

الجمهورية الجزائرية الديمقراطية الشعبية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET  
POPULAIRE

وزارة التعليم العالي و البحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

ÉCOLE NATIONALE POLYTECHNIQUE

LABORATOIRE DE COMMANDE DES PROCESSUS

DÉPARTEMENT D'AUTOMATIQUE



Mémoire de projet de fin d'études pour l'obtention du diplôme d'Ingénieur d'État  
en Automatique.

## Modélisation et Commande d'Ordre Fractionnaire du Robot Parallèle Delta ISIR 88. Simulation et Expérimentation

Réalisé par :

**BEN RABAH Mohamed**

**BENALI Abderahmane**

Sous la direction de :

**Pr. BOUKHETALA Djamel**

**Dr. BOUDJEDIR Chems Eddine**

Présenté et soutenu publiquement le 07/07/2021

Composition du Jury :

Président	Mr. BOUCHERIT Mohamed Seghir	Professeur	École Nationale Polytechnique
Promoteurs	Mr. BOUKHETALA Djamel	Professeur	École Nationale Polytechnique
	Mr. BOUDJEDIR Chems Eddine	Docteur	Université de Science et Technologie Houari Boumediene
Examineur	Mr. BOUDANA Djamel	Professeur	École Nationale Polytechnique

ENP 2021



الجمهورية الجزائرية الديمقراطية الشعبية

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET  
POPULAIRE

وزارة التعليم العالي و البحث العلمي

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

ÉCOLE NATIONALE POLYTECHNIQUE

LABORATOIRE DE COMMANDE DES PROCESSUS

DÉPARTEMENT D'AUTOMATIQUE



Mémoire de projet de fin d'études pour l'obtention du diplôme d'Ingénieur d'État  
en Automatique.

## Modélisation et Commande d'Ordre Fractionnaire du Robot Parallèle Delta ISIR 88. Simulation et Expérimentation

Réalisé par :

**BEN RABAH Mohamed**

**BENALI Abderahmane**

Sous la direction de :

**Pr. BOUKHETALA Djamel**

**Dr. BOUDJEDIR Chems Eddine**

Présenté et soutenu publiquement le 07/07/2021

Composition du Jury :

Président	Mr. BOUCHERIT Mohamed Seghir	Professeur	École Nationale Polytechnique
Promoteurs	Mr. BOUKHETALA Djamel	Professeur	École Nationale Polytechnique
	Mr. BOUDJEDIR Chems Eddine	Docteur	Université de Science et Technologie Houari Boumediene
Examineur	Mr. BOUDANA Djamel	Professeur	École Nationale Polytechnique

ENP 2021

## ملخص

إن الهدف الذي ترمي إليه هذه المذكرة هو النمذجة و التحكم في الآلة المتوازية دلتا إزير ٨٨. فهي تعتبر آلة مشهورة اجتاحت في الآونة الأخيرة الساحة الصناعية نظرا لأدائها السريع و الدقيق. تبدأ الدراسة بنبذة نصف فيها بشكل عام الآلات و الآلات المتوازية بشكل مستفيض. ثم تطرقنا بعدها إلى النمذجة الهندسية، الحركية و الديناميكية. قارنا بعد ذلك مختلف آليات التحكم (PD، PD ذو درجة كسرية و TDC). تعتبر هذه القوانين متينة بحيث تتيح التحكم في الروبوت على الرغم من وجود أخطاء في النمذجة. تتضمن الدراسة كلا من المحاكاة و التجربة.

كلمات مفتاحية: الآلة دلتا، آلات المتوازية، نمذجة و تحكم، التحكم بالإعتماد على المعلومات السابقة TDC، التحكم بمتحكمات ذات درجة كسرية.

---

## Abstract

The aim of this work is the modeling and control of the parallel robot Delta ISIR88. This robot has gained a reputation in pick and place operations in the industry because of its fast and accurate performance. The study begins with an introduction about the different types of robots. Afterwards, a global modeling for the Delta robot ISIR88 is presented. Several control structures are considered and compared (PD, fractional order PD and TDC). These control laws are robusts that they enable us to control the robot despite modeling uncertainties. The study involves both simulation and experimentation.

**Keywords:** Delta robot, parallel robots, modeling and control, Time Delay Control "TDC", fractional order controllers.

---

## Résumé

Ce mémoire de projet de fin d'études porte sur la modélisation et la commande du robot Delta ISIR88; utilisé dans plusieurs domaines d'industrie grâce à sa rapidité et sa précision. L'étude débute par une introduction générale sur les différents types de robots avec focalisation sur les robots parallèles. Par la suite, une modélisation globale du robot Delta ISIR88 a été faite. Plusieurs commandes ont été synthétisées et comparées (PD, PD d'ordre fractionnaire et TDC). Ces lois de commandes sont robustes du fait qu'elles nous permettent de commander le robot en présence des incertitudes de modélisation. L'étude comporte la simulation et l'expérimentation de ces lois de commande.

**Mots clés :** robot Delta, robots parallèles, modélisation et commande, commande à retard TDC, commande d'ordre fractionnaire.

---

---

# Dédicaces

---

À nos chers parents ;

À nos frères et soeurs ;

À nos familles ;

À nos amis qui nous ont soutenu.

---

# Remerciements

---

En préambule à ce projet de fin d'études, nous souhaiterions adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué de près ou de loin à l'élaboration de ce travail.

Nous tenons à remercier spécialement nos promoteurs Pr Boukhetala et Dr Boudjedir pour nous avoir encadré, pour les précieux conseils qu'ils nous ont donné, pour leur aide et pour leur présence tout au long de ce projet.

Nous remercions également tout nos enseignants de l'école nationale polytechnique qui nous ont tant donné tout au long de notre cursus. Nos profonds remerciements à Mr Chekireb.

Enfin, nous souhaiterions remercier nos familles, plus particulièrement nos parents qui nous ont apporté un soutien sans failles et sans lesquels nous ne serions jamais arrivés à ce stade.

# Table des matières

**Table des figures**

**Liste des tableaux**

**Liste des abréviations**

**Liste des symboles**

<b>Introduction générale</b>	<b>14</b>
<b>1 État de l'art</b>	<b>16</b>
1.1 Introduction . . . . .	16
1.2 Définitions de base . . . . .	16
1.2.1 Les articulations . . . . .	16
1.2.2 L'espace cartésien ou opérationnel . . . . .	17
1.2.3 L'espace articulaire . . . . .	17
1.2.4 Configuration singulière . . . . .	18
1.3 Types des robots manipulateurs . . . . .	18
1.3.1 Robots sériels . . . . .	18
1.3.2 Robots parallèles . . . . .	19
1.3.3 Récapitulatif . . . . .	19
1.4 Historique . . . . .	20
1.5 Le robot Delta . . . . .	22
1.5.1 Définition et Description . . . . .	22
1.5.2 Applications : . . . . .	23
1.5.3 Le robot DELTA ISIR88 . . . . .	24
1.6 Conclusion . . . . .	27
<b>2 Modélisation du robot Delta</b>	<b>28</b>
2.1 Introduction . . . . .	28
2.2 Modélisation géométrique . . . . .	28

2.2.1	Modèle géométrique direct . . . . .	28
2.2.2	Modèle géométrique inverse . . . . .	30
2.3	Espace de travail . . . . .	31
2.4	Modélisation cinématique . . . . .	32
2.4.1	Modèle cinématique direct . . . . .	32
2.4.2	Modèle cinématique inverse . . . . .	33
2.5	Singularités . . . . .	34
2.6	Modèle des accélérations . . . . .	34
2.7	Trajectoire . . . . .	35
2.7.1	Lois de mouvements . . . . .	35
2.7.2	Génération de trajectoires . . . . .	37
2.8	Modélisation dynamique . . . . .	39
2.8.1	Les paramètres dynamiques . . . . .	39
2.8.2	Modèle dynamique basé sur le principe de travail virtuel . . . . .	40
2.9	Conclusion . . . . .	42
<b>3</b>	<b>Commande PD d'ordre fractionnaire</b>	<b>43</b>
3.1	introduction . . . . .	43
3.2	Opérateurs d'ordre fractionnaire . . . . .	43
3.2.1	Définition mathématique . . . . .	43
3.2.2	Propriétés des opérateurs d'ordre fractionnaire . . . . .	45
3.2.3	Transformée de Laplace des opérateurs fractionnaires . . . . .	46
3.2.4	Approximation des opérateurs d'ordre fractionnaire . . . . .	47
3.3	Régulateur PID . . . . .	48
3.4	Régulateur PD d'ordre fractionnaire . . . . .	49
3.4.1	Ajustement des paramètres du contrôleur . . . . .	52
3.5	Commande PD d'ordre fractionnaire par la méthode de couple calculé . . . . .	54
3.6	Simulation . . . . .	55
3.6.1	Test de poursuite . . . . .	56
3.6.2	Test de robustesse . . . . .	57
3.7	Expérimentation . . . . .	58
3.7.1	Test avec une dynamique faible . . . . .	60
3.7.2	Test avec une dynamique plus élevée . . . . .	62
3.7.3	Test de robustesse . . . . .	64
3.7.4	Comparaison et discussion . . . . .	66
3.8	Conclusion . . . . .	67
<b>4</b>	<b>La commande "Time Delay Control"</b>	<b>68</b>
4.1	Introduction . . . . .	68
4.2	"Time Delay Control" (TDC) . . . . .	68
4.3	La commande TDC avec mode glissant . . . . .	70
4.3.1	Le mode glissant et le phénomène de Chattering . . . . .	70
4.3.2	Conception de la commande et analyse de convergence . . . . .	71
4.4	La commande TDC avec mode glissant d'ordre fractionnaire . . . . .	73
4.4.1	Conception de la commande et analyse de convergence . . . . .	73
4.5	Simulation . . . . .	75

4.5.1	Test de poursuite . . . . .	75
4.5.2	Test de robustesse . . . . .	76
4.6	Expérimentation . . . . .	77
4.6.1	Test avec une dynamique faible . . . . .	78
4.6.2	Test avec une dynamique plus élevée . . . . .	80
4.6.3	Test de robustesse . . . . .	82
4.6.4	Comparaison et discussion . . . . .	84
4.7	Conclusion . . . . .	84
	<b>Conclusion générale</b>	<b>85</b>
	<b>Bibliographie</b>	<b>87</b>

# Table des figures

1.1	Représentation d'un point . . . . .	17
1.2	Architecture des robots sériels . . . . .	18
1.3	Robot SCARA Fanuc SR-3iA . . . . .	18
1.4	Architecture parallèle . . . . .	19
1.5	Robot parallèle Fanuc . . . . .	19
1.6	Un des premiers robots parallèles, inventé en 1928 par Gwinnett . . . . .	21
1.7	Robot parallèle de Pollard 1934 . . . . .	21
1.8	Plateforme de Gough 1947 . . . . .	21
1.9	Les applications de la plateforme de Stewart par Hexcel Corporation (a) série R (b) série P . . . . .	21
1.10	Schéma descriptif de robot DELTA à 4ddl . . . . .	23
1.11	Application de robot DELTA . . . . .	24
1.12	Schéma mécanique du robot Delta ISIR88 . . . . .	25
1.13	Photo réelle de moteur UGTMEM-03 LB2 . . . . .	26
2.1	Modèle géométrique direct de Delta . . . . .	28
2.2	L'intersection des sphères . . . . .	29
2.3	Vues de face et de profile d'une chaîne « i » du robot Delta . . . . .	29
2.4	Modèle géométrique inverse de Delta . . . . .	31
2.5	Espace de travail du Delta . . . . .	31
2.6	Singularités de Delta . . . . .	34
2.7	Loi de mouvement Bang-bang . . . . .	35
2.8	Loi de mouvement Trapézoïdale . . . . .	36
2.9	Projection de la coordonnée curviligne $s(t)$ donnant $x(t), y(t), z(t)$ . . . . .	37
2.10	Génération de mouvement dans l'espace opérationnel . . . . .	37
2.11	Trajectoire ellipsoïde . . . . .	38
2.12	Trajectoire en segments de droites . . . . .	39
2.13	Force gravitationnelle agissant sur la partie supérieure du bras d'un robot Delta . . . . .	41
3.1	Structure de commande PD sur le robot ISIR88 . . . . .	49
3.2	PID d'ordre fractionnel vs PID classique : de points à plan : (a) ordre entier et (b) ordre fractionnaire . . . . .	50
3.3	Réponse fréquentielle d'un régulateur PID classique avec $k_p = k_i = k_d = 1$ . . . . .	50

3.4	Réponse fréquentielle d'un contrôleur PID classique avec $k_p = k_d = 1, k_i = 3$ . . . . .	50
3.5	Réponse fréquentielle d'un contrôleur PID fractionnaire avec $k_p = k_d = 1, k_i = 1, \mu = \lambda = 0.5$ . . . . .	51
3.6	Structure d'un régulateur PD d'ordre fractionnaire . . . . .	52
3.7	Principe de fonctionnement de la toolbox "Simulink Design Optimisation" . . . . .	53
3.8	Structure du commande de couple calculée avec PD d'ordre fractionnaire . . . . .	55
3.9	Résultats obtenus par la simulation des trois commandes (suivie de trajectoire) . . . . .	56
3.10	Les erreurs de position opérationnelle obtenus par les trois régulateurs . . . . .	56
3.11	Les couples obtenus par la simulation des trois régulateurs . . . . .	56
3.12	Résultats obtenus par la simulation des trois commandes (suivie de trajectoire) avec une masse supplémentaire de 200g . . . . .	57
3.13	Les erreurs de position opérationnelle obtenus par les trois régulateurs avec une masse de 200g . . . . .	57
3.14	Les couples obtenus par la simulation des trois régulateurs avec une masse de 200g . . . . .	57
3.15	Photographie du banc d'expérimentation avec Robot Delta ISIR88 . . . . .	59
3.16	trajectoire désirée . . . . .	59
3.17	Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot . . . . .	60
3.18	Les couples obtenus par l'implémentation des trois régulateurs sur le robot . . . . .	61
3.19	Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot pour une dynamique rapide . . . . .	62
3.20	Les couples obtenus par l'implémentation des trois régulateurs sur le robot pour une dynamique rapide . . . . .	63
3.21	Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot avec une masse de 46g . . . . .	64
3.22	Les couples obtenus par l'implémentation des trois régulateurs sur le robot avec une masse de 46g . . . . .	65
4.1	Le régulateur TDC . . . . .	70
4.2	Fonction "saturation" . . . . .	71
4.3	Le régulateur TDC avec mode glissant . . . . .	73
4.4	Le régulateur TDC avec mode glissant fractionnaire . . . . .	74
4.5	Résultats obtenus par la simulation des trois commandes TDC (suivie de trajectoire) . . . . .	75
4.6	Les erreurs de position opérationnelle obtenus par les trois commandes TDC . . . . .	76
4.7	Les couples obtenus par la simulation des trois commandes TDC . . . . .	76
4.8	Résultats obtenus par la simulation des trois commandes TDC (suivie de trajectoire) avec une masse de 200g . . . . .	76
4.9	Les erreurs de position articulaire obtenus par les trois commandes TDC avec une masse de 200g . . . . .	77
4.10	Les couples obtenus par la simulation des trois commandes TDC avec une masse de 200g . . . . .	77
4.11	Erreurs de position opérationnelle des trois commandes TDC pour une dynamique faible . . . . .	78
4.12	Couples appliqués par les trois commandes TDC pour une dynamique faible . . . . .	79

4.13 Erreurs de position opérationnelle des trois commandes TDC pour une dynamique plus élevée . . . . .	80
4.14 Couples appliqués par les trois commandes TDC pour une dynamique plus élevée	81
4.15 Erreurs de position opérationnelle des trois commandes TDC avec une masse de 46g	82
4.16 Couples appliqués par les trois commandes TDC avec une masse de 46g . . . . .	83

# Liste des tableaux

1.1	Tableau comparatif entre robot série et robot parallèle . . . . .	20
1.2	Caractéristiques mécaniques de robot ISIR88 . . . . .	26
1.3	Caractéristiques des actionneurs de robot Delta ISIR88 . . . . .	27

# Liste des abréviations

CRONE	Commande Robuste d'Ordre Non Entier
CTC	Computed Torque Control
ENP	École Nationale Polytechnique
EPFL	École Polytechnique Fédérale de Lausanne
FOPD	Proportionnel Dérivée d'Ordre Fractionnaire
LCP	Laboratoire de Commande des Processus
MAD	Modèle des Accélérations Direct
MAI	Modèle des Accélérations Inverse
MaxAE	Maximum Absolute Error
MCD	Modèle Cinématique Direct
MCI	Modèle Cinématique Inverse
MDD	Modèle Dynamique Directe
MDI	Modèle Dynamique Inverse
MGD	Modèle Géométrique Direct
MGI	Modèle Géométrique Inverse
PD	Proportionnel Dérivée
RMSE	Root Mean Square Error
TDC	Time Delay Control
TDE	Time Delay Estimation
TDCFOSM	Time Delay Control with Fractional Order Sliding Mode
TDCSM	Time Delay Control with Sliding Mode
TSM	Terminal Sliding Mode

# Liste des symboles

$\ddot{\theta}_i$	Accélération angulaire de bras
$\ddot{\alpha}_i$	Accélération angulaire de moteur
$\ddot{q}_i$	Accélération articulaire
$\dot{P}$	Accélération de la nacelle
$q_i$	Coordonnée articulaire
$P$	Coordonnées de la nacelle
$\tau_i$	Couple apporté au bras i
$\tau_m$	Couple fourni par le moteur
$\gamma$	Fonction gamma
$e'(t)$	La dérivée de l'erreur de poursuite
$e(t)$	L'erreur de poursuite
$l_1$	Longueur de bras
$l_2$	Longueur des barres parallèles
$m_b$	Masse de bras
$m_c$	Masse de coude
$m_n$	Masse de la nacelle
$m_{ab}$	Masse des barres parallèles
${}_aD_t^\mu$	Opérateur d'ordre fractionnaire
$\theta_i$	Position angulaire de bras
$\alpha_i$	Position angulaire de moteur
$r$	Rapport de masse de l'avant bras
$\dot{\theta}_i$	Vitesse angulaire de bras
$\dot{\alpha}_i$	Vitesse angulaire de moteur
$\dot{q}_i$	Vitesse articulaire
$\dot{P}$	Vitesse de la nacelle

# Introduction générale

Ces dernières années, un intérêt croissant a été dédié aux robots parallèles. La structure de ces robots possède en fait un certain nombre d'avantages par rapport aux robots séries. Le plus important est certainement la possibilité de maintenir les moteurs fixés sur la base permettant ainsi une réduction importante de la masse mobile active de la structure du robot. En plus de garder les moteurs sur la base du robot est une exigence quand un entraînement direct est utilisé. Les robots parallèles sont donc bien adaptés à l'actionnement direct. Un autre avantage des robots parallèles est leur grande rigidité. Ces caractéristiques permettent des manipulations plus précises et beaucoup plus rapides. Ils souffrent cependant surtout d'un espace de travail limité qui peut être considéré comme l'intersection de l'espace de travail individuel de chaque bras de série constituant le robot. Cette limitation a été partiellement résolue par la découverte de Le robot DELTA, un nouveau type de robot parallèle dédié à la manipulation d'objets légers. Il a été développé par le professeur Clavel dans les années 80 à l'École Polytechnique Fédérale de Lausanne (EPFL) [1].

