec l'environnement. Il faut aussi assurer que la prise en compte des butées articulaires, n'affecte pas la colinéarité entre la force de l'opérateur et la vitesse désirée. Autrement-dit, elle n'affecte pas la qualité de transparence de la co-manipulation.
- **L'implémentation sur le robot réel** : la mise en pratique de la technique de commande étudiée sur le robot réel Denso VP-6242G devrait suivre différentes étapes : l'implémentation du générateur de trajectoire, la prise en compte du modèle du capteur de force, l'implémentation de la loi de commande et l'étude de la stabilité de l'interaction homme-robot.
- **La co-manipulation en 6D** : actuellement, le schéma de commande permet de déplacer librement le point d'interaction homme-robot durant l'interaction. Il serait utile d'étendre l'étude de manière à permettre d'orienter l'organe terminal suivant le désir de l'opérateur. Pour cela, nous pensons envisageable de prendre en compte les couples mesurés par le capteur, et d'établir un lien entre cette information et l'orientation de l'organe terminal.
- **Le guide virtuel** : le principe de la pénalisation de la force de l'opérateur pourrait être appliqué pour mettre en place un guide virtuel du robot, en mode de co-manipulation. Cela consisterait à imposer au point d'interaction un mouvement transparent le long du guide, sans permettre à l'opérateur de s'en écarter.

# Bibliographie

- [1] <https://www.ati-ia.com/products/ft/sensors.aspx>.
- [2] <https://www.densorobotics-europe.com/fr/product/vp-6242g>.
- [3] <https://www.nibib.nih.gov/news-events/newsroom/adding-feeling-robot-assisted-surgery>.
- [4] <https://www.python.org/>.
- [5] <https://www.quanser.com/products/joint-control-robot-6-dof-d/>.
- [6] <https://www.quanser.com/products/quarc-real-time-control-software/>.
- [7] <https://www.wxpython.org/>.
- [8] <http://www.konecranes.fi/laitteet/kevytnosturiarjestelmat/atb-airbalancer-paineilmakevennin>.
- [9] Edgar R. Powell. - *Powell tool balancer*. - Brevet US3428298A, Feb. 1969.
- [10] Michael Kornely. - *Self balancing electric hoist*. - Brevet US4658971A,, Apr. 1987.
- [11] Iso 10218 :1992 manipulating industrial robots – safety, 1992.
- [12] Homayoon Kazerooni . - *Human power amplifier for vertical maneuvers*. - Brevet US5865426A,, Feb. 1999.
- [13] Iso 10218-1 :2006 robots for industrial environments – safety requirements – part 1 : Robot, 2006.
- [14] Iso 10218-1 :2011 robots and robotic devices – safety requirements for industrial robots – part 1 : Robots, 2011.
- [15] S. Alfayad, F. B. Ouezdou, F. Namoun, and G. Gheng. High performance integrated electro-hydraulic actuator for robotics - part i : Principle, prototype design and first experiments. *Sensors and Actuators A : Physical*, 169(1) :115 – 123, 2011.
- [16] S. Alfayad, F. B. Ouezdou, F. Namoun, and G. Gheng. High performance integrated electro-hydraulic actuator for robotics - part ii : Theoretical modelling, simulation, control & comparison with real measurements. *Sensors and Actuators A : Physical*, 169(1) :124 – 132, 2011.