Dans le cadre de notre projet de fin d'étude et pour les raisons qu'on vient d'évoquer, nous avons choisi de travailler sur le robot ISIR88 à 3 degré de liberté conçu par l'équipe de Clavel de l'EPFL et disponible au niveau du Laboratoire de Commande des Processus de l'École Nationale Polytechnique.

Le présent mémoire est structuré en quatre chapitres :

Dans le premier chapitre, nous présenterons la catégorie des robots parallèles avec quelques définitions de base, leurs domaines d'application et les notions nécessaires pour entamer notre sujet principal qui est la modélisation et la commande du robot Delta.

Dans le deuxième chapitre, on présente les modèles : géométrique, cinématique et dynamique du robot Delta avec quelques caractéristiques et spécifications géométriques nécessaires pour notre étude comme l'espace de travail, les configurations singulières, et la génération de la trajectoire.

Dans le troisième chapitre, on commence par une introduction mathématique aux opérateurs fractionnaires puis on présentera un régulateur qui se base sur ces opérateurs qui est le PD d'ordre fractionnaire ainsi que les résultats de la simulation et l'implémentation de ce dernier sur le robot Delta ISIR88 en le comparant avec le PD classique.

Dans le quatrième et dernier chapitre, nous allons synthétiser théoriquement les commandes que nous allons appliquer sur le robot Delta ISIR88 à savoir la commande Time delay control (TDC) simple, la TDC avec mode glissant et la TDC avec mode glissant d'ordre fractionnaire. Une interprétation des résultats expérimentaux est aussi donnée avec une étude comparative entre les 3 commandes.

Enfin, nous terminons par une conclusion générale englobant les perspectives et les investigations futures à poursuivre.

# État de l'art

## 1.1 Introduction

D'après l'Association Française de Normalisation (AFNOR), un robot industriel est un "manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel" [2].

Dans ce chapitre nous présenterons la catégorie des robots parallèles avec quelques définitions de base, leurs domaines d'application et les notions nécessaires pour entamer notre sujet principal qui est la modélisation et la commande du robot Delta.

## 1.2 Définitions de base

### 1.2.1 Les articulations

Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté ( $n$ ) de l'un par rapport à l'autre. En robotique le cas le plus fréquent est  $n = 1$  dans ce cas l'articulation est soit rotoïde soit prismatique.

#### Articulation rotoïde :

Il s'agit d'une articulation de type pivot réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre deux corps est donnée par l'angle autour du même axe.

### Articulation prismatique :

Il s'agit d'une articulation de type glissière qui réduit le mouvement entre deux corps à une translation le long d'un axe commun. La position relative entre deux corps est mesurée par la distance le long du même axe.

### 1.2.2 L'espace cartésien ou opérationnel

L'espace opérationnel d'un robot est celui dans lequel est représentée la position de l'organe terminal (point  $P$ ). On représente les coordonnées d'un point dans l'espace cartésien par le vecteur :

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (1.1)$$

Avec la représentation graphique (Figure 1.1) du point  $P$  :

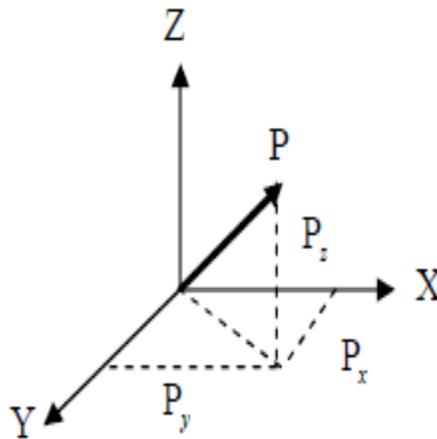


FIGURE 1.1: Représentation d'un point

### 1.2.3 L'espace articulaire

L'espace articulaire est celui dans lequel est représenté la position de tous les corps du robot, dans le cas de l'Orthoglide c'est la position des 3 actionneurs (trois prismatiques). On représente les coordonnées articulaires par :

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (1.2)$$

## 1.2.4 Configuration singulière

Les singularités sont un problème des architectures parallèles, aux configurations singulières le robot parallèle est exposé à un comportement inhabituel comme la perte ou un gain d'un degré de liberté, un mouvement dans une direction inatteignable ou encore un mouvement de l'outil final lorsque les actionneurs sont verrouillés [3].

## 1.3 Types des robots manipulateurs

### 1.3.1 Robots sériels

Les robots sériels, caractérisés par le fait qu'une seule chaîne cinématique relie la base fixe à l'organe terminal mobile, cette chaîne est une succession de corps reliés entre eux par une articulation d'un seul degré de liberté (rotatoire ou prismatique). Chaque articulation est muée par un actionneur.

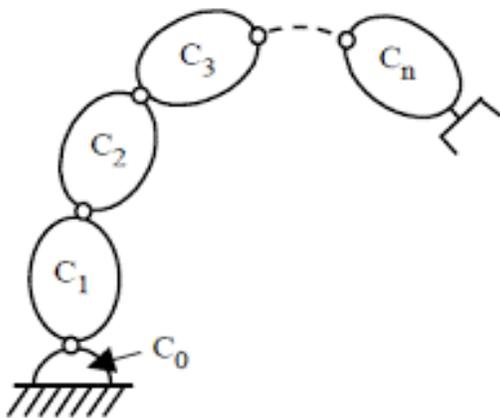


FIGURE 1.2: Architecture des robots sériels



FIGURE 1.3: Robot SCARA Fanuc SR-3iA

#### Avantages :

- Grand volume de travail dû à la structure ouverte de la chaîne.
- Modélisation et commande simples.

#### Inconvénients :

- Leur hystérésis inévitable et leur flexibilité hors axe s'accumulent le long de la chaîne cinématique du bras.
- Une inertie élevée due aux masses réparties sur toute la chaîne cinématique (actionneurs et organes de transmission), peut engendrer des erreurs de positionnement, des vibrations et surtout limiter la rapidité
- Faible rapport (charge transportable / masse totale du robot) qui n'est jamais supérieur à 0.3 dans le meilleur des cas

- Précision médiocre, ceci est dû à l'accumulation d'erreurs le long de la chaîne cinématique.

### 1.3.2 Robots parallèles

Les robots parallèles caractérisés par le fait que plus d'une chaîne cinématique assure la liaison entre la base fixe et l'organe terminal mobile. Certains robots peuvent être une combinaison de cinématiques sérielles et parallèles. Vu les caractéristiques de ces cinématiques, nous classerons pour simplifier les robots hybrides avec les robots parallèles.

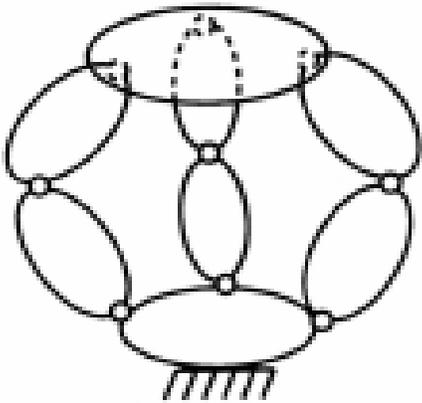


FIGURE 1.4: Architecture parallèle



FIGURE 1.5: Robot parallèle Fanuc

#### Avantages :

- Un rapport masse transportable / masse en mouvement du robot très supérieur à celui des robots sériels.
- Une bonne rigidité due aux chaînes cinématiques fermées.
- Une meilleure précision que les robots sériels.
- Possibilité de déplacements à très haute vitesse.

#### Inconvénients :

- Modélisation souvent complexe.
- Un volume de travail faible par rapport à l'encombrement de la machine
- Des amplitudes en orientation souvent limités à cause des singularités et des collisions internes.

### 1.3.3 Récapitulatif

Une brève comparaison est présentée dans le tableau suivant [4] :

Critères de comparaison	Robot série	Robot parallèles
Charge transportable	Faible	Grande
Masse mobile(interie)	Grande	Faible
Rigidité	Faible	Grande
Précision	Faible	Grande
Vitesse et Accélération	Faible	Très grandes
Ratio volume de travail/encombrement	Grande	Faible
Volume de travail	Grande	Faible
Singularité dans l'espace de travail	Rare	Présence de singularité
Obtention du modèle géométrique direct	Simple	Difficile
Obtention du modèle géométrique inverse	Simple	Difficile
Synthèse des lois de commande	Difficile	Difficile
Conception	Simple	Difficile

TABLE 1.1: Tableau comparatif entre robot série et robot parallèle

## 1.4 Historique

Les travaux théoriques sur les mécanismes parallèles remontent à plusieurs siècles quand les géomètres français et anglais étaient particulièrement intéressés par les polyèdres. Cependant, un des premiers robots parallèles a été conçu par James E. Gwinnett, un agriculteur aux États-Unis, pour lequel il a déposé un brevet en 1928. Son invention était une plate-forme à mouvement sphérique à 3 degrés de liberté (ddl) pour une utilisation dans les salles de cinéma (Figure 1.6).

En 1934, Willard L. G. Pollard a déposé un brevet pour une machine de pulvérisation de peinture. L'invention comporte un système de commande et un manipulateur. Le système de commande se compose de films perforés, et le manipulateur est essentiellement un robot à cinq barres (Figure 1.7).

En 1947 Gough a établi les principes de base d'un mécanisme cinématique avec une structure en boucle fermée (Figure 1.8) qui permet le positionnement et l'orientation d'une plate-forme mobile afin de tester l'usure des pneus.

En 1965, Stewart a suggéré l'utilisation d'une telle structure pour les simulateurs de vol et le mécanisme Gough est parfois appelé plate-forme de Stewart.

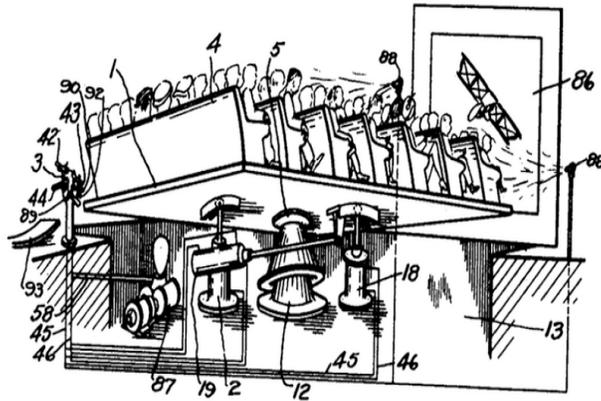


FIGURE 1.6: Un des premiers robots parallèles, inventé en 1928 par Gwinnett

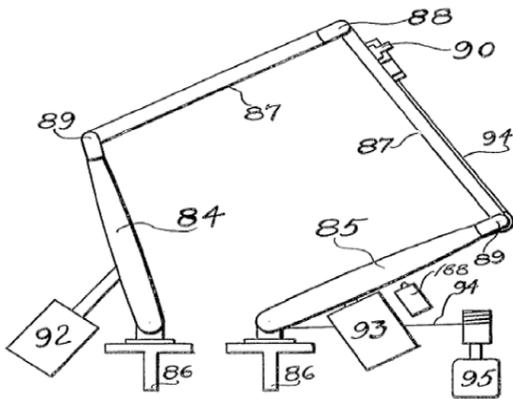


FIGURE 1.7: Robot parallèle de Pollard 1934

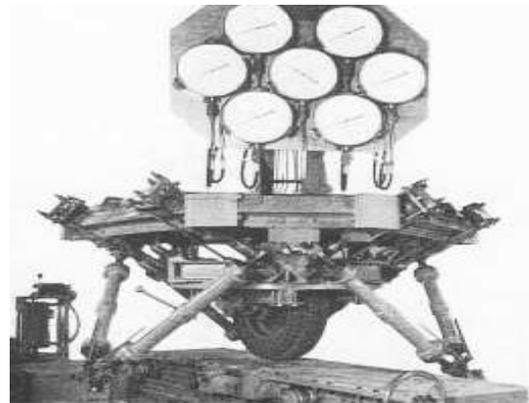


FIGURE 1.8: Plateforme de Gough 1947

Sur la base de la structure de la plate-forme de Stewart, Hexcel Corporation a développé les robots de la série R (Figure 1.9-a) pour des applications biomédicales et les robots de la série P (Figure 1.9-b) pour des applications industrielles, telles que le soudage, l'ébavurage et le fraisage.

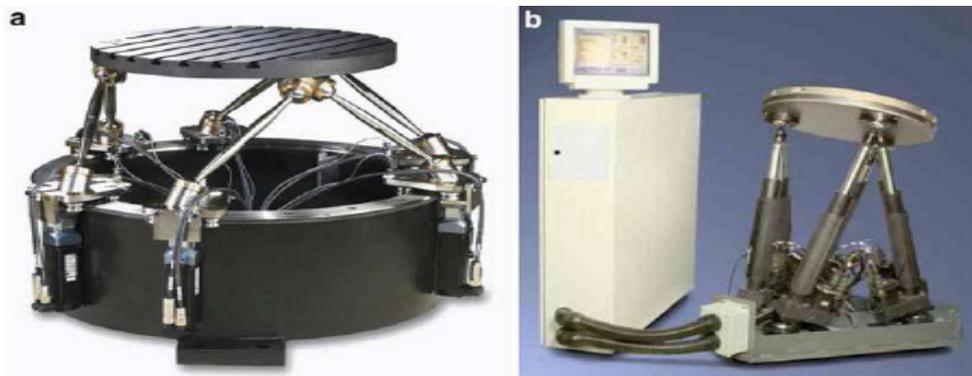


FIGURE 1.9: Les applications de la plateforme de Stewart par Hexcel Corporation (a) série R (b) série P

L'autre événement majeur de la robotique parallèle est l'invention du premier robot parallèle léger, le robot Delta, par le professeur Clavel en 1985 à l'EPFL (École Polytechnique Fédérale de Lausanne). L'objectif recherché est le déplacement de pièces légères à cadence élevée; les applications visées sont dans les domaines du conditionnement dans le secteur alimentaire, la dépalettisation et la palettisation au début ou à la fin d'une chaîne d'assemblage, le montage de composantes mécaniques ou électriques sur des platines ou des circuits imprimés. Le premier robot Delta développé est à quatre degrés de liberté, avec une structure mobile comportant les originalités suivantes :

- Les trois premiers degrés de liberté forment un parallélogramme de l'espace à structure parallèle.
- Le maintien des orientations de l'élément mobile (la nacelle) est assuré de façon totalement passive par la disposition cinématique (sans nécessiter de capteur, de contrôleur, ni d'apport d'énergie).
- Tous les moteurs sont fixés sur le bâti et l'inertie totale est très faible

L'idée de base derrière la conception du robot Delta est l'utilisation des parallélogrammes. Un parallélogramme permet une articulation de sortie à rester dans une orientation fixe par rapport à une articulation d'entrée. L'utilisation de ces trois parallélogrammes restreint complètement l'orientation de la plate-forme mobile qui reste uniquement en translation selon les trois degrés de liberté. Les liaisons d'entrée des trois parallélogrammes sont montées en rotation sur des leviers par des articulations rotoïdes. Les liaisons pivots des leviers rotatifs sont actionnées de deux manières différentes : avec rotation (DC ou AC servo) avec des moteurs ou des actionneurs linéaires. Enfin, une quatrième jambe est utilisée pour transmettre un mouvement rotatif de la base à un effecteur monté sur la plate-forme mobile.

Pour améliorer la précision et la vitesse du robot Delta, Alain Codourey a réalisé un robot Delta à 3ddl et à entraînement direct, c'est-à-dire, sans engrenages entre les actionneurs et les bras. Les avantages de cette construction sont principalement l'élimination du jeu et l'absence de frottement, il a moins de composants et il est moins bruyant.

L'utilisation d'actionneurs montés à la base et des articulations de faibles masses permettent à la nacelle de réaliser des accélérations jusqu'à  $50G$  ( $G$  : force de gravité, elle est égale à  $9.8m/s^2$ ) dans les environnements expérimentaux et  $12G$  dans les applications industrielles. Cela rend le robot Delta un candidat idéal pour les opérations de prise et dépose des objets légers (de  $10gr$  à  $1kg$ ) [5].

## 1.5 Le robot Delta

### 1.5.1 Définition et Description

Le robot Delta est considéré comme le premier robot parallèle léger. Il a été développé par le professeur Clavel dans les années 80 à l'École Polytechnique Fédérale de Lausanne

(EPFL) [1]. C'est l'un des premiers robots parallèles ayant moins de mobilités que les six ddl habituels. Le robot Delta original était muni de trois actionneurs rotatifs fixés sur la base. Il fait partie de la famille des manipulateurs à trois ddl et plus particulièrement des manipulateurs pour translation. Les trois ddl du robot Delta lui permettent de positionner son organe terminal dans l'espace avec une orientation fixe aux tolérances de fabrication et de montage près. Il existe également un robot Delta dit linéaire, pour lequel les actionneurs rotatifs sont remplacés par des actionneurs linéaires [6]. L'Orthoglide fait partie de cette famille des robots Delta à actionneurs linéaires et présente en plus la particularité d'être isotrope au centre de son volume de travail [7].

Le robot Delta est constitué de trois chaînes cinématiques identiques. L'utilisation de parallélogrammes spatiaux permet à la nacelle de se déplacer parallèlement à la base du robot. Une variante de ce robot à quatre ddl existe. Une chaîne RUPU est ajoutée pour obtenir la rotation infinie de l'organe terminal par rapport à la normale sur le plan de la nacelle.

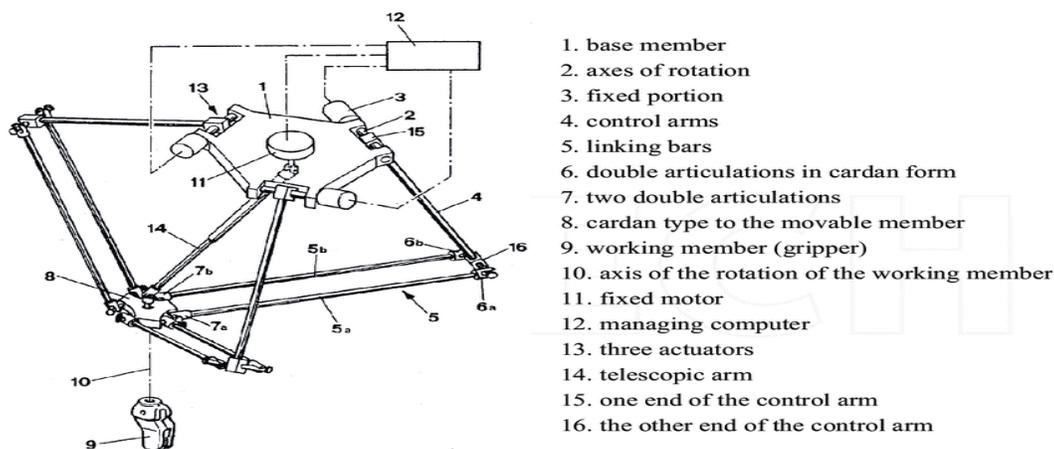


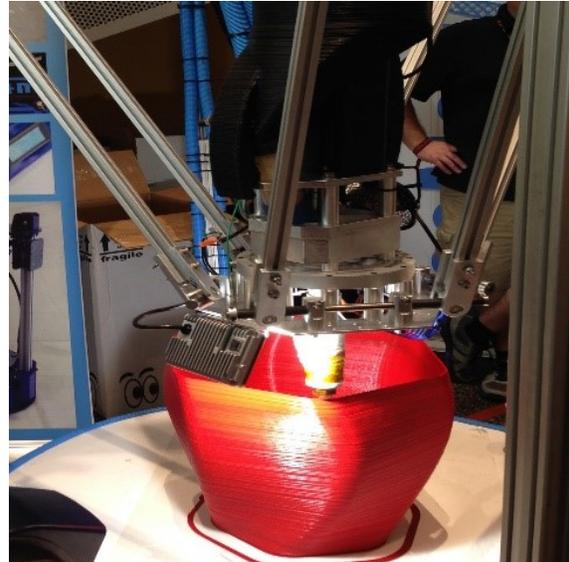
FIGURE 1.10: Schéma descriptif de robot DELTA à 4ddl

### 1.5.2 Applications :

- Industrie électronique : assemblage, montage de composants...etc.
- Industrie alimentaire et pharmaceutique : conditionnement et emballage de chocolats, biscuits, médicaments ... etc.
- Application médicale et télé-chirurgie.
- Les systèmes de micro-positionnement et impression 3D.
- La structure d'un robot delta peut également être également utilisée pour créer des contrôleurs haptiques.
- Le robot delta est utilisé pour des différents applications comme le dessin, jeux d'échecs ... etc.



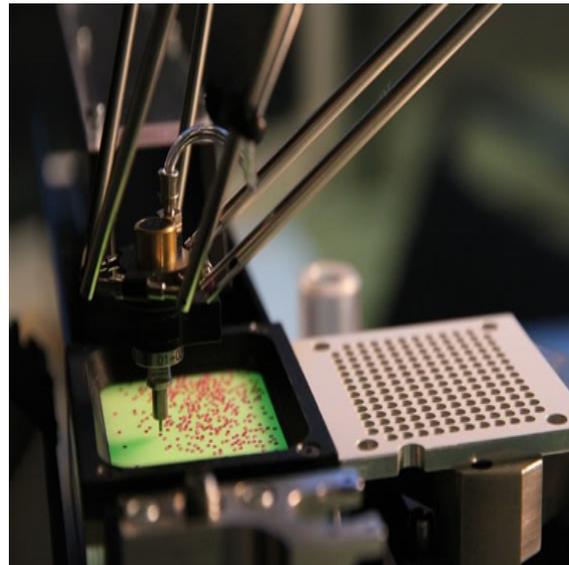
(a) Robot Delta en industrie pharmaceutique



(b) Impression 3D avec robot Delta



(c) Robot Delta en chirurgie



(d) Robot Delta en industrie électronique

FIGURE 1.11: Application de robot DELTA

### 1.5.3 Le robot DELTA ISIR88

Ce robot parallèle à 4 degrés de liberté ( $x$ ,  $y$ ,  $z$  et  $\sigma$ ) représenté par la figure (1.12) [8], spécialement dédié à la manipulation de pièces légères (0 à 20 g), peut atteindre une cadence de travail deux à trois fois supérieur à celle des robots séries à titre d'exemple, il peut assurer le transfert de 3 pièces par seconde par 3 mouvements aller et retour complets [1].

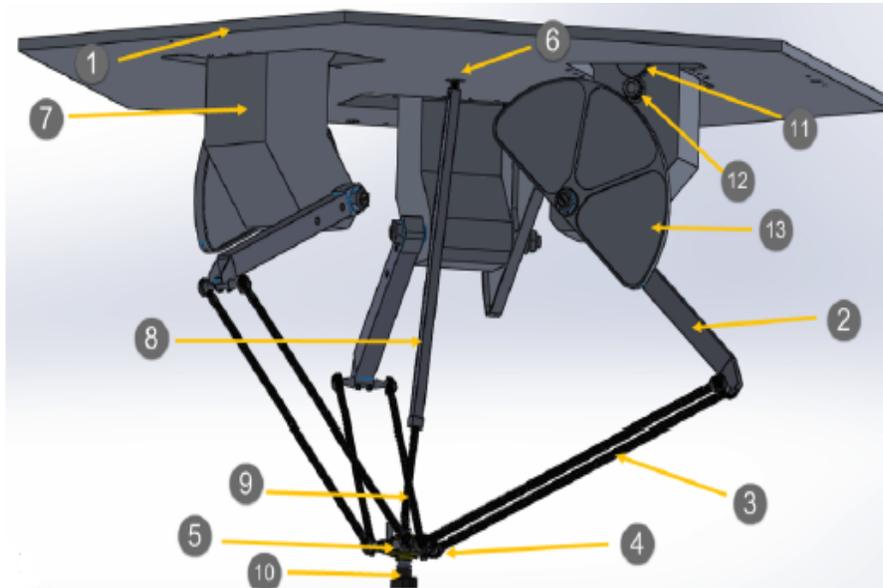


FIGURE 1.12: Schéma mécanique du robot Delta ISIR88

La structure est un parallélogramme constitué de 3 chaînes cinématiques parallèles identiques. Aux extrémités de ces chaînes on trouve une base fixe (1) en haut et la nacelle en bas (5).

Chaque chaîne cinématique est constituée d'un bras (2) joint à deux barres parallèles (3). Le mouvement de chaque bras est assuré par un moteur à courant continu (7) placé sur la base fixe et dont les caractéristiques sont détaillées ci-après.

Les articulations aux extrémités des barres parallèles sont du type rotule (4), chaque barre peut donc tourner autour de son axe longitudinal; cette rotation ne perturbe pas le comportement de cette structure articulée formant le parallélogramme de l'espace. La transmission entre un moteur et un bras n'est pas directe; en effet elle est assurée par un système de réduction constitué par une poulie (11), un pignon (12) et une roue (13) afin d'obtenir un rapport de réduction égal à 12 :1. (Couple au niveau du bras est égal à 12 fois le couple au généré par le moteur). Donc si on note par  $\alpha_i$  la position angulaire du bras et  $\tau_i$  le couple rapporté au bras, et  $\alpha_m$  la position angulaire du moteur et  $\tau_m$  le couple fourni par le moteur, nous obtenons les relations suivantes :

$$\begin{cases} \tau_i = 12 \tau_m \\ \alpha_i = \frac{1}{12} \alpha_m \end{cases} \quad (1.3)$$

Le fait que les actionneurs se situent sur la base fixe et que la chaîne cinématique est légère permet à la nacelle d'atteindre de grandes accélérations. Mais la structure fermée de la chaîne ne permet qu'un espace restreint de travail qu'on peut décrire comme un cylindre.

A noter qu'un quatrième degré de liberté peut être rajouté moyennant un arbre télescopique (8) entraîné par le moteur (6) et muni par une articulation de type cardan (9) permettant la rotation de l'effecteur (10) autour de lui-même [9]. Durant notre étude, cette rotation n'est pas prise en considération (i.e. l'arbre télescopique est enlevé).

## Caractéristiques mécaniques

Paramètre	Symbole	Valeur
Longueur de bras	$l_1$	0.205 m
Longueur des barres parallèles	$l_2$	0.380 m
Masse de la nacelle	$m_n$	0.042 Kg
Masse de bras	$m_b$	0.098 Kg
Masse des barres parallèles	$m_{ab}$	0.028 Kg
Masse de coude	$m_c$	0.015 Kg
Rapport de masse de l'avant-bras	r	2/3

TABLE 1.2: Caractéristiques mécaniques de robot ISIR88

## Caractéristiques des actionneurs

Les actionneurs de notre robot sont des moteurs à courant continu de la marque "YASKAWA UGTMEM-03 LB2".



FIGURE 1.13: Photo réelle de moteur UGTMEM-03 LB2

Dont les spécifications sont présentés ci-dessous :

Spécifications	Valeurs
Couple nominal maximal	1.01 N.m
Couple nominal	0.49 N.m
Constante de couple	0.119 N.m/A $\pm$ 10%
Résistance d'enroulement d'induit	1.36 $\Omega \pm$ 10%
Inductance d'induit	1.2 mH
Courant maximal	8.7 A
Constante de tension	12.5 V/1000rpm $\pm$ 10%
Inertie	0.372 Kg.m <sup>2</sup> x 10 <sup>-4</sup>
Taux de puissance	6.44 KW/s
Vitesse nominale	1000 r/min
Vitesse maximale sans charge	6000 r/min

TABLE 1.3: Caractéristiques des actionneurs de robot Delta ISIR88

## 1.6 Conclusion

Ce chapitre commence par une présentation des robots manipulateurs en général, puis il sera consacré au robot Delta, objet de notre travail de recherche qui représente la dernière génération des robots parallèles et il est caractérisé par sa légèreté, sa précision et ses accélérations très élevées.

De plus, nous avons présenté un bref historique qui nous a permis de remonter jusqu'au robot Delta. Par ailleurs, une description du robot Delta ISIR88 disponible au niveau de Laboratoire de Commande des Processus en mettant en avant ses constituants et ses caractéristiques mécaniques. En fait, il s'agit d'un modèle particulier du robot Delta à entraînement rotatif contenant des réducteurs entre l'actionneur et le bras.

# Modélisation du robot Delta

## 2.1 Introduction

Afin de commander, simuler le comportement dynamique du robot, il est nécessaire de le modéliser, ces modèles permettront de connaître le comportement géométrique, cinématique et dynamique de la machine. Ces outils mathématiques doivent permettre de passer de l'espace de travail de l'outil final (espace cartésien) à l'espace des coordonnées des actionneurs (espace articulaire) et réciproquement, d'où la notion de modèle inverse et direct [3]. Il est présenté dans ce chapitre, les modèles géométriques, cinématiques, dynamiques direct et inverse.

## 2.2 Modélisation géométrique

Le modèle géométrique du robot permet d'exprimer les coordonnées opérationnelles  $(x, y, z)$  de sa nacelle en fonction des angles articulaires  $(\alpha_1, \alpha_2, \alpha_3)$  ou l'inverse.

### 2.2.1 Modèle géométrique direct

La figure (2.1) représente le modèle géométrique direct du robot Delta :



FIGURE 2.1: Modèle géométrique direct de Delta

Considérons trois sphères chacune avec le centre au coude ( $C_i$ ) de chaque chaîne de bras du robot, et avec les longueurs d'avant-bras ( $l_i$ ) comme rayon. Le modèle géométrique direct d'un robot Delta-3 peut alors être calculé à l'aide de l'intersection

entre ces trois sphères. En visualisant ces trois sphères, ils se croiseront à deux endroits. Un point d'intersection où  $z$  est positif et un point d'intersection où  $z$  est négatif. Basé sur l'image de base  $\{R\}$  où l'axe  $z$  est positif vers le haut, le centre de la nacelle sera le point d'intersection lorsque  $z$  est négatif. La figure (2.2) montre l'intersection entre trois sphères. Où deux sphères se croisent dans un cercle et puis la troisième sphère croise ce cercle à deux endroits [10].

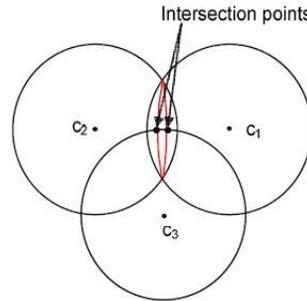


FIGURE 2.2: L'intersection des sphères

La figure (2.3) donne le schéma annoté d'un bras du Delta :

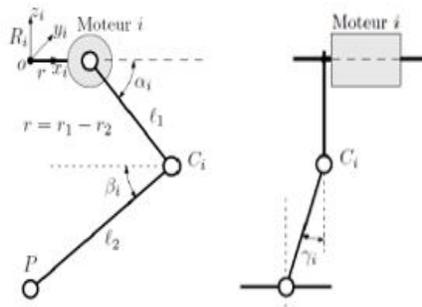


FIGURE 2.3: Vues de face et de profil d'une chaîne « i » du robot Delta

En se basant sur la figure (2.3), les coordonnées du point  $C_i$  dans le repère  $R_i$  avec ( $i = 1, 2, 3$ ) sont données par :

$$\overrightarrow{OC_i/R_i} = \begin{pmatrix} r + l_1 \cos(\alpha_i) \\ 0 \\ -l_1 \sin(\alpha_i) \end{pmatrix} \quad (2.1)$$

Pour passer au repère de base nous faisons une rotation de  $\phi_i = (0^\circ, 120^\circ, 240^\circ)$  autour de  $z$ , donc on aura :

$$\overrightarrow{OC_i/R_i} = \begin{pmatrix} r + l_1 \cos(\alpha_i) \\ 0 \\ -l_1 \sin(\alpha_i) \end{pmatrix} \quad (2.2)$$

Les équations des 3 sphères représentées dans la figure (2.2) sont données par :

$$\begin{aligned} ((r + l_1 \cos(\alpha_1)) \cos \phi_1 - x)^2 + ((r + l_1 \cos(\alpha_1)) \sin \phi_1 - y)^2 + (-l_1 \sin(\alpha_1) - z)^2 &= l_2^2 \\ ((r + l_1 \cos(\alpha_2)) \cos \phi_2 - x)^2 + ((r + l_1 \cos(\alpha_2)) \sin \phi_2 - y)^2 + (-l_1 \sin(\alpha_2) - z)^2 &= l_2^2 \\ ((r + l_1 \cos(\alpha_3)) \cos \phi_3 - x)^2 + ((r + l_1 \cos(\alpha_3)) \sin \phi_3 - y)^2 + (-l_1 \sin(\alpha_3) - z)^2 &= l_2^2 \end{aligned} \quad (2.3)$$

En posant :

$$\begin{aligned} D_i &= -l_2^2 + l_1^2 + r^2 + 2r l_1 \cos(\alpha_i) \\ E_i &= 2(r l_1 \cos(\alpha_i)) \cos(\phi_i) \\ F_i &= 2(r + l_1 \cos(\alpha_i)) \sin(\phi_i) \\ G_i &= -2l_1 \sin(\alpha_i) \end{aligned} \quad (2.4)$$

Et en notant :

$$\begin{aligned} H_1 &= E_1 G_2 - E_1 G_3 - E_2 G_1 + E_2 G_3 + E_3 G_1 - E_3 G_2 \\ H_2 &= -E_1 F_2 + E_1 F_3 + E_2 F_1 - E_2 F_3 - E_3 F_1 + E_3 F_2 \\ H_3 &= -E_1 D_2 + E_1 D_3 + E_2 D_1 - E_2 D_3 - E_3 D_1 + E_3 D_2 \\ H_4 &= F_1 D_2 - F_1 D_3 - F_2 D_1 + F_2 D_3 + F_3 D_1 - F_3 D_2 \\ H_5 &= -F_1 G_2 + F_1 G_3 + F_2 G_1 - F_2 G_3 - F_3 G_1 + F_3 G_2 \end{aligned} \quad (2.5)$$

Alors la résolution de ce système d'équations nous donne :

$$\begin{aligned} x &= z \frac{H_5}{H_2} + \frac{H_4}{H_2} \\ y &= z \frac{H_1}{H_2} + \frac{H_3}{H_2} \\ z &= -\frac{M \pm \sqrt{M^2 - 4LN}}{2L} \end{aligned} \quad (2.6)$$

Avec :

$$\begin{aligned} L &= \frac{H_5^2 + H_1^2}{H_2^2} + 1 \\ M &= 2 \frac{H_5 H_4 + H_1 H_3}{H_2^2} - \frac{H_5 E_1 + H_1 F_1}{H_2} - G_1 \\ N &= \frac{H_4^2 + H_3^2}{H_2^2} - \frac{H_4 E_1 + H_3 F_1}{H_2} + D_1 \end{aligned} \quad (2.7)$$

## 2.2.2 Modèle géométrique inverse

La figure (2.4) représente le modèle géométrique inverse du robot Delta :



FIGURE 2.4: Modèle géométrique inverse de Delta

Considérant,

$$Q_i = 2x\cos(\theta_i) + 2y\sin(\theta_i)$$

$$S = \frac{1}{l_1} - x^2 - y^2 - z^2 + l_2^2 - l_1^2 - R^2$$

Avec :

$$\theta_i = 12 \alpha_i$$

La résolution des équations donne l'expression suivante :

$$\operatorname{tg}\left(\frac{\alpha_i}{2}\right) = \frac{-2z \pm \sqrt{4z^2 + 4R^2 - S^2 + Q_i^2\left(1 - \frac{R^2}{l_1^2}\right) + Q_i\left(-2\frac{RS}{l_1} - 4R\right)}}{-2R - S - Q_i\left(\frac{R}{l_1} - 1\right)} \quad (2.8)$$

## 2.3 Espace de travail

L'espace du travail est la zone que le centre de la nacelle peut atteindre. C'est une caractéristique très importante pour le choix des solutions industrielles. En prenant compte des contraintes de construction et des singularités, l'espace de de travail du Delta est un cylindre de diamètre "D" et de hauteur "H" éloigné de "Z<sub>h</sub>" de la base fixe du robot comme le montre la figure (2.5) :

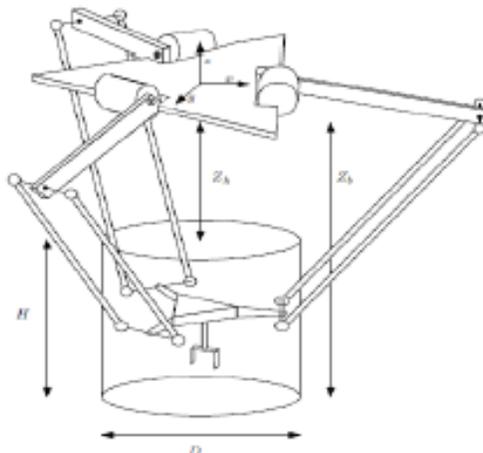


FIGURE 2.5: Espace de travail du Delta

Les expressions des paramètres "H" "D" et "Z<sub>h</sub>" sont les suivantes :

$$\begin{aligned}
 D &= 2l_2 \sin(\gamma_{max}) = 385.67mm \\
 H &= l_1 \sqrt{\left(1 + \frac{l_2}{l_1}\right)^2 - \left(\frac{r}{l_1} + 2 \frac{l_2 \sin(\gamma_{max})}{l_1 \sqrt{3}}\right)^2} - l_1 \sqrt{1 + \left(\frac{l_2}{l_1}\right)^2 - \frac{l_2}{l_1} \sqrt{3}} = 196.88mm \\
 Z_h &= -l_1 \sqrt{1 + \left(\frac{l_2}{l_1}\right)^2 - 2 \frac{l_2}{l_1} \cos(\alpha_i + \beta_i)_{min}} = -161.48mm
 \end{aligned} \tag{2.9}$$

## 2.4 Modélisation cinématique

La modélisation cinématique permet d'établir la relation entre les vitesses articulaires et les vitesses de l'organe terminal.