- [17] C. H. An, C. G. Atkeson, and J. M. Hollerbach. Estimation of inertial parameters of rigid body links of manipulators. In *1985 24th IEEE Conference on Decision and Control*, pages 990–995, Dec. 1985.
- [18] M. H. Ang, W. Lin, and S. Lim. A walk-through programmed robot for welding in shipyards. *Industrial Robot : the international journal of robotics research and application*, 26(5) :377–388, 1999.
- [19] A. Bahloul, S. Tliba, and Y. Chitour. Dynamic parameters identification of an industrial robot : A constrained nonlinear wls approach. In *Proc. of the 26th IEEE Mediterranean Conference on Control and Automation (MED 2018)*, pages 21–26, Zadar, Croatia, June 19-22, 2018. IEEE.
- [20] A. Bahloul, S. Tliba, and Y. Chitour. Dynamic parameters identification of an industrial robot with and without payload. In *Proc. of the 18th IFAC Symposium on System Identification (SYSID 2018)*, pages 443–448, Stockholm, Sweden, July 9-11, 2018. IFAC.
- [21] A. Bicchi, M. Bavaro, G. Boccadamo, D. D. Carli, R. Filippini, G. Grioli, M. Piccigallo, A. Rosi, R. Schiavi, S. Sen, and G. Tonietti. Physical human-robot interaction : Dependability, safety, and performance. In *2008 10th IEEE International Workshop on Advanced Motion Control*, pages 9–14, Mar. 2008.
- [22] A. Bicchi, M. A. Peshkin, and J. E. Colgate. *Safety for Physical Human–Robot Interaction*, pages 1335–1348. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [23] M. Brunot, A. Janot, F. Carrillo, and H. Garnier. Comparison between the idim-iv method and the didim method for industrial robots identification. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 571–576, July 2017.
- [24] M. Brunot, A. Janot, F. Carrillo, and H. Garnier. A pragmatic and systematic statistical analysis for identification of industrial robots. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 559–564, July 2017.
- [25] B. Cagneau, G. Morel, D. Bellot, N. Zemiti, and G. A. d’Agostino. A passive force amplifier. In *2008 IEEE International Conference on Robotics and Automation*, pages 2079–2084, May 2008.
- [26] D. G. Caldwell, N. G. Tsagarakis, S. Kousidou, N. Costa, and I. Sarakoglou. "soft" exoskeletons for upper and lower body rehabilitation - design, control and testing. *International Journal of Humanoid Robotics*, 04(03) :549–573, 2007.
- [27] E. Dombre and W. Khalil. *Robot Manipulators - Modeling Performance Analysis and Control*. ISTE, Jan. 2007.
- [28] V. Duchaine and C. Gosselin. Safe, stable and intuitive control for physical human-robot interaction. In *2009 IEEE International Conference on Robotics and Automation*, pages 3383–3388, May 2009.

- [29] V. Duchaine and C. M. Gosselin. Investigation of human-robot interaction stability using lyapunov theory. In *2008 IEEE International Conference on Robotics and Automation*, pages 2189–2194, May 2008.
- [30] G. E. C. F. Croshaw, P. Hardiman i arm test - hardiman i prototype project. Technical report, General Electric Rep, Dec. 1969.
- [31] F. Ferraguti, C. T. Landi, C. Secchi, C. Fantuzzi, M. Nolli, and M. Pesamosca. Walk-through programming for industrial applications. *Procedia Manufacturing*, 11 :31 – 38, 2017. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.
- [32] T. Fukuda, Y. Fujisawa, K. Kosuge, F. Arai, E. Muro, H. Hoshino, K. Miyazaki, K. Ohtsubo, and K. Uehara. Manipulator for man-robot cooperation. In *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on*, volume 2, pages 996–1001, Oct. 1991.
- [33] M. Gautier. Numerical calculation of the base inertial parameters of robots. In *Proceedings., IEEE International Conference on Robotics and Automation*, volume 2, pages 1020–1025, May 1990.
- [34] M. Gautier and W. Khalil. A direct determination of minimum inertial parameters of robots. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, volume 3, pages 1682–1687, Apr. 1988.
- [35] M. Gautier and W. Khalil. On the identification of the inertial parameters of robots. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 3, pages 2264–2269, Dec. 1988.
- [36] M. Gautier and W. Khalil. Identification of the minimum inertial parameters of robots. In *Proceedings, 1989 International Conference on Robotics and Automation*, volume 3, pages 1529–1534, May 1989.
- [37] M. Gautier and W. Khalil. Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on Robotics and Automation*, 6(3) :368–373, June 1990.
- [38] M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. In *Proceedings of the 30th IEEE Conference on Decision and Control*, volume 1, pages 494–499, Dec. 1991.
- [39] Y. Iwamura, M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. Do elderly people prefer a conversational humanoid as a shopping assistant partner in supermarkets? In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 449–457, Mar. 2011.
- [40] M. Jakopc, S. J. Harris, F. R. y Baena, P. Gomes, and B. L. Davies. The acrobot® system for total knee replacement. *Industrial Robot : the international journal of robotics research and application*, 30(1) :61–66, 2003.
- [41] A. Janot, P. O. Vandanjon, and M. Gautier. A generic instrumental variable approach for industrial robot identification. *IEEE Transactions on Control Systems Technology*, 22(1) :132–145, Jan. 2014.

- [42] S. Jlassi. *Formulation et études des problèmes de commande en co-manipulation robotique*. Theses, Université Paris Sud - Paris XI, Nov 2013.
- [43] S. Jlassi, S. Tliba, and Y. Chitour. On human-robot co-manipulation for handling tasks : Modeling and control strategy. *IFAC Proceedings Volumes*, 45(22) :710 – 715, 2012. 10th IFAC Symposium on Robot Control.
- [44] S. Jlassi, S. Tliba, and Y. Chitour. An event-controlled online trajectory generator based on the human-robot interaction force processing. *Industrial Robot : the international journal of robotics research and application*, 41(1) :15–25, 2014.
- [45] S. Jlassi, S. Tliba, and Y. Chitour. An online trajectory generator-based impedance control for co-manipulation tasks. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 391–396, Feb. 2014.
- [46] H. Kazerooni. Extender : a case study for human-robot interaction via transfer of power and information signals. In *Proceedings of 1993 2nd IEEE International Workshop on Robot and Human Communication*, pages 10–20, Nov. 1993.
- [47] W. Khalil and D. Creusot. Symoro+ : A system for the symbolic modelling of robots. *Robotica*, 15(2) :153–161, Mar. 1997.
- [48] W. Khalil and J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179, Apr 1986.
- [49] W. Khalil and J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179, Apr. 1986.
- [50] W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, and G. Ecorchard. Opensymoro : An open-source software package for symbolic modelling of robots. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1206–1211, July 2014.
- [51] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Rob. Res.*, 5(1) :90–98, Apr. 1986.
- [52] M. Kimmel and S. Hirche. Active safety control for dynamic human-robot interaction. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4685–4691, Sept. 2015.
- [53] K. Kosuge, Y. Fujisawa, and T. Fukuda. Control of robot directly maneuvered by operator. In *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 1, pages 49–54, July 1993.
- [54] A. Kouris, F. Dimeas, and N. Aspragathos. Contact distinction in human-robot cooperation with admittance control. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1951–1956, Oct. 2016.
- [55] J. Kruger, T. Lien, and A. Verl. Cooperation of human and machines in assembly lines. *CIRP Annals*, 58(2) :628 – 646, 2009.
- [56] R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. S. Jensen, L. L. Whitcomb, and R. H. Taylor. Preliminary experiments in cooperative human/robot force