### 2.4.1 Modèle cinématique direct

Le modèle cinématique direct calcule les vitesses des coordonnées de l'outil final en fonction des vitesses articulaires, il est donné par la relation suivante :

$$\dot{X} = J(q)\dot{q} \tag{2.10}$$

Avec  $J(q)$ , la matrice Jacobienne de dimension  $(m \ n)$  du mécanisme,  $n$  étant le nombre d'articulations du robot et  $m$  le nombre de ddl.

Pour calculer la matrice Jacobienne du robot Delta, on peut utiliser l'ensemble des contraintes reliant l'espace opérationnel avec l'espace articulaire. Les trois équations de contraintes pour le robot Delta peut être choisies comme suit :

$$\|C_i P\|_2^2 - l_2^2 = 0 \quad i = 1, 2, 3. \tag{2.11}$$

Où la longueur de l'avant-bras  $l_2$  est supposée constante. Soit  $s_i$  dénoté le vecteur  $C_i P$ . Comme la norme euclidienne peut être écrite comme  $s_i^T s_i$ , ainsi d'après la figure (2.3) le vecteur  $s_i$  peut être écrit comme suit [10] :

$$s_i = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = R_{0/i} \left( \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \cos(\alpha_i) \\ 0 \\ l_1 \sin(\alpha_i) \end{bmatrix} \right) \tag{2.12}$$

Où

$$R_{0/i} = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) & 0 \\ \sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

On dérive l'équation (2.11) par rapport au temps :

$$s_i^T \dot{s}_i + \dot{s}_i^T s_i = 0 \tag{2.14}$$

Le premier terme de (2.14) est nul, donc :

$$s_i^T \dot{s}_i = 0 \quad (2.15)$$

La dérivée de  $s_i$  est donnée par :

$$\dot{s}_i = \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} - R_{0/i} \begin{bmatrix} -l_1 \sin(\alpha_i) \\ 0 \\ l_1 \cos(\alpha_i) \end{bmatrix} \dot{\alpha}_i = \dot{P} - b_i \dot{\alpha}_i \quad i = 1, 2, 3 \quad (2.16)$$

Avec :

$$b_i = R_{0/i} \begin{bmatrix} -l_1 \sin(\alpha_i) \\ 0 \\ l_1 \cos(\alpha_i) \end{bmatrix} \quad (2.17)$$

Ainsi :

$$s_i^T \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} - s_i^T b_i \dot{\alpha}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.18)$$

Qui peut être exprimé sous forme de matrice pour les trois bras de robot comme :

$$\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \dot{P} - \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \dot{\alpha} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.19)$$

Maintenant on peut tirer la matrice jacobienne d'un robot Delta à partir de l'équation (2.19) :

$$\dot{P} = J \dot{\alpha} \quad (2.20)$$

Avec :

$$J = \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \quad (2.21)$$

À noter que la matrice jacobienne  $J$  ne dépend pas seulement de  $\alpha$ , comme c'est généralement le cas pour les robots série, mais elle est aussi une fonction des coordonnées opérationnelles  $P$ , qui peut être calculée avec le modèle géométrique direct du robot Delta-3.

## 2.4.2 Modèle cinématique inverse

Le modèle cinématique inverse permet le passage des vitesses de la nacelle  $\dot{P}$  vers les vitesses articulaires  $\dot{\alpha}$ . On reprend l'équation (2.10) en multipliant ses deux membres par  $J^{-1}$ , nous obtenons :

$$\dot{q} = J^{-1}(q) \dot{X} \quad (2.22)$$

Cette relation impose l'inversibilité de la matrice  $J$  ( $\det(J) \neq 0$ ). En fait, les points pour lesquels la matrice  $J$  n'est pas inversible (i.e ( $\det(J) = 0$ )) représentent des singularités.

## 2.5 Singularités

En général, la résolution du déterminant de la matrice Jacobienne permet de déterminer facilement les points singuliers. En outre, la complexité de la matrice jacobienne de robot Delta a poussé le constructeur [1] [11] à faire une analyse des positions de l'avant-bras (barres parallèles) les unes par rapport aux autres. Ceci a permis de définir les configurations singulières représentées dans la figure (2.6) :

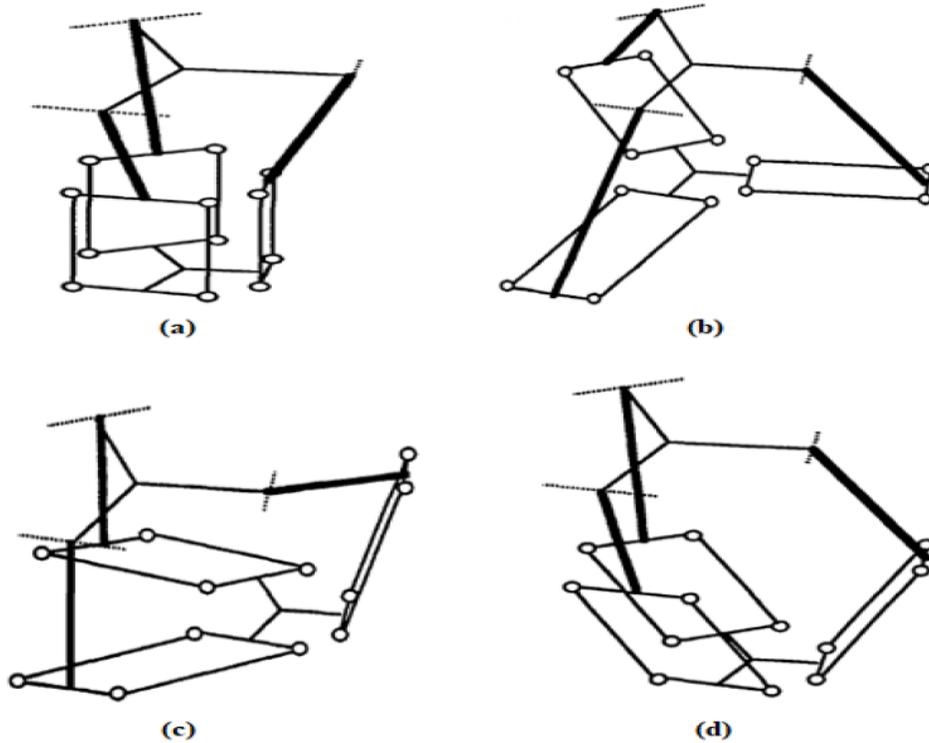


FIGURE 2.6: Singularités de Delta

## 2.6 Modèle des accélérations

Le modèle des accélérations spécifie une cartographie de l'accélération dans l'espace articulaire à l'accélération dans l'espace cartésien [10]. Et pour calculer ce dernier on peut dériver l'équation (2.19) trouvée précédemment par rapport au temps :

$$\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \ddot{P} + \begin{bmatrix} \dot{s}_1^T \\ \dot{s}_2^T \\ \dot{s}_3^T \end{bmatrix} \dot{P} - \left( \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \ddot{\alpha} + \begin{bmatrix} \dot{s}_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{s}_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{s}_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix} \dot{\alpha} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.23)$$

Compte tenu de l'équation (2.21), on peut écrire :

$$\ddot{P} = \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \left( - \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} J + \begin{bmatrix} s_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & s_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & s_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix} \right) \dot{\alpha} + J \ddot{\alpha} \quad (2.24)$$

De cette relation on peut constater que l'accélération de la nacelle  $\ddot{P}$  est reliée à l'accélération articulaire  $\ddot{\alpha}$  comme suit :

$$\ddot{P} = J \dot{\alpha} + J \ddot{\alpha} \quad (2.25)$$

## 2.7 Trajectoire

Dans cette section, on s'intéresse à la génération de la trajectoire de la nacelle dans l'espace opérationnel ainsi qu'au comportement dont évolue la trajectoire dans le temps (vitesse et accélération). La génération de trajectoires s'intéresse alors à deux aspects : aspect géométrique qui concerne les formes de trajectoire (géométriquement dans l'espace) et l'aspect temporel qui concerne l'évolution de la position, de la vitesse et de l'accélération. Les deux aspects sont fortement liés car tout erreur géométrique engendrera une erreur temporelle et vice-versa.

### 2.7.1 Lois de mouvements

#### Loi Bang-Bang

Cette loi de mouvement nous permet de contrôler l'évolution de la position, vitesse et accélération. Cependant cette loi de mouvement n'est pas optimale car si on impose le temps et la distance à parcourir, on ne peut plus imposer la vitesse et accélération maximales. La figure 2.7 représente la position, vitesse et accélération correspondant à une telle loi de mouvement.

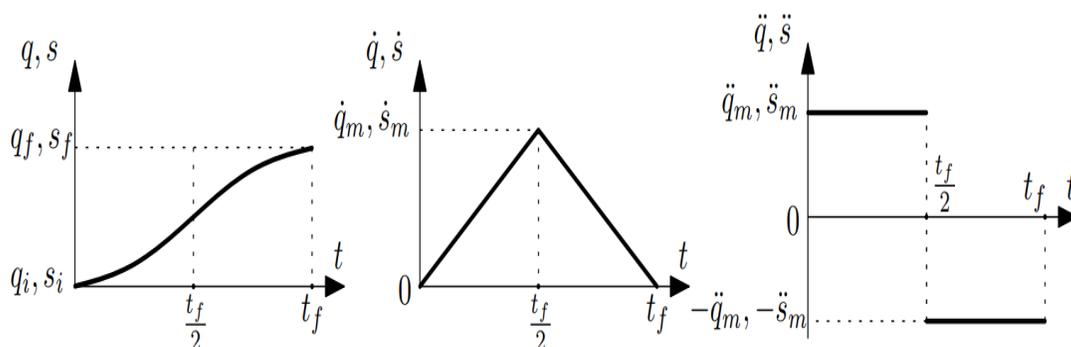


FIGURE 2.7: Loi de mouvement Bang-bang

## Loi Trapèze

Cette loi est basé sur la loi Bang-Bang, en ajoutant un palier à vitesse fixe entre les deux étapes d'accélération et de décélération. On peut donc contrôler la vitesse maximale même si le temps de parcours et la distance sont fixés (en changeant la durée du palier à vitesse constante). Le mouvement est plus uniforme (changement de la consigne d'accélération est moins brusque que la loi Bang-Bang) et le temps de parcours est optimisé, on peut imposer la vitesse et accélération maximales (couple nominal). La figure 2.8 représente la position, vitesse et accélération correspondant à une telle loi de mouvement.

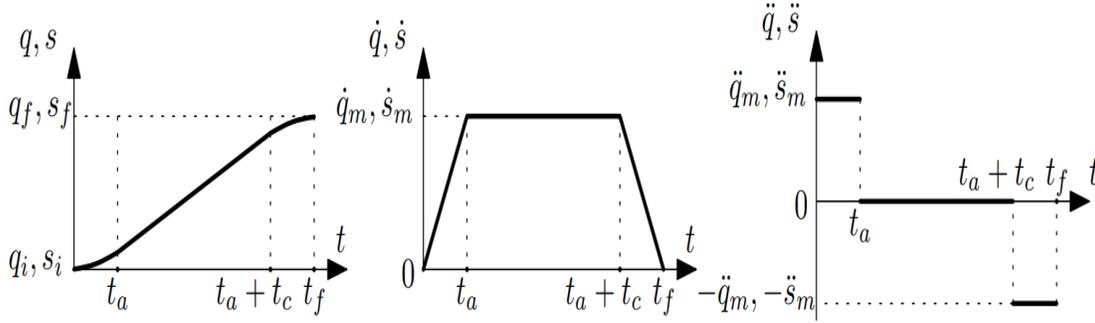


FIGURE 2.8: Loi de mouvement Trapézoïdale

Où :

- $t_a$  : temps de la phase d'accélération.
- $t_c$  : temps de la phase à vitesse constante.
- $t_f$  : temps total du parcours.
- $q_i(s_i)$  : position articulaire (curviligne) initiale.
- $q_f(s_f)$  : position articulaire (curviligne) finale.
- $\dot{q}_m(\dot{s}_m)$  : vitesse articulaire (curviligne) maximale.
- $\ddot{q}_m(\ddot{s}_m)$  : accélération articulaire (curviligne) maximale.

Les lois de mouvement selon  $s$  (variable curviligne) sont données par :

$$\begin{cases} t_f = 2t_a + t_c \\ \dot{s}_m = \ddot{s}_m t_a \\ s_f = s_i + D \\ s(t) = \begin{cases} s_i + \ddot{s}_m \left(\frac{t^2}{2}\right) & \text{pour : } t \leq t_a \\ s_i + \ddot{s}_m \left(\frac{t_a^2}{2}\right) + \dot{s}_m (t - t_a) & \text{pour : } t_a \leq t \leq t_a + t_c \\ s_i + \ddot{s}_m \left(\frac{t_a^2}{2}\right) + \dot{s}_m (t - t_a) - \ddot{s}_m \left(\frac{(t - t_a - t_c)^2}{2}\right) & \text{pour : } t_a + t_c \leq t \leq t_f \end{cases} \end{cases} \quad (2.26)$$

Avec  $D$  est la distance à parcourir.

Ces équations nous permettent de contrôler plusieurs aspects (la distance du parcours, le temps de parcours de la distance, la durée de la phase d'accélération et de

vitesse constantes et l'accélération et la vitesse maximale) en fixant les paramètres de la loi de mouvement. Cette loi sera donc utilisée dans la génération des différentes trajectoires.

## 2.7.2 Génération de trajectoires

La génération de trajectoires dans l'espace articulaire n'est pas envisageable pour le robot Delta. La génération de trajectoires dans l'espace opérationnel s'impose donc dans ce cas. La génération de trajectoire dans l'espace opérationnel passe par la connaissance de l'évolution de la coordonnée curviligne  $s(t)$  à partir de laquelle l'évolution des coordonnées articulaire  $x(t), y(t), z(t)$  est obtenu par projection (figure 2.9) [12]. Le modèle géométrique inverse est ensuite appliqué pour avoir les références articulaires à injecter aux actionneurs (figure 2.10).

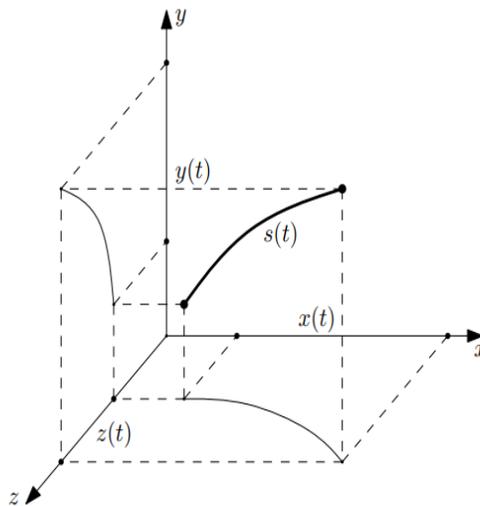


FIGURE 2.9: Projection de la coordonnée curviligne  $s(t)$  donnant  $x(t), y(t), z(t)$

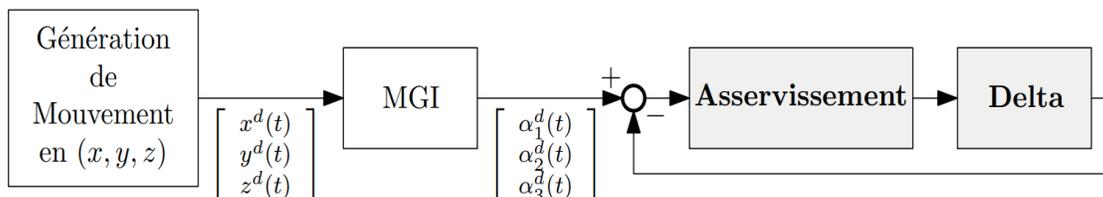


FIGURE 2.10: Génération de mouvement dans l'espace opérationnel

### Trajectoire rectiligne :

La génération de trajectoires rectilignes prend comme argument le point de départ et le point d'arrivée.

Le cas d'une trajectoire rectiligne selon l'axe  $z$  et selon l'axe  $x$  est présenté. Ces deux trajectoires nous permettront par la suite de composer une séquence de prise-dépose par une combinaison adéquate.