- control for robot assisted microsurgical manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, pages 610–617, Apr. 2000.
- [57] X. Lamy, F. Collédani, F. Geffard, Y. Measson, and G. Morel. Human force amplification with industrial robot : Study of dynamic limitations. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2487–2494, Oct. 2010.
- [58] X. Lamy, F. Collédani, F. Geffard, Y. Measson, and G. Morel. Overcoming human force amplification limitations in comanipulation tasks with industrial robot. In *2010 8th World Congress on Intelligent Control and Automation*, pages 592–598, July 2010.
- [59] K. Lee, S. Lee, J. Choi, S. Lee, and C. Han. The application of the human-robot cooperative system for construction robot manipulating and installing heavy materials. In *2006 SICE-ICASE International Joint Conference*, pages 4798–4802, Oct. 2006.
- [60] H.-O. Lim and K. Tanie. Collision-tolerant control of human-friendly robot with viscoelastic trunk. *IEEE/ASME Transactions on Mechatronics*, 4(4) :417–427, Dec. 1999.
- [61] L. Ljung. *System Identification : Theory for User*. Prentice Hall, 2 edition, 1999.
- [62] S. Lu and J. H. Chung. Collision detection enabled weighted path planning : a wrist and base force/torque sensors approach. In *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 165–170, July 2005.
- [63] M. V. Mark W. Spong, Seth Hutchinson. *Robot Modeling and Control*. Wiley, 2005.
- [64] H. Mehdi and O. Boubaker. Rehabilitation of a human arm supported by a robotic manipulator : A position/force cooperative control. *Journal of Computer Science*, pages 912–919, 2010.
- [65] B. Mitchell, J. Koo, I. Iordachita, P. Kazanzides, A. Kapoor, J. Handa, G. Hager, and R. Taylor. Development and application of a new steady-hand manipulator for retinal surgery. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 623–629, Apr. 2007.
- [66] M. H. A. J. Ngoc Dung Vuong. Dynamic model identification for industrial robots. *Acta polytechnica Hungarica*, 6(5) :51–68, 2009.
- [67] M. Ragaglia, L. Bascetta, and P. Rocco. Detecting, tracking and predicting human motion inside an industrial robotic cell using a map-based particle filtering strategy. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 369–374, July 2015.
- [68] G. Raiola. *Co-manipulation with a library of Virtual Guides*. PhD thesis, Université Paris-Saclay ENSTA-Paris-Tech, 2017.



- [69] S. S. Restrepo, G. Raiola, P. Chevalier, X. Lamy, and D. Sidobre. Iterative virtual guides programming for human-robot comanipulation. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 219–226, July 2017.
- [70] J. G. Romero, R. Ortega, and I. Sarras. A globally exponentially stable tracking controller for mechanical systems using position feedback. *IEEE Transactions on Automatic Control*, 60(3) :818–823, Mar. 2015.
- [71] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, 43(3) :253–270, 2008.
- [72] M. Shariatee, H. Khosravi, and E. Fazl-Ersi. Safe collaboration of humans and scara robots. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 589–594, Oct. 2016.
- [73] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer Publishing Company, Incorporated, 2nd edition, 2016.
- [74] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics Modelling, Planning and Control*. Springer-Verlag London, 1st edition, 2009.
- [75] J. Swevers, W. Verdonck, and J. D. Schutter. Dynamic model identification for industrial robots. *IEEE Control Systems*, 27(5) :58–71, Oct. 2007.
- [76] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. Dejuan, and L. Kavoussi. A steady-hand robotic system for microsurgical augmentation. *The International Journal of Robotics Research*, 18 :1201–1210, Dec. 1999.
- [77] F. Vicentini, M. Giussani, and L. M. Tosatti. Trajectory-dependent safe distances in human-robot interaction. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–4, Sept. 2014.
- [78] V. Villani, F. Pini, F. Leali, and C. Secchi. Survey on human-robot collaboration in industrial settings : Safety, intuitive interfaces and applications. *Mechatronics*, 2018.
- [79] B. Whitsell and P. Artemiadis. Physical human-robot interaction (phri) in 6 dof with asymmetric cooperation. *IEEE Access*, 5 :10834–10845, 2017.
- [80] Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani, and K. Suita. Human-robot contact in the safeguarding space. *IEEE/ASME Transactions on Mechatronics*, 2(4) :230–236, Dec. 1997.
- [81] K. Yoshida and W. Khalil. Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters. *The International Journal of Robotics Research*, 19(5) :498–510, May 2000.
- [82] Y. Zhan, X. g. Duan, and J. x. Li. Review of comanipulation robot in surgery. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1466–1471, Aug. 2015.



**Titre :** Sur la commande de robots manipulateurs industriels en configuration de co-manipulation robotique.  
**Mots-clés :** Interaction Physique Homme-Robot, Co-manipulation robotique, Robot manipulateur industriel, Identification, Commande en impédance.

**Résumé :** Durant ce travail de thèse, nous nous sommes intéressés à la commande d'un robot manipulateur industriel, configuré pour une co-manipulation avec un opérateur humain, en vue de la manutention de charges lourdes. Dans un premier temps, nous avons présenté une vue d'ensemble des études qui ont été menées dans ce cadre. Ensuite, nous avons abordé la modélisation et l'identification des paramètres dynamiques du robot Denso VP-6242G. Nous avons utilisé le logiciel OpenSYMORO pour calculer son modèle dynamique. Après une présentation détaillée de la méthode d'identification des paramètres de robots manipulateurs, nous l'avons appliqué au cas de notre robot. Cela nous a permis d'obtenir un vecteur des paramètres qui garantit une matrice d'inertie définie positive pour n'importe quelle configuration articulaire du robot, tout en assurant une bonne qualité de reconstruction des couples pour des vitesses articulaires constantes, ou variables au cours du temps. Par la suite, nous avons détaillé les nouvelles fonctionnalités proposées pour le générateur de trajectoire en temps réel, sur lequel repose notre schéma de commande. Nous avons présenté une méthode d'estimation de la

force de l'opérateur à partir des mesures de la force d'interaction entre le robot et l'opérateur, tout en tenant compte de la pénalisation de la force de l'opérateur afin d'avoir une image de cette dernière permettant de générer une trajectoire qui respecte les limites de l'espace de travail. Des tests du générateur de trajectoire simulant différents cas de figure possibles nous ont permis de vérifier l'efficacité des nouvelles fonctionnalités proposées. Le générateur permet de produire une trajectoire dans l'espace de travail tridimensionnel selon la direction de l'effort appliqué par l'opérateur, ce qui contribue à l'exigence de transparence recherchée en co-manipulation robotique. Dans la dernière partie, nous avons présenté et validé en simulation une commande en impédance dont les trajectoires de référence sont issues du générateur développé. Les résultats obtenus ont donné lieu à une bonne qualité de poursuite des trajectoires désirées. D'autre part, le respect des limites virtuelles de l'espace de travail a également été pris en compte. Cependant, les trajectoires articulaires correspondantes peuvent franchir les limites définies pour préserver l'intégrité du robot.

**Title :** On the control of industrial robots for robotic comanipulation tasks

**Keywords :** Physical Human-Robot Interaction, Robotic comanipulation, Industrial robot, Identification, Impedance Control.

**Abstract :** In this thesis, we were interested in the control of industrial manipulators in co-manipulation mode with a human operator for the handling of heavy loads. First, we have presented an overview of existing studies in this framework. Then, we have addressed the modeling and the identification of dynamic parameters for the Denso VP-6242G robot. We have used the OpenSYMORO software to calculate its dynamical model. After a detailed presentation of the method for identifying the robot's parameters, we have applied it to the case of our robot. This allowed us to obtain a vector of the parameters which guarantees a positive definite inertia matrix for any configuration of the robot, as well as a good quality of reconstruction of the torques in the case of constant joint velocities or in the case of variable ones over time. To continue, we have detailed the new features that have been proposed for the on-line trajectory generator, for which the control scheme is based on. We have presented a method for estimating the operator's force from the measurements of the interaction force between the robot and the opera-

tor, while taking into account for the penalization of the operator's force in order to have an information of this last which allows to generate a trajectory that respects the limits of the workspace. Some tests of the trajectory generator simulating different possible scenarios have allowed us to check the effectiveness of the new proposed features. The generator makes it possible to produce a trajectory in the three-dimensional workspace according to the direction of the force applied by the operator, which contributes to fulfill the requirement of transparency that is sought in a co-manipulation. In the last part, we have presented and validated, in simulation, an impedance control whose reference trajectories are delivered by the proposed generator. The obtained results have shown a good trajectory tracking. On the other hand, the satisfaction of the virtual bounds of the workspace has also been nicely taken into account. However, the corresponding articular trajectories can cross the bounds defined to preserve the integrity of the robot.