La projection se fait selon les équations suivantes (trajectoire selon l'axe  $x$  et  $z$  respectivement) :

$$\begin{cases} x(t) = x_i \pm s(t) \\ y(t) = y_i \\ z(t) = z_i \end{cases} \quad \begin{cases} x(t) = x_i \\ y(t) = y_i \\ z(t) = z_i \pm s(t) \end{cases} \quad (2.27)$$

### Trajectoire ellipsoïde :

En industrie, le robot Delta est principalement utilisé dans les opérations de prise-dépose. Une trajectoire en demi-ellipse peut être utilisée pour une telle tâche (figure 2.11). le cas d'un ellipsoïde selon l'axe  $x$  est présenté par les équations suivantes :

$$\begin{cases} x(t) = x_i \pm A[1 - \cos(\frac{\pi S(t)}{D})] \\ y(t) = y_i \\ z(t) = z_i + B \sin(\frac{\pi S(t)}{D}) \end{cases} \quad (2.28)$$

Où  $D = \pi \sqrt{\frac{1}{2}(A^2 + B^2)}$  est la distance parcourue par le robot.

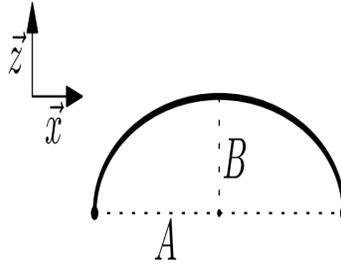


FIGURE 2.11: Trajectoire ellipsoïde

Cette trajectoire ne peut pas être utilisée pour une tâche de prise-dépose avec des différentes hauteurs entre le point de départ et le point d'arrivée, pour cela on ajoute des segments de droites aux bouts de la demi-ellipse. Néanmoins, cette trajectoire reste très intéressante puisqu'elle ne présente pas une discontinuité de courbure.

### Trajectoire de prise-dépose en segments de droites :

Trois segments de droite sont utilisés afin de construire cette trajectoire de prise-dépose. Deux segments selon l'axe  $z$  ( $H_1, H_2$ ) et un segment intermédiaire selon l'axe  $x$  ( $D$ ) (figure 2.12)

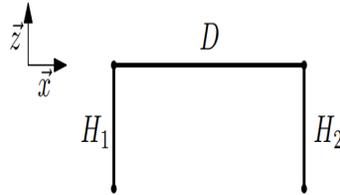


FIGURE 2.12: Trajectoire en segments de droites

Cette trajectoire peut être utilisée pour une prise-dépose à hauteur différente, par contre elle présente des discontinuités de courbure qui nous amène à arrêter le robot au niveau des points de jonction, ce qui fait perdre le robot en rapidité et rend cette trajectoire non optimale.

## 2.8 Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou les forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique par une relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (2.29)$$

Avec :

$\Gamma$  : vecteur des couples/forces des actionneurs selon que l'articulation est rotoïde ou prismatique. Dans la suite on écrit tout simplement couples ;

$q$  : vecteur des positions articulaires ;

$\dot{q}$  : vecteur des vitesses articulaires ;

$\ddot{q}$  : vecteur des accélérations articulaires ;

$f_e$  : vecteur représentant l'effort extérieure (forces et moments) qu'exerce le robot sur l'environnement.

Parmi les applications du modèle dynamique, on peut citer :

- La simulation qui utilise le modèle dynamique direct ;
- Le dimensionnement des actionneurs
- L'identification des paramètres inertiels et des paramètres de frottement du robot ;
- La commande, qui utilise le modèle dynamique inverse ;

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots, les plus utilisés sont ceux de Lagrange et de Newton-Euler et celui basé sur le travail virtuel qui sera appliqué sur notre robot dans les sections qui suivent.

### 2.8.1 Les paramètres dynamiques

Les paramètres suivants sont utilisés pour calculer le modèle dynamique du robot Delta-3. Ils sont énumérés ici pour plus de commodité [10].

La masse totale agissant sur la nacelle est la somme de la masse de la nacelle  $m_n$ , la masse

de la charge  $m_{charge}$  et les 3 masses rapportées de chacun des 3 avant-bras  $3(1-r)m_{ab}$ .

$$m_{nt} = m_n + m_{charge} + 3(1-r)m_{ab} \quad (2.30)$$

$r$  est le rapport de la masse de l'avant-bras. La valeur de  $r$  est choisie égale à  $2/3$ . La position du centre de masse pour chacune des parties supérieures de bras est calculée à l'aide de l'équation du centre de masse.

$$r_{Gb} = l_1 \frac{\frac{1}{2}m_b + m_c + rm_{ab}}{m_{br}} \quad (2.31)$$

Avec :

$$m_{br} = m_b + m_c + rm_{ab} \quad (2.32)$$

Où  $m_b$  est la masse de la partie supérieure du bras,  $m_c$  est la masse de coude,  $m_{ab}$  est la masse de l'avant-bras.

L'inertie agit sur la partie supérieure de bras est la somme de l'inertie du moteur  $I_m$ , l'inertie de frein moteur  $I_{frein}$ . L'inertie du moteur et du frein moteur peut être exprimée comme le produit de l'inertie au niveau articulaire et le carré du rapport de réduction. L' inertie au niveau articulaire est alors donnée par :

$$I_{bi} = I_m k_r^2 + I_{frein} k_r^2 + I_{bc} \quad (2.33)$$

Où  $I_{bc}$  est la somme de l'inertie créée par la masse du bras et de l'inertie créée par le masse au point d'extrémité du bras supérieur, ce qui donne :

$$I_{bc} = \frac{m_b}{3} l_1^2 + l_1^2 (m_c + rm_{ab}) = l_1^2 \left( \frac{m_b}{3} + m_c + rm_{ab} \right) \quad (2.34)$$

## 2.8.2 Modèle dynamique basé sur le principe de travail virtuel

Le travail virtuel sur un système est le travail résultant soit de forces virtuelles agissant par un déplacement réel, soit de forces réelles agissant par un déplacement virtuel [13]. Le terme déplacement peut désigner une translation ou une rotation et le terme force peut aussi désigner une force ou un moment. Le principe ne fait référence à aucun paramètre géométrique ou inertiel et peut donc être considéré comme un lien fondamental entre la dynamique et la cinématique du système [10] [14].

Comme tout système de forces généralisées peut être utilisé, l'égalité des travaux virtuels associés aux deux systèmes de coordonnées d'intérêt en robotique donne :

$$\tau^T \cdot \delta \alpha = \tau_n^T \cdot \delta X_n \quad (2.35)$$

Où  $\tau$  est le vecteur de force/couple correspondant au déplacement virtuel de l'espace articulaire  $\delta \alpha$  et  $\tau_n$  est la force/couple agissant sur la nacelle correspondant au déplacement virtuel  $\delta X_n$  dans l'espace cartésien.

Lorsqu'on introduit la relation entre la vitesse articulaire et la vitesse cartésienne donnée par l'équation (2.10) dans l'équation on peut constater que la matrice Jacobienne peut être utilisée pour transformer la force/couple agissante dans l'espace cartésien comme

$\tau^T = \tau_n^T \cdot J$  ce qui est équivalent à  $\tau = J^T \cdot \tau_n$ .

Selon l'hypothèse ci-dessus, on peut réduire le robot à 4 corps : la nacelle et les trois bras supérieurs. Ensuite, le calcul de la contribution du couple/des forces agissant sur la nacelle peut être transféré à l'espace articulaire à l'aide du Jacobien selon le principe de travail virtuel comme indiqué ci-dessus.

Il existe deux forces agissent sur la nacelle. La force de gravité  $G_n$  et la force inertielle (d'Alembert)  $F_n$ . Ses expressions sont données par :

$$G_m = m_{nt} \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T \quad F_n = m_{nt} \ddot{P} \quad (2.36)$$

La contribution de ses deux forces dans l'espace articulaire peut être calculé avec la transposée de la matrice Jacobienne :

$$\tau_n = J^T F_n = J^T m_{nt} \ddot{P} \quad (2.37)$$

$$\tau_{G_n} = J^T G_n = J^T m_{nt} \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T \quad (2.38)$$

Deux types de couples agissent sur les articulations actionnées de chaque bras. Le couple  $\tau_{Gb}$  produit par la force gravitationnelle de chaque bras supérieur et le couple  $\tau_b$  produit par la force d'inertie (d'Alembert) agissant sur chaque bras supérieur. La contribution gravitationnelle peut être calculée comme la force qui agit perpendiculairement à la partie supérieure du bras à travers le centre de masse, comme le montre la figure (2.13).

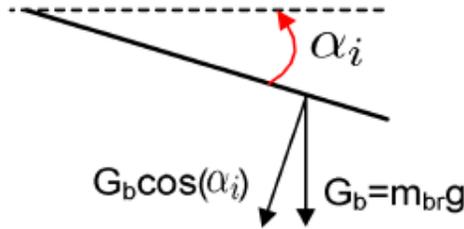


FIGURE 2.13: Force gravitationnelle agissant sur la partie supérieure du bras d'un robot Delta

Ça donne :

$$\tau_{Gb} = r_{Gb} G_b \begin{pmatrix} \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \end{pmatrix}^T \quad (2.39)$$

Où  $G_b$  est la force gravitationnelle agissant sur le point de centre de masse de chaque bras supérieur. La contribution du couple à partir de chaque bras supérieur jusqu'aux articulations actionnées peut être exprimée par :

$$\tau_b = I_b \ddot{\alpha} \quad (2.40)$$

Où  $I_b$  est la matrice d'inertie du bras dans l'espace articulaire et il est donné par :

$$I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \quad (2.41)$$

D'après le principe de d'Alembert la contribution de toutes les forces inertielles doit être égale à la contribution de toutes les forces non inertielles, ceci appliqué au niveau articulaire conduit à :

$$\tau + \tau_{Gn} + \tau_{Gb} = \tau_b + \tau_n \quad (2.42)$$

Où  $\tau$  est le vecteur des couples qui doivent être appliqués aux trois articulations actionnées.  $\tau_n$  contient le terme  $\ddot{P}$  qui peut être exprimé dans l'espace articulaire par l'équation (2.25). Cela donne :

$$\tau = (I_b + m_{nt} J^T J) \ddot{\alpha} + (J^T m_{nt} \dot{J}) \dot{\alpha} - (\tau_{Gn} + \tau_{Gb}) \quad (2.43)$$

Qui peut s'écrire comme suit :

$$\tau = M(\alpha) \ddot{\alpha} + C(\alpha, \dot{\alpha}) \dot{\alpha} + G(\alpha) \quad (2.44)$$

Où  $M(\alpha) \in \mathbb{R}^{3 \times 3}$  est la matrice d'inertie,  $C(\alpha, \dot{\alpha}) \in \mathbb{R}^{3 \times 3}$  décrit la prise en compte des forces centrifuges et de Coriolis et  $G(\alpha) \in \mathbb{R}^{3 \times 3}$  contient les forces de gravité agissant sur le manipulateur.

## 2.9 Conclusion

A travers ce chapitre, nous avons présenté les modèles : géométrique, cinématique et dynamique du robot Delta ISIR88. Pour le modèle géométrique, l'absence d'une méthode systématique nous a conduit à adopter le modèle établi par R. Clavel. Puis, à l'aide d'un système d'équations de contraintes reliant les coordonnées articulaires aux celles opérationnelles, A. Olsson nous a proposé une méthode pour déterminer le modèle cinématique qui sera dérivé plus tard pour obtenir le modèle des accélérations. Ensuite, une modélisation dynamique a été effectuée en se basant sur le principe de travail virtuel.

Pour une meilleure exploitation du robot Delta ISIR88, nous avons défini son espace de travail, ses singularités et la méthode de génération des trajectoires de prise-dépose.

# Commande PD d'ordre fractionnaire

## 3.1 introduction

Malgré un éventail très large de processus existants et malgré les diverses formulations envers les exigences requises, le correcteur proportionnel intégral dérivé (*PID*), dont l'architecture relativement dépouillée consiste en seulement trois termes distincts, s'est imposé dans plus de 90% des boucles de contre-réaction. La prédominance incontestée de ce type de correcteur provient, outre de sa simplicité extrême, des performances qu'il peut offrir aux systèmes en boucle fermée, satisfaisant très souvent les cahiers des charges, si ses paramètres sont choisis judicieusement [15].

Néanmoins, dû à l'augmentation de la complexité des systèmes industriels et aux exigences très élevées en matière de performances et robustesse, pour beaucoup d'applications industrielles, le correcteur classique *PID* ne peut être capable d'accomplir les performances ainsi que la robustesse désire. Alors une des possibilités pour résoudre ces exigences est d'utiliser les correcteurs d'ordre fractionnaire qui ont plus de paramètres à régler.

Dans ce chapitre on va commencer par une introduction mathématique aux opérateurs fractionnaires puis on présentera un des correcteurs qui se base sur ces opérateurs qui est le *PID* d'ordre fractionnaire ainsi que les résultats de la simulation et l'implémentation de ce dernier sur le robot Delta ISIR88.

## 3.2 Opérateurs d'ordre fractionnaire

### 3.2.1 Définition mathématique

Le calcul fractionnaire est une généralisation de l'intégration et de la différentiation à l'opérateur fondamental d'ordre non entier  ${}_a D_t^\mu$  où  $a$  et  $t$  sont des limites de l'opération [16]. L'opérateur intégro-différentiel continu est défini comme :

$${}_a D_t^\mu = \begin{cases} \frac{d^\mu}{dt^\mu} & \mu > 0 \\ 1 & \mu = 0 \\ \int_a^t (dt)^{-\mu} & \mu < 0 \end{cases} \quad (3.1)$$

Où  $\mu$  est l'ordre de l'opération, généralement  $\mu \in \mathbb{R}$ . Les définitions utilisées pour la diffère-intégral d'ordre fractionnaire général sont la définition de Grünwald-Letnikov, la définition de Riemann Liouville et la définition de Caputo.

### Définition de Grünwald-Letnikov

Une des définitions les plus rencontrées de la dérivée d'ordre fractionnaire est appelée définition de Grünwald Letnikov [16], donnée par :

$${}_a D_t^\mu f(t) = \frac{{}_a d_t^\mu f(t)}{dt^\mu} = \lim_{h \rightarrow 0} \left\{ \frac{1}{h^\mu} \sum_{k=0}^N (-1)^k \binom{N}{k} f(t - kh) \right\} \quad \{n, N \in \mathbb{N}\} \quad (3.2)$$

Cette définition provient de la généralisation aux ordres fractionnaires de la formule donnant la dérivée d'ordre entier supérieur  $n$  d'une fonction continue  $f(t)$ , soit :

$$D^n f(t) = \lim_{h \rightarrow 0} \left\{ \frac{1}{h^n} \sum_{k=0}^N (-1)^k \binom{N}{k} f(t - kh) \right\} \quad \{n, N \in \mathbb{N}\} \quad (3.3)$$

Sous la condition que les dérivées  $f^{(k)}(t) (k = 1, 2, \dots, m+1)$  sont continues dans  $[t_0, t]$  avec  $m \leq \mathbb{R}(\mu) \leq m+1$ , la définition de Grünwald-Letnikov peut être aussi donnée par :

$${}_a D_t^\mu f(t) = \sum_{k=0}^m \left\{ \frac{f^{(k)}(a)(t-a)^{-\mu+k}}{\gamma(-\mu+k+1)} + \frac{1}{\gamma(-\mu+m+1)} \int_a^t (t-\tau)^{m-\mu} f^{(m+1)}(\tau) d\tau \right\} \quad (3.4)$$

Où  $\gamma(\mu)$  est la fonction Gamma généralisée aux nombres fractionnaires, définie par :

$$\gamma(\mu) = \int_0^\infty \exp^{-x} x^{-\mu-1} dx \quad (3.5)$$

### Définition de Riemann-Liouville

Une autre définition très utilisée de la dérivée d'ordre fractionnaire est appelée définition de Riemann-Liouville. Inspirée de la formule de Cauchy donnée par l'équation :

$${}_a D_t^\mu f(t) = \int_a^t \frac{(t-x)^{n-1}}{(n-1)!} f(x) dx \quad (3.6)$$

La définition complète de Riemann-Liouville de la dérivée ou de l'intégrale d'ordre fractionnaire  $\mu$  de  $f(t)$  est donnée par [16] :

$${}_a D_t^\mu f(t) = \begin{cases} \frac{1}{\Gamma(-\mu)} \int_a^t (t-x)^{-\mu-1} f(x) dx, & \text{si } \Re(\mu) < 0 \\ f(t), & \text{si } \Re(\mu) = 0 \\ D^n [{}_a D_t^{\mu-n} f(x)], n = \min k \in \mathbb{N} : k > \Re(\mu), & \text{si } \Re(\mu) > 0 \end{cases} \quad (3.7)$$

Ou  $\mu \in \mathbb{C}$ ,  $n$  un entier positif et  $f$  une fonction localement intégrable définie sur  $[t_0, \infty]$ .

### Définition de Caputo

A la fin des années 60, dans le cadre de ses travaux sur la dissipation dans un matériau viscoélastique linéaire, Caputo a introduit une autre définition de la dérivation d'ordre fractionnaire [17]. L'expression mathématique de cette définition est :

$${}_a D_t^\mu f(t) = \frac{1}{\Gamma(r-\mu)} \int_a^t (t-\tau)^{r-\mu-1} f^{(r)}(\tau) d\tau \quad (3.8)$$

Avec  $r$  est un entier positif vérifiant l'inégalité  $(1-r) < \mu < r$ .  $f^{(r)}(\tau)$ , étant la dérivée d'ordre entier  $r$ , par rapport à  $\tau$ , de la fonction  $f(\tau)$ .  ${}_a D_t^\mu f(t)$  désigne la dérivée d'ordre fractionnaire  $\mu$  de la fonction  $f(t)$  entre  $a$  et  $t$  selon la définition de Caputo.

### 3.2.2 Propriétés des opérateurs d'ordre fractionnaire

Les principales propriétés des dérivées et intégrales d'ordre fractionnaire sont les suivantes :

- Deux paramètres, " $\mu$ " et " $a$ " sont nécessaires pour définir un opérateur d'ordre fractionnaire.
- La définition de la dérivation d'ordre fractionnaire étant basée sur celle d'une intégration d'ordre fractionnaire, une dérivation d'ordre fractionnaire revêt un caractère global contrairement à une dérivation entière. Il s'avère en effet que la dérivée d'ordre fractionnaire d'une fonction  $f$  nécessite la connaissance de  $f(t)$  sur l'intervalle  $[a, t]$ , alors que dans le cas entier, seule la connaissance locale de  $f$  autour de  $t$  est nécessaire. Cette propriété permet d'interpréter les systèmes d'ordre fractionnaire comme des systèmes à mémoire longue, les systèmes entiers étant alors interprétables comme des systèmes à mémoire courte.
- La définition de Grünwald Letnikov décrit l'unification des deux notions, l'intégral et la dérivée d'ordre fractionnaire.
- La dérivée d'ordre fractionnaire de l'intégrale du même ordre d'une fonction temporelle  $f(t)$  donne :  ${}_a D_t^\mu \circ {}_a I_t^\mu f(t) = f(t)$  avec  $\Re(\mu) > 0$ . Cette relation n'étant pas toujours vraie pour  $\Re(\mu) < 0$ .
- Les opérateurs d'intégration d'ordre fractionnaire vérifient la propriété de semi-groupe, soit :

$${}_a I_t^{n_1} \circ {}_a I_t^{n_2} f(t) = {}_a I_t^{n_1+n_2} f(t) \quad \text{avec :} \quad \begin{cases} \Re(n_1) > 0 \\ \Re(n_2) > 0 \end{cases} \quad (3.9)$$

- Contrairement aux opérateurs d'intégration d'ordre fractionnaire, les opérateurs de dérivation d'ordre fractionnaire (réel ou complexe), ne vérifient la propriété de semi-groupe que sous certaines conditions, soit :

$${}_a D_t^r ({}_a D_t^n f(t)) = {}_a D_t^{r+n} f(t) \quad \text{avec :} \quad \begin{cases} r & \text{entier positif} \\ n & \text{arbitraire} \end{cases} \quad (3.10)$$

- L'opérateur de dérivation d'ordre fractionnaire (réel ou complexe) est linéaire. Ainsi, si  $y$  et  $g$  sont deux fonctions continues et  $(\lambda, \mu)$  réelles, on aura :  $D^{(\mu)}(\lambda.y + \mu.g) = \lambda.D^{(\mu)}(y) + \mu.D^{(\mu)}(g)$
- Pour  $\mu = 0$ ,  $D^{(\mu)}y(t)$  est l'opérateur identité  $D^{(0)}y(t) = y(t)$

### 3.2.3 Transformée de Laplace des opérateurs fractionnaires

Les définitions temporelles des opérateurs différentiels non entiers souffrent d'un formalisme mathématique quelque peu compliqué, mais leur expression dans le domaine de Laplace relève d'une simplicité remarquable [18].

#### Transformée de Laplace de l'intégrale d'ordre fractionnaire

Nous commencerons par la transformée de Laplace de l'intégrale d'ordre fractionnaire de Riemann-Liouville d'ordre  $\mu > 0$  qu'on peut l'écrire comme une convolution des fonctions  $g(t) = \frac{1}{\Gamma(\mu)} \cdot t^{\mu-1}$  et  $f(t)$ .

$$I^\mu f(t) = D^{-\mu} = \frac{1}{\Gamma(\mu)} \int_0^t (t-\tau)^{\mu-1} f(\tau) d\tau = t^{\mu-1} \cdot f(t) \quad (3.11)$$

La transformée de Laplace de la fonction  $t^\mu$  est :

$$G(s) = L\{t^{\mu-1}; s\} = \Gamma(\mu) \cdot s^{-\mu} \quad (3.12)$$

En utilisant la formule de la transformée de Laplace de la convolution :

$$L\{f(t) \cdot d(t); s\} = F(s) \cdot G(s) \quad (3.13)$$

On obtient la transformée de Laplace de l'intégrale de Riemann-Liouville, de Grünwald-Letnikov et de Caputo :

$$L\{I^\mu f(t)\} = s^{-\mu} \cdot F(s) \quad (3.14)$$

#### Transformée de Laplace de la dérivée d'ordre fractionnaire

Nous citons dans ce qui suit les transformées de Laplace des différentes définitions de la dérivée.

- Dérivée de Riemann Liouville

$$L\{D^\mu f(t)\} = s^{-\mu} F(s) - \sum s^k [D^{\mu-k-1} f(t)]_{t=0} \quad (3.15)$$

Avec  $(n - 1) < \mu < n$ . Cette transformée de Laplace de la dérivée de Riemann-Liouville est bien connue. Mais son applicabilité en pratique est limitée à cause de l'absence d'interprétation physique des valeurs limites des dérivées d'ordre fractionnaire pour  $t = 0$ .

- Dérivée de Grünwald- Letnikov

$$L\{D^\mu f(t)\} = s^{-\mu} F(s) \quad (3.16)$$

- Dérivée de Caputo

$$L\{D^\mu f(t)\} = s^{-\mu} F(s) - \sum_{k=0}^{n-1} s^{\mu-k-1} f^k(0) \quad (3.17)$$

### 3.2.4 Approximation des opérateurs d'ordre fractionnaire

D'habitude les simulations sont effectuées avec un logiciel préparé pour traiter seulement les puissances d'ordre entier de  $s$ . Alors il est très important de trouver des approximations d'ordre entier pour des fonctions de transfert d'ordre fractionnaire. Autrement dit, lorsque les simulations doivent être exécutées ou les correcteurs doivent être implémentés, les fonctions de transfert d'ordre fractionnaire sont remplacées par des fonctions de transfert d'ordre entier, avec un comportement assez identique à celles désirées, mais beaucoup plus facile à manipuler. Il existe différentes méthodes pour trouver de telles approximations, les approximations disponibles dans le domaine  $s$  sont appelées des approximations analogiques ou des approximations du domaine fréquentiel. Dans la littérature il existe plusieurs méthodes d'approximation analogique des systèmes d'ordre fractionnaire, et les plus utilisées sont :

- Méthode EFC (Expansion Fractionnaire Continue)
- Méthode de Charef
- Méthode d'Oustaloup
- Méthode de Carlson

La méthode d'Oustaloup est plus élaborée et plus pratique pour les approximations des fonctions de transferts d'ordre fractionnaire, c'est pourquoi on a choisi de la présenter en détail dans la suite.

#### Approximation d'Oustaloup

Approximation du filtre d'Oustaloup à un différenciateur d'ordre fractionnaire  $G(s) = s^\mu$  est largement utilisée dans les applications. Le filtre Oustaloup peut être conçu :

$$G_t(s) = K \prod_{i=1}^N \left\{ \frac{s + w'_i}{s + w_i} \right\} \quad (3.18)$$

Où :

$$w'_i = w_b w_u^{(2i-1-\mu)/N} K = w_h^\mu w_u = \sqrt{\frac{w_h}{w_b}} \quad (3.19)$$

L'approximation est conçue pour les fréquences  $w \in [w_b, w_h]$  et  $N$  est l'ordre de l'approximation. Comme on peut le voir son représentation prend la forme d'un produit d'une série des systèmes stables linéaires de premier ordre. Comme on peut l'observer, le choix d'une large bande des résultats d'approximation et de grands  $w_u$  et un ordre  $N$  élevé donne un espacement des pôles de près de  $w_h$  à ceux très proches de  $w_b$ . Cet espacement est logarithmique avec un groupement proche de  $-w_b$  et pose des problèmes de discrétisation. Le choix d'une large bande d'approximation est d'autre part souhaitable, parce que l'approximation se comporte le mieux à l'intérieur de l'intervalle et pas à sa limite, donc certaines marges doivent être maintenues [19].

### 3.3 Régulateur PID

Ce régulateur relie directement le signal de commande  $u(t)$  au signal d'écart  $e(t)$ , et remplit essentiellement trois fonctions :

- ✓ il fournit un signal de commande en tenant compte de l'évolution de signal de sortie par rapport à la consigne.
- ✓ il élimine l'erreur statique grâce au terme intégrateur.
- ✓ il anticipe les variations de la sortie grâce au terme dérivateur.

Sa description temporelle est la suivante :

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3.20)$$

Avec :  $e(t) = y_r(t) - y(t)$  est l'erreur à l'instant  $t$  entre le signal référence  $y_r$  et le signal actuel  $y$

Sa fonction de transfert s'écrit :

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (3.21)$$

Dans notre cas le système à commander est le robot Delta ISIR88, tel que l'entrée de commande est le couple  $\tau$  et on a comme sortie la position articulaire  $\alpha$ . La structure de commande est représentée par la figure (3.1) :

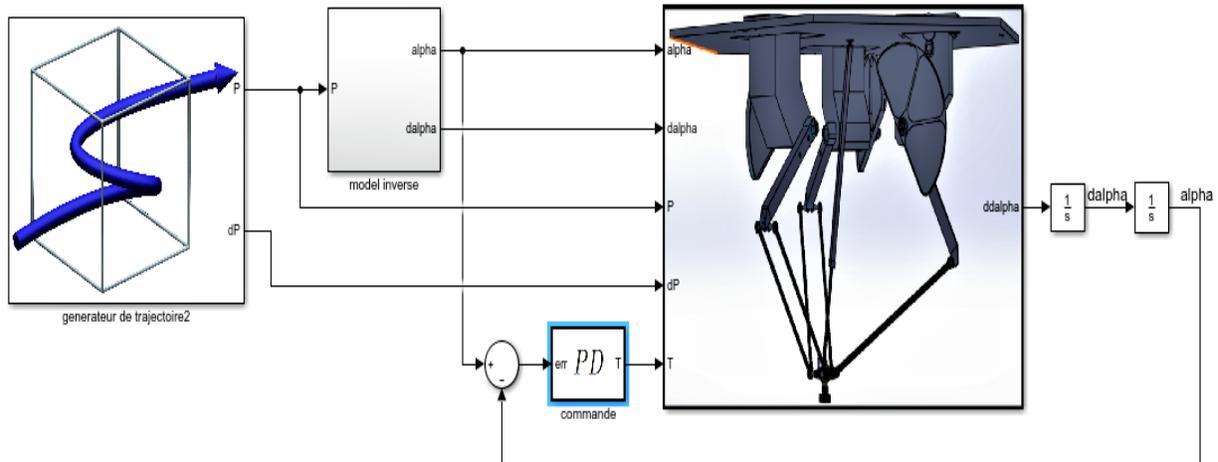


FIGURE 3.1: Structure de commande PD sur le robot ISIR88

Cette commande est implémentée par défaut dans notre robot Delta ISIR88 par le constructeur (EPFL), avec comme paramètres :  $k_p = 2.2$  et  $k_d = 0.0145$ . Donc on peut décrire la commande de ce régulateur comme suit :

$$U_i = k_p(\alpha_{d_i} - \alpha_i) + k_d(\dot{\alpha}_{d_i} - \dot{\alpha}_i) \quad i = 1, 2, 3 \quad (3.22)$$

Avec :

- $\alpha_i(\dot{\alpha}_i)$  : position (vitesse) articulaires courantes de l'articulation  $i$ .
- $\alpha_{d_i}(\dot{\alpha}_{d_i})$  : position (vitesse) articulaire désirée suivant l'articulation  $i$ .
- $U_i$  : le couple ( $T$ ) appliqué sur le bras  $i$ .
- $i$  : numéro de l'articulation.

### 3.4 Régulateur PD d'ordre fractionnaire

L'amélioration et l'optimisation des résultats d'une commande sont toujours demandées dans le domaine d'industrie tout en gardant la simplicité d'algorithmes de contrôle, c'est pour cela plusieurs travaux ont été fait pour améliorer les résultats du PID (pour sa simplicité et facilité d'implémentation). Chaque action du régulateur PID a des avantages et des inconvénients, par exemple l'action dérivée augmente la stabilité relative en introduisant un déphasage de  $\pi/2$  dans le domaine fréquentiel mais augmente aussi la sensitive du système aux bruits en ajoutant une pente de  $20dB/dec$  en fréquentiel. Il est donc évident qu'on peut atteindre un compromis entre les avantages et les inconvénients. Pour cela, Poudlubny a proposé une généralisation du régulateur  $PID$  classique à la forme  $PI^\lambda D^\mu$  nommé la  $PID$  d'ordre fractionnaire, où  $\lambda$  et  $\mu$  sont des réels positifs tel que  $0 < \lambda < 2$  et  $0 < \mu < 2$ .

De point de vue graphique, la figure (3.2) montre les possibilités de contrôle à l'aide d'un régulateur d'ordre fractionnaire, en étendant les quatre points de contrôle du  $PID$  au

quart de plan défini en sélectionnant les valeurs de  $\lambda$  et de  $\mu$ . Cette extension donne plus de flexibilité dans la conception des commandes [17].

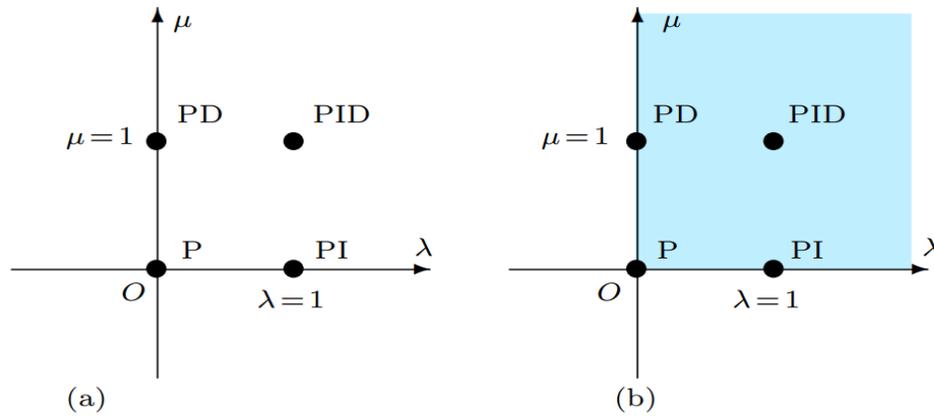


FIGURE 3.2: PID d'ordre fractionnel vs PID classique : de points à plan : (a) ordre entier et (b) ordre fractionnaire

Pour mieux visualiser cette effet on donne le tracé de Bode de deux correcteurs *PID*, le premier avec ( $K_p = K_d = 1, K_i = 3$ ) et un deuxième avec ( $K_p = K_d = K_i = 1$ ).

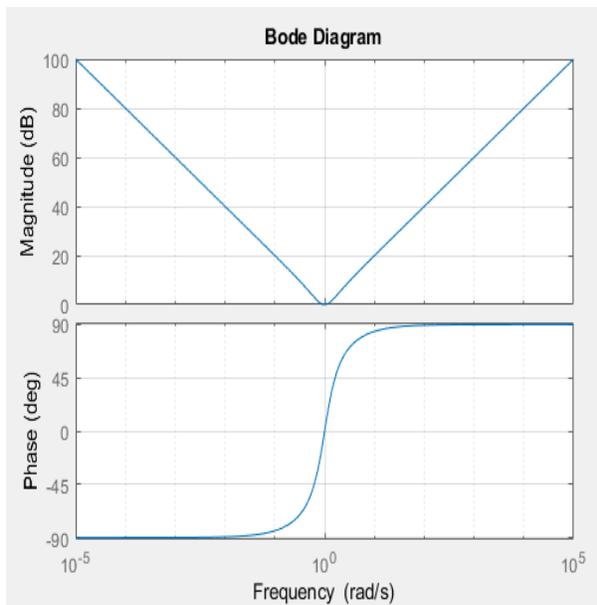


FIGURE 3.3: Réponse fréquentielle d'un régulateur PID classique avec  $k_p = k_i = k_d = 1$

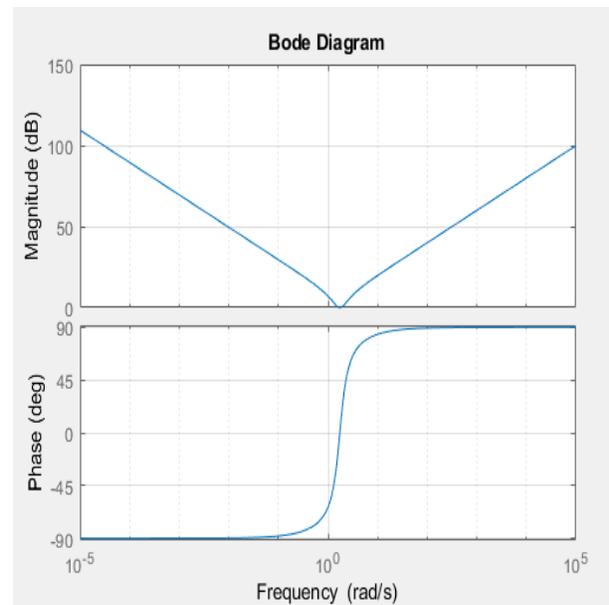


FIGURE 3.4: Réponse fréquentielle d'un contrôleur PID classique avec  $k_p = k_d = 1, k_i = 3$

On remarque d'après ces deux figures que la valeur et la position de point minimale du gain et de point d'inflexion ont changé, mais les pentes de gain et les bornes de la phase sont restées les mêmes [17].

On introduit maintenant le régulateur PID d'ordre fractionnaire qui a comme loi de commande :

$$u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^\mu e(t) \quad \text{avec : } \lambda, \mu \in \mathbb{R}^2 \quad (3.23)$$

En appliquant la transformée de Laplace à l'équation (3.23) avec des conditions initiales nulles, la fonction de transfert du contrôleur est donnée par :

$$C_f(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu = k \frac{(s/w_f)^{\lambda+\mu} + s \delta_f s^\lambda / w_f + 1}{s^\lambda} \quad (3.24)$$

La méthode d'Oustaloup est utilisée pour tracer le diagramme de Bode de la fonction de transfert du *PID* d'ordre fractionnaire ( la méthode d'Oustaloup est toujours utilisée pour avoir la réponse d'un système fractionnaire dans le domaine fréquentiel). Les résultats sont présentés dans la figure (3.5) :

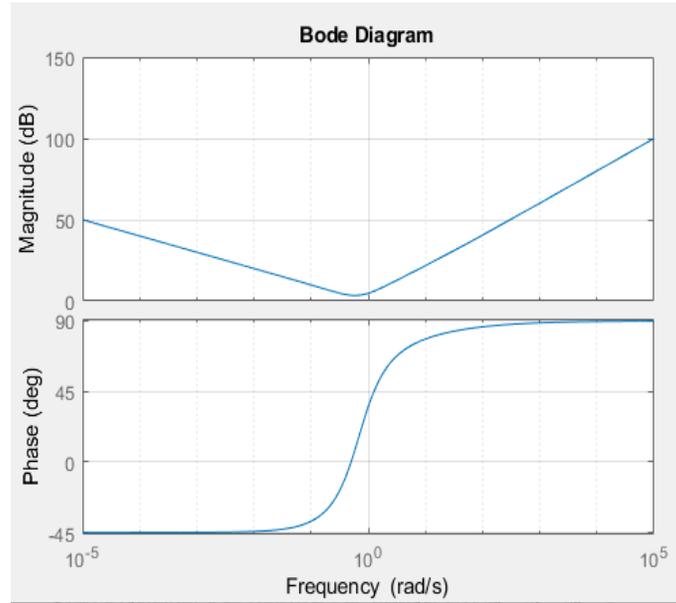


FIGURE 3.5: Réponse fréquentielle d'un contrôleur PID fractionnaire avec  $k_p = k_d = 1, k_i = 1, \mu = \lambda = 0.5$

On peut remarquer que le régulateur d'ordre fractionnaire nous permet de contrôler au même temps la pente du gain et les asymptotes de la phase dans les hautes et basses fréquences. Par conséquent, le régulateur *PID* d'ordre fractionnaire est plus flexible et plus adapté pour trouver une meilleure réponse.

Dans notre cas, la méthode de Grünwald-Letnikov est utilisée pour le calcul fractionnaire avec l'approche numérique basée sur le développement de Taylor [20] suivant :

$${}_{(k-L_m/h)}D_{t_k}^q f(t) \approx h^{-q} \sum_{j=0}^k (-1)^j \binom{q}{j} f(t_{k-j}) = h^{-q} \sum_{j=0}^k c_j^{(q)} f(t_{k-j}) \quad (3.25)$$

Avec  $L_m$  est représentée la longueur de notre mémoire,  $t_k = Kh$  tel que  $h$  est la période d'échantillonnage, et  $c_j^{(q)}$  ( $j = 0, 1, \dots, k$ ) sont des coefficients binomiales; qui peuvent être calculés avec l'expression (3.26) :

$$c_0^{(q)} = 1 \quad , \quad c_j^{(q)} = \left(1 - \frac{1+q}{j}\right) c_{j-1}^{(q)} \quad (3.26)$$

Finalement, la structure de commande PD d'ordre fractionnaire appliquée sur le robot ISIR88 est représentée par la figure (3.6) :

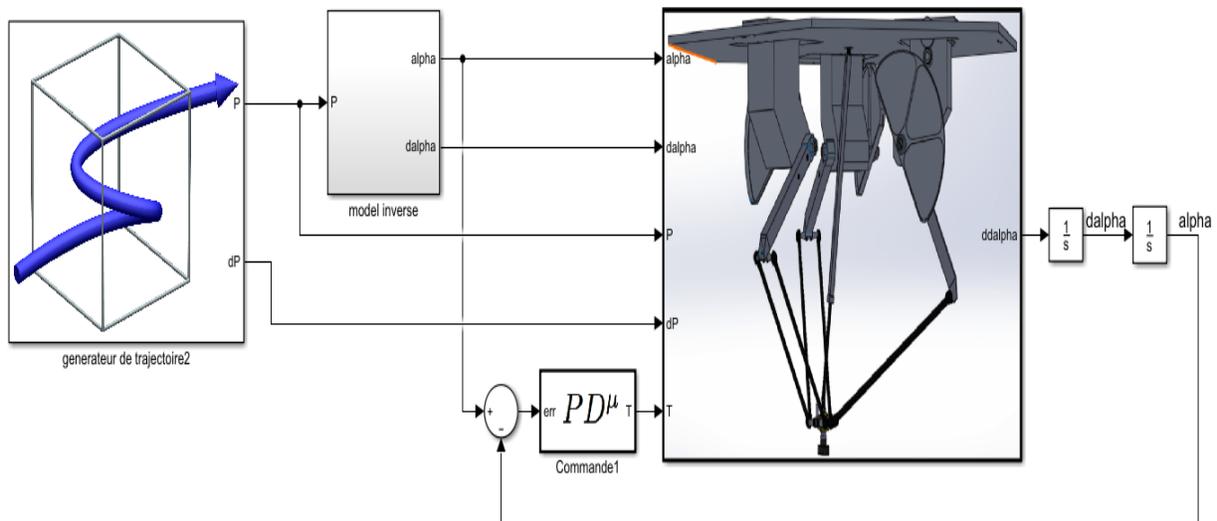


FIGURE 3.6: Structure d'un régulateur PD d'ordre fractionnaire

### 3.4.1 Ajustement des paramètres du contrôleur

L'algorithme principal pour trouver les paramètres du contrôleur PD d'ordre fractionnaire est représentée par la figure (3.7). Concernant l'optimisation, on a utilisé la toolbox Simulink "Simulink Design Optimization" qui utilise les mêmes fonctions de la toolbox Matlab "FMINCON". Cette toolbox permet l'optimisation de plusieurs paramètres dans un schéma-bloc Simulink simultanément, elle offre le choix de la fonction objective à optimiser et le choix d'algorithmes d'optimisation (la descente de gradient, pattern search ...etc.) [21]. Dans notre cas on a utilisé l'algorithme de recherche de motifs "Pattern Search", cet algorithme est une famille de méthodes d'optimisation numériques qui ne nécessitent pas le calcul du gradient, et donc il peut être utilisé sur les fonctions qui ne sont pas continues ou différentiables.

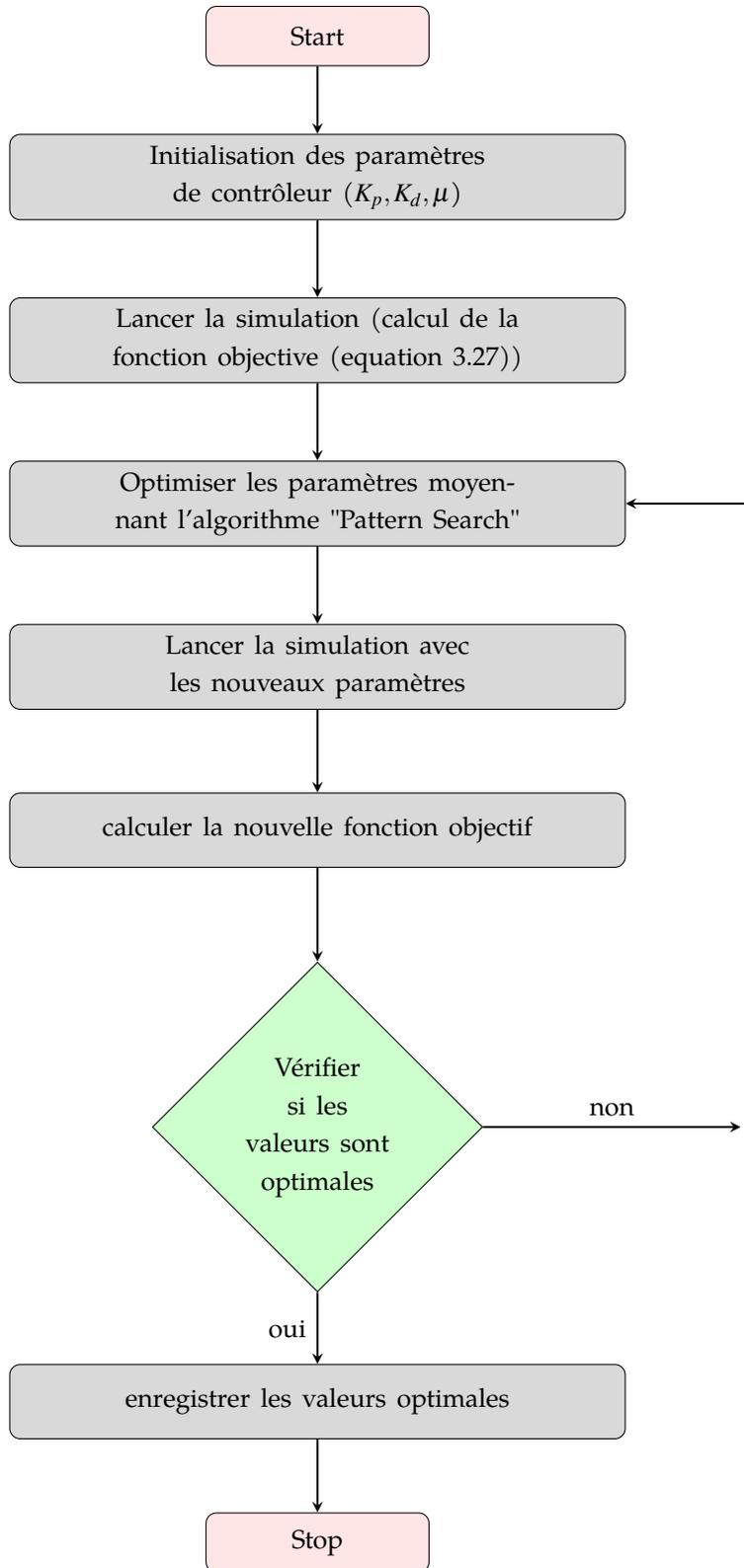


FIGURE 3.7: Principe de fonctionnement de la toolbox "Simulink Design Optimisation"

### L'algorithme de recherche de motifs

L'algorithme de recherche de motifs trouve une séquence de points,  $x_0, x_1, x_2, \dots$  qui s'approchent d'un point optimal. La valeur de la fonction objective diminue ou reste la même de chaque point de la séquence au suivant.

L'algorithme commence par définir des vecteurs de motif avec la méthode d'hypercube latin qui est une méthode d'échantillonnage statistique permettant de générer un échantillon quasi aléatoire de valeurs de paramètres à partir d'une distribution multidimensionnelle, ces vecteurs sont ajoutés au point de départ. L'algorithme sonde les points de maillage obtenus en calculant leurs valeurs de fonction objective jusqu'à ce qu'il trouve celle dont la valeur est inférieure à la fonction objective du point de départ, ici le sondage à l'itération 1 est réussi. L'algorithme ensuite définit le point de départ pour l'itération suivante comme ce point optimal obtenu [22]. On itère l'algorithme jusqu'à avoir la différence entre deux fonctions objectives successives inférieures à une erreur relative prédéfinie ( $10^{-6}$  dans notre cas).

La fonction objective utilisée dans l'optimisation est la suivante :

$$\text{Min}f(x) = \sum_{i=1}^3 \left( \sum_{k=0}^{n-1} (Ae_i^2(t = kh) + B\tau_i^2(t = kh)) \right) \quad (3.27)$$

Avec :

$e$  : Erreur de position articulaire (en rad).

$\tau$  : Le couple appliqué sur le bras.

$H$  : La période d'échantillonnage.

$n$  : Nombre de pas de la simulation ( $= tf/h$ ).

$A (B)$  : Pondération de l'erreur (couple) (dans notre cas  $A = 100, B = 1$ )

## 3.5 Commande PD d'ordre fractionnaire par la méthode de couple calculé

Les régulateurs PD et PD d'ordre fractionnaire utilisent partiellement le modèle dynamique du système et ils ne tiennent pas compte de la présence des perturbations externes et d'incertitude paramétrique, qui affectent le robot dans les tâches de poursuite. Des techniques de contrôle robustes comme la théorie de la rétroaction quantitative ("Quantitative Feedback theory QFT") ou H infini sont utilisés pour des systèmes similaires. Toutefois, la conception et la mise en œuvre de ces techniques sont complexes. Pour cela on utilise la méthode de couple calculé ("Computed Torque Control") qui utilise toute la dynamique, et donc, prend compte des perturbations...etc. En outre, l'utilisation des opérateurs d'ordre fractionnaire permet d'améliorer la performance et la robustesse du système.

On reprend le modèle dynamique inverse du robot donnée par l'équation (2.44) :

$$\tau = M(\alpha)\ddot{\alpha} + C(\alpha, \dot{\alpha}) + g(\alpha) \quad (3.28)$$

L'application d'un régulateur *PD* d'ordre fractionnaire avec la méthode de couple calculé sur notre robot nous donne la loi de commande suivante :

$$\begin{cases} \tau = M(\alpha)u + C(\alpha, \dot{\alpha}) + G(\alpha) \\ u = \ddot{\alpha}_d + k_p e(t) + k_d D^\mu e(t) \end{cases} \quad (3.29)$$

Avec  $\ddot{\alpha}_d$  est l'accélération d'articulation désirée,  $e(t)$  est l'erreur de position articulaire,  $\mu$  est l'ordre fractionnaire de dérivation,  $k_p$  et  $k_d$  sont deux matrices diagonales. La structure de commande est représentée par la figure (3.8).

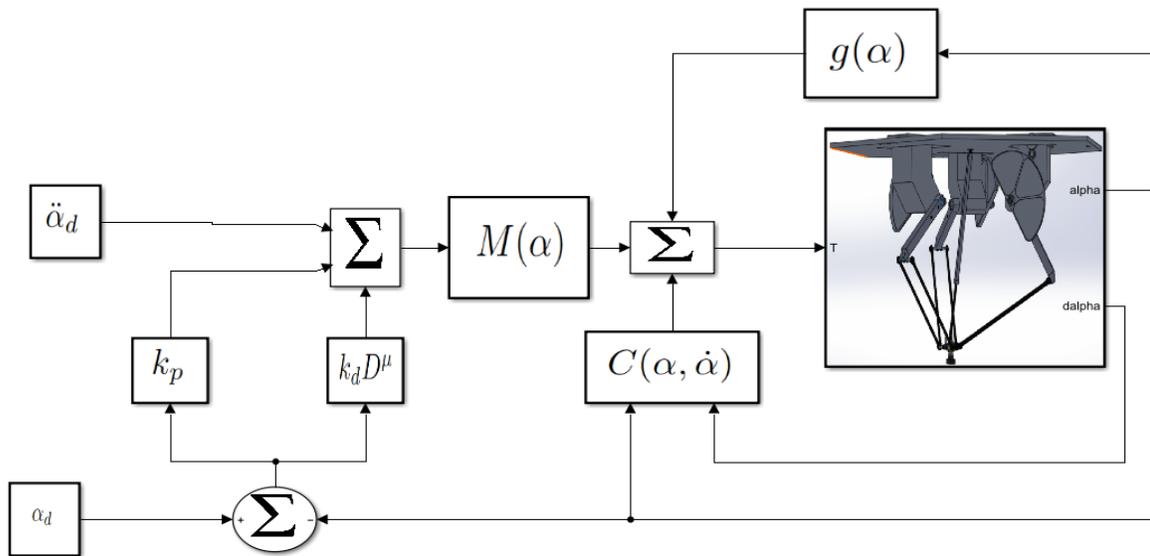


FIGURE 3.8: Structure du commande de couple calculée avec PD d'ordre fractionnaire

### 3.6 Simulation

Dans cette section nous présenterons les résultats de la simulation des trois régulateurs sur le robot ISIR88 à l'aide de "Matlab SIMULINK", tel que les paramètres de chacun sont donnés comme suit :

- **Régulateur PD :**  
 $k_p = 2.29765625$ ,  $k_d = 0.012546875$ .
- **Régulateur PD d'ordre fractionnaire :**  
 $k_p = 2.29765625$ ,  $k_d = 0.012546875$  et  $\mu = 1.11767578125$ .
- **Régulateur PD d'ordre fractionnaire avec CTC :**  
 $\mu = 1.2087$ ;  $k_p = \text{diag}(218.04, 218.04, 218.04)$ ;  $k_d = \text{diag}(0.672, 0.672, 0.672)$ .

### 3.6.1 Test de poursuite

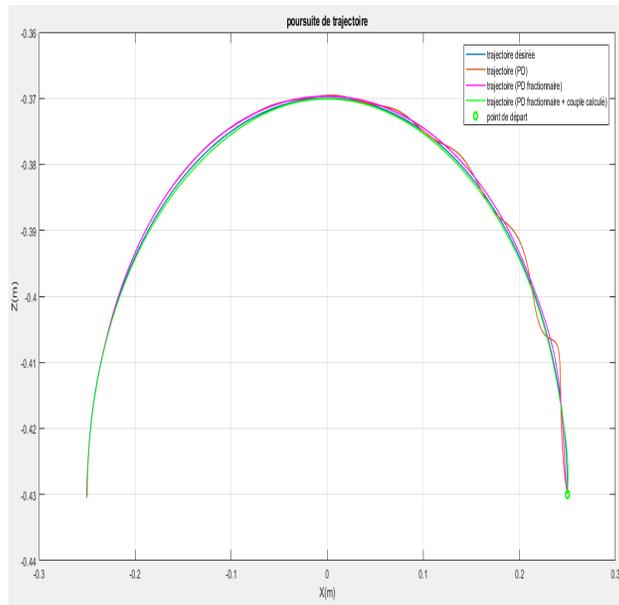


FIGURE 3.9: Résultats obtenus par la simulation des trois commandes (suivie de trajectoire)

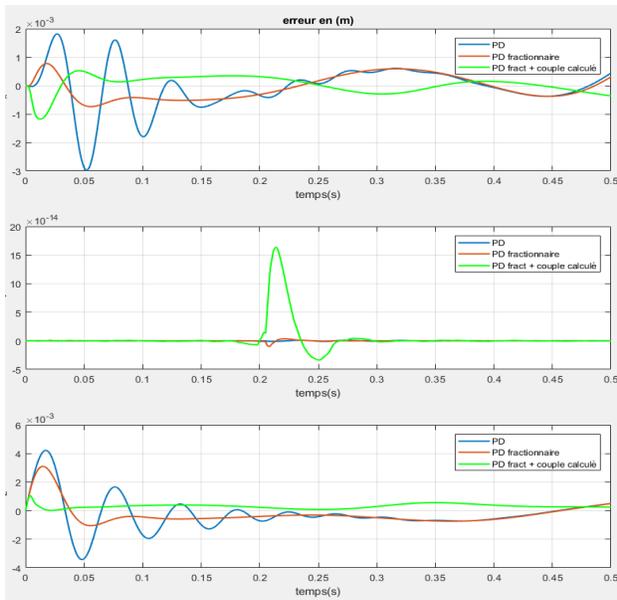


FIGURE 3.10: Les erreurs de position opérationnelle obtenus par les trois régulateurs

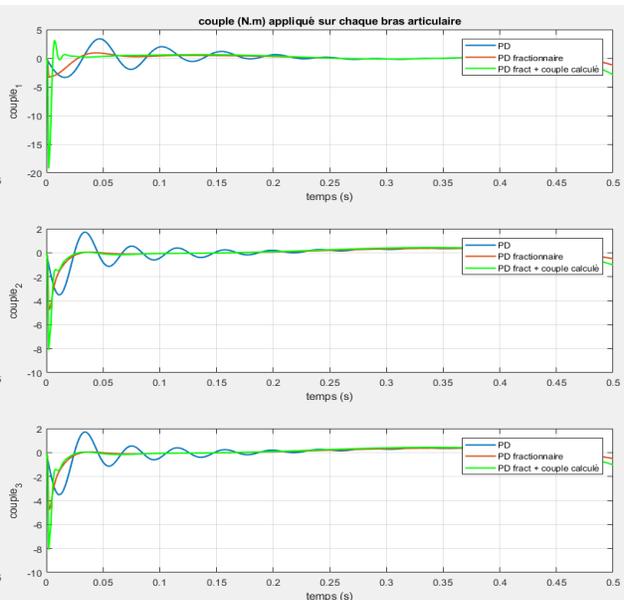


FIGURE 3.11: Les couples obtenus par la simulation des trois régulateurs

### 3.6.2 Test de robustesse

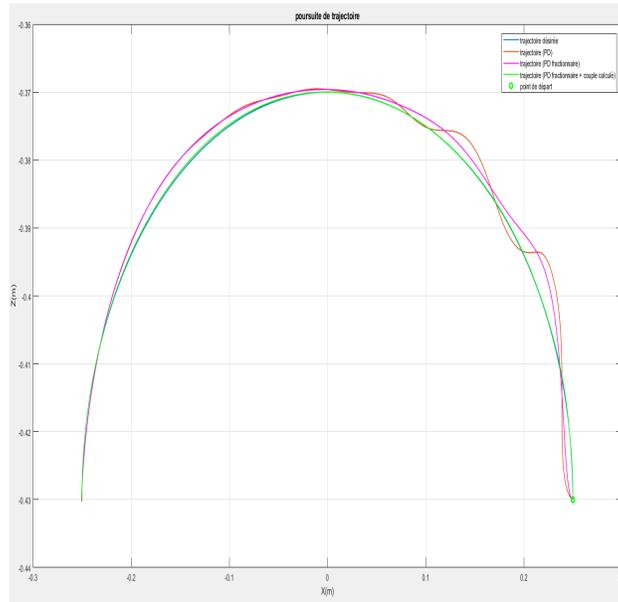


FIGURE 3.12: Résultats obtenus par la simulation des trois commandes (suivie de trajectoire) avec une masse supplémentaire de 200g

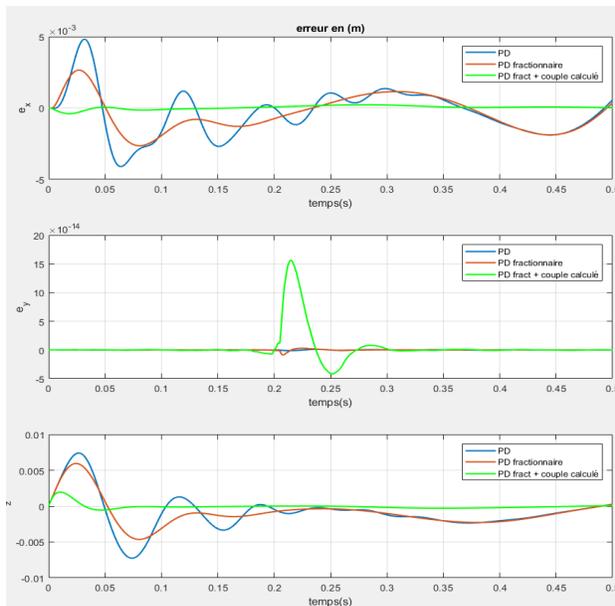


FIGURE 3.13: Les erreurs de position opérationnelle obtenus par les trois régulateurs avec une masse de 200g

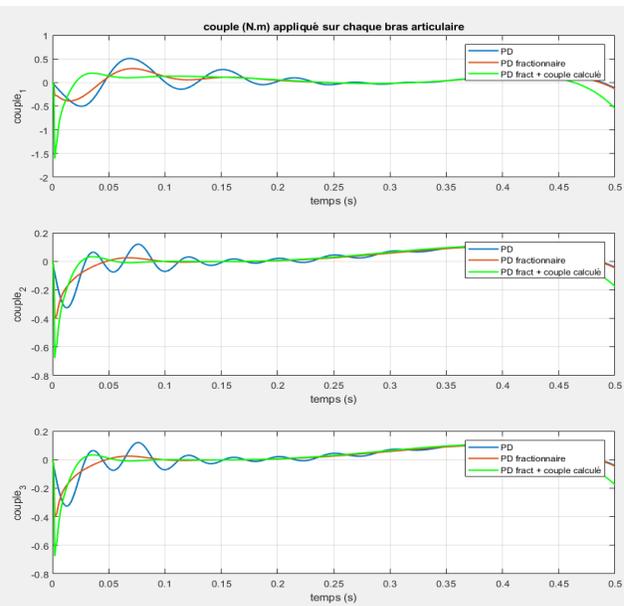


FIGURE 3.14: Les couples obtenus par la simulation des trois régulateurs avec une masse de 200g

Le tracé des résultats dans une seule figure nous a permis de bien visualiser la différence entre les trois régulateurs. Tout d'abord, d'après les résultats de poursuite de

la figure (3.9) on peut constater que l'introduction de terme dérivateur d'ordre fractionnaire dans le régulateur PD a amélioré le comportement de robot, tel que la poursuite obtenue avec le PD d'ordre fractionnaire est meilleure que celle obtenue avec le PD classique; une autre amélioration est atteinte par l'introduction du modèle dynamique moyennant la commande de couple calculé est aussi claire dans la figure (3.9), ce dernier permet au régulateur de se rapprocher rapidement du couple nécessaire. La figure (3.10) renforce les remarques précédentes là qu'on voit les erreurs de la position de la nacelle diminuent une première fois avec le PD d'ordre fractionnaire par rapport à ceux obtenus avec le PD et diminuent pour une deuxième fois avec la CTC. Les couples obtenus avec les deux régulateurs d'ordres fractionnaires ont diminué par rapport à ceux obtenus avec le PD classique figure (3.11) sauf que la commande par couple calculé a provoqué un pic relativement élevé dans le début mais ça s'améliore très rapidement, ce qui veut dire que ces commandes sont optimaux en terme d'énergie.

Après l'ajout d'une masse supplémentaire, on voit que le régulateur avec couple calculé est le meilleur avec une poursuite quasiment parfaite. Le régulateur PD d'ordre fractionnaire reste toujours meilleur que le PD classique, ce résultat qu'on peut tirer facilement de la figure (3.13). En terme de robustesse, on peut dire que le PD d'ordre fractionnaire est plus robuste que le PD classique; mais cette robustesse peut être considérablement améliorée en introduisant le modèle dynamique par la méthode de couple calculé.

## 3.7 Expérimentation

Le robot DELTA ISIR88 du Laboratoire de Commande des Processus "LCP" de l'ENP (figure 3.15) est doté d'un ordinateur spécial ayant un système d'acquisition des données des capteurs et des moteurs facilitant ainsi la tâche de commande et donnant la possibilité d'enregistrement des résultats [23].

Sur le plan software, FlexCom est une plateforme de programmation des lois de commande à implémenter et des trajectoires existent et jouissent d'une simplicité de modification des programmes remarquables [24]. La description des outils de programmation ainsi que les étapes de mise en marche sont données dans [9].

A noter que les différentes lois de commande sont programmées par nous-mêmes en C++.

Pour chacun des régulateurs, à savoir le PD et PD d'ordre fractionnaire, trois tests ont été effectués avec la même trajectoire de la figure (3.16) et les mêmes constantes utilisées dans la simulation; Le générateur de trajectoire a été configuré pour une trajectoire linéaire selon un seul axe, avec un profil trapézoïdal en vitesse (dans l'espace opérationnel), suivie d'une prise dépose entre deux points qui s'effectue quatre fois (pour avoir suffisamment d'information sur le suivie de trajectoire). Un premier test est effectué avec une dynamique faible (une accélération maximale de  $10m/s^2$  et une vitesse maximale de  $5m/s$ ); dans le deuxième test, on augmente la dynamique du système (une accélération maximale de  $15m/s^2$  et une vitesse maximale de  $7.5m/s$ ) et le dernier un test de robustesse qui consiste à tester le robot avec une masse supplémentaire (46g dans notre cas).



FIGURE 3.15: Photographie du banc d'expérimentation avec Robot Delta ISIR88

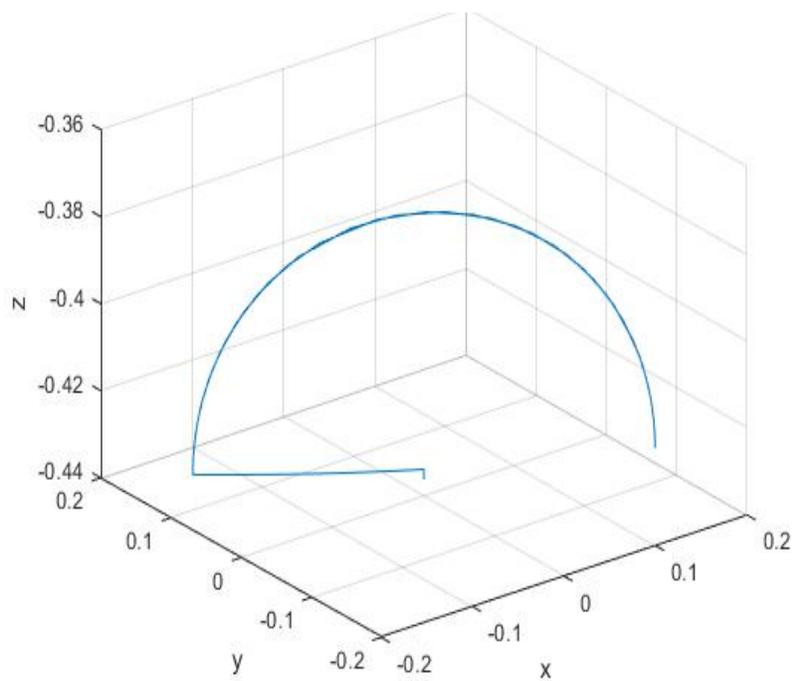


FIGURE 3.16: trajectoire désirée

### 3.7.1 Test avec une dynamique faible

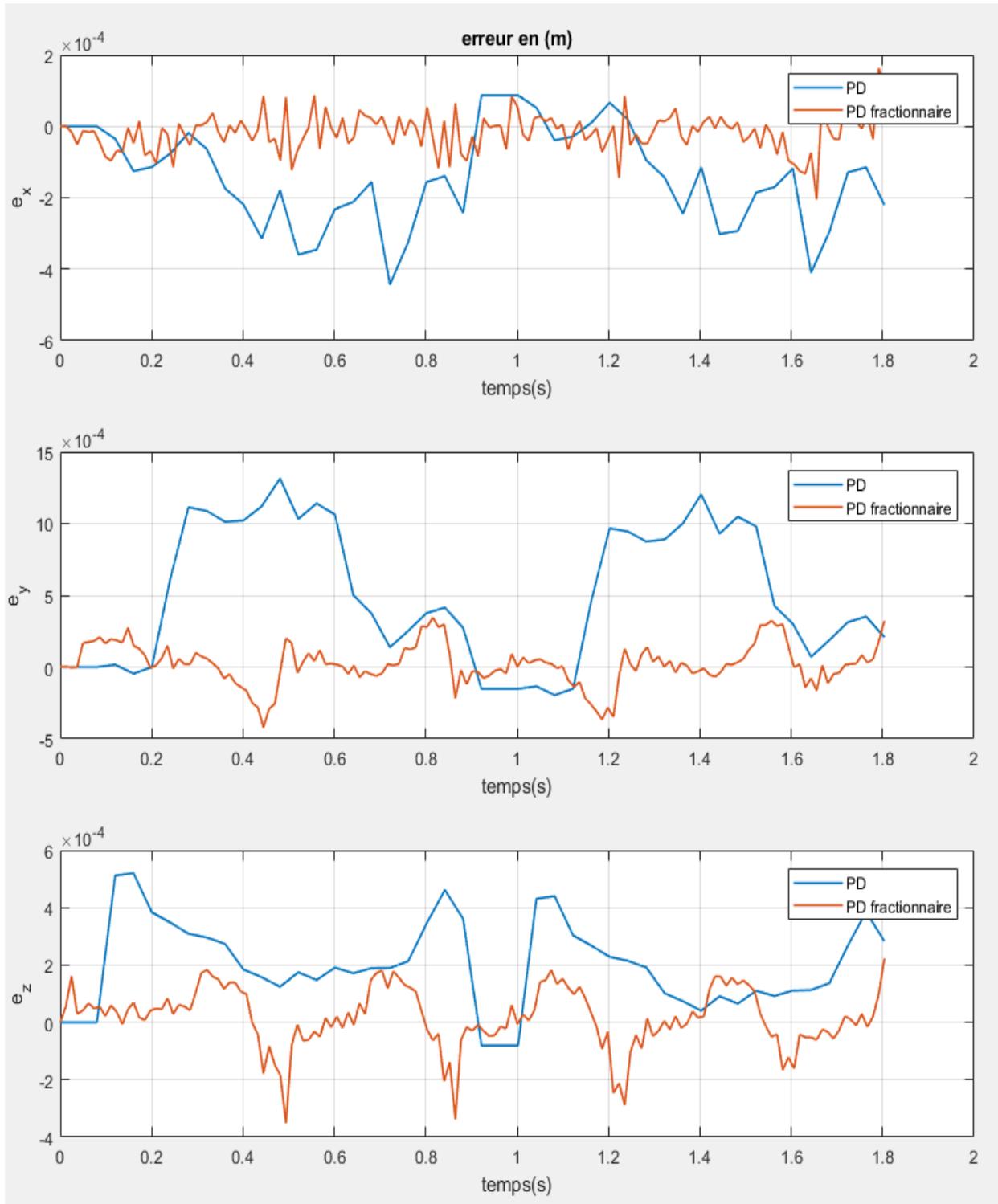


FIGURE 3.17: Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot

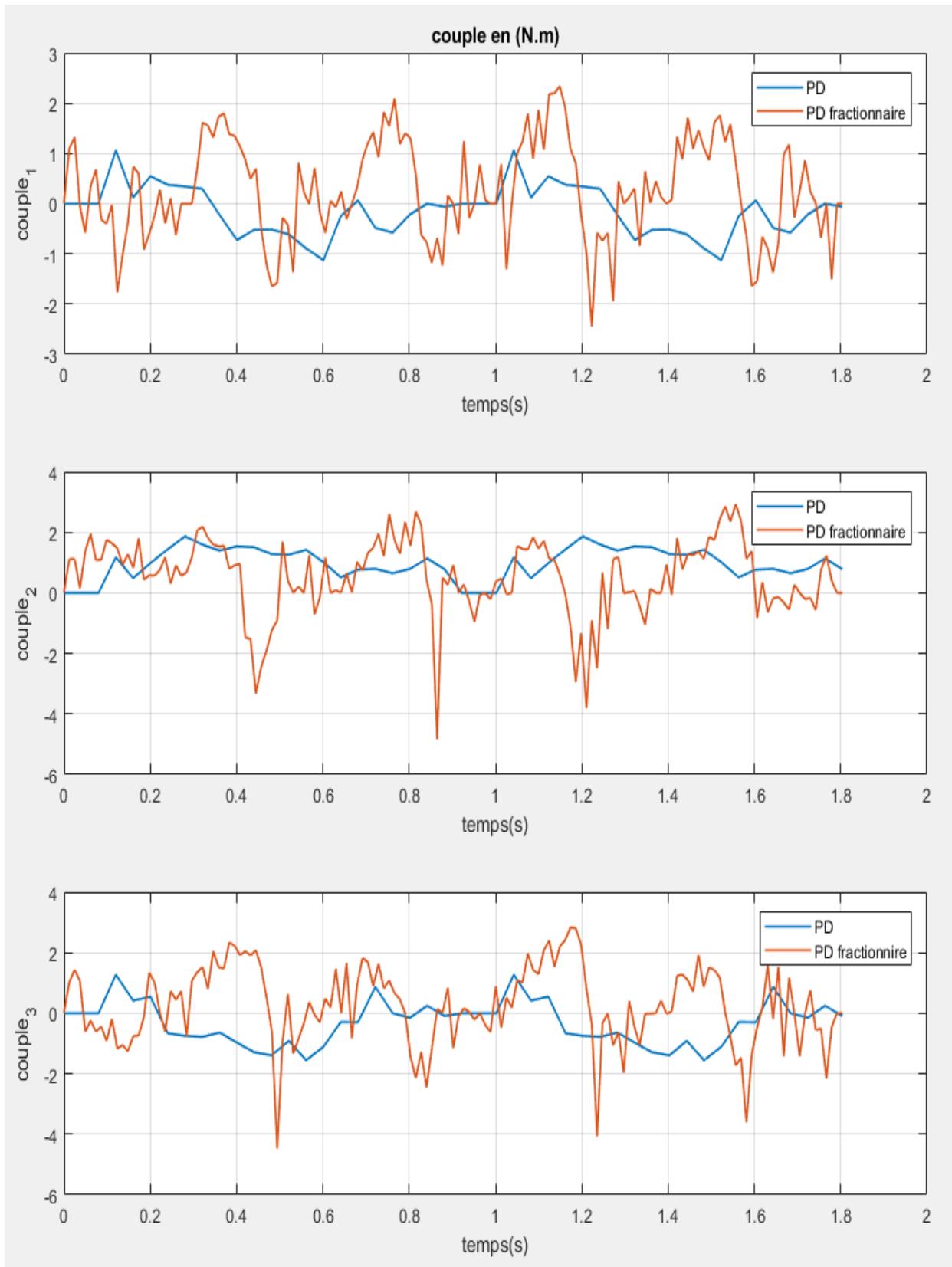


FIGURE 3.18: Les couples obtenus par l'implémentation des trois régulateurs sur le robot

### 3.7.2 Test avec une dynamique plus élevée

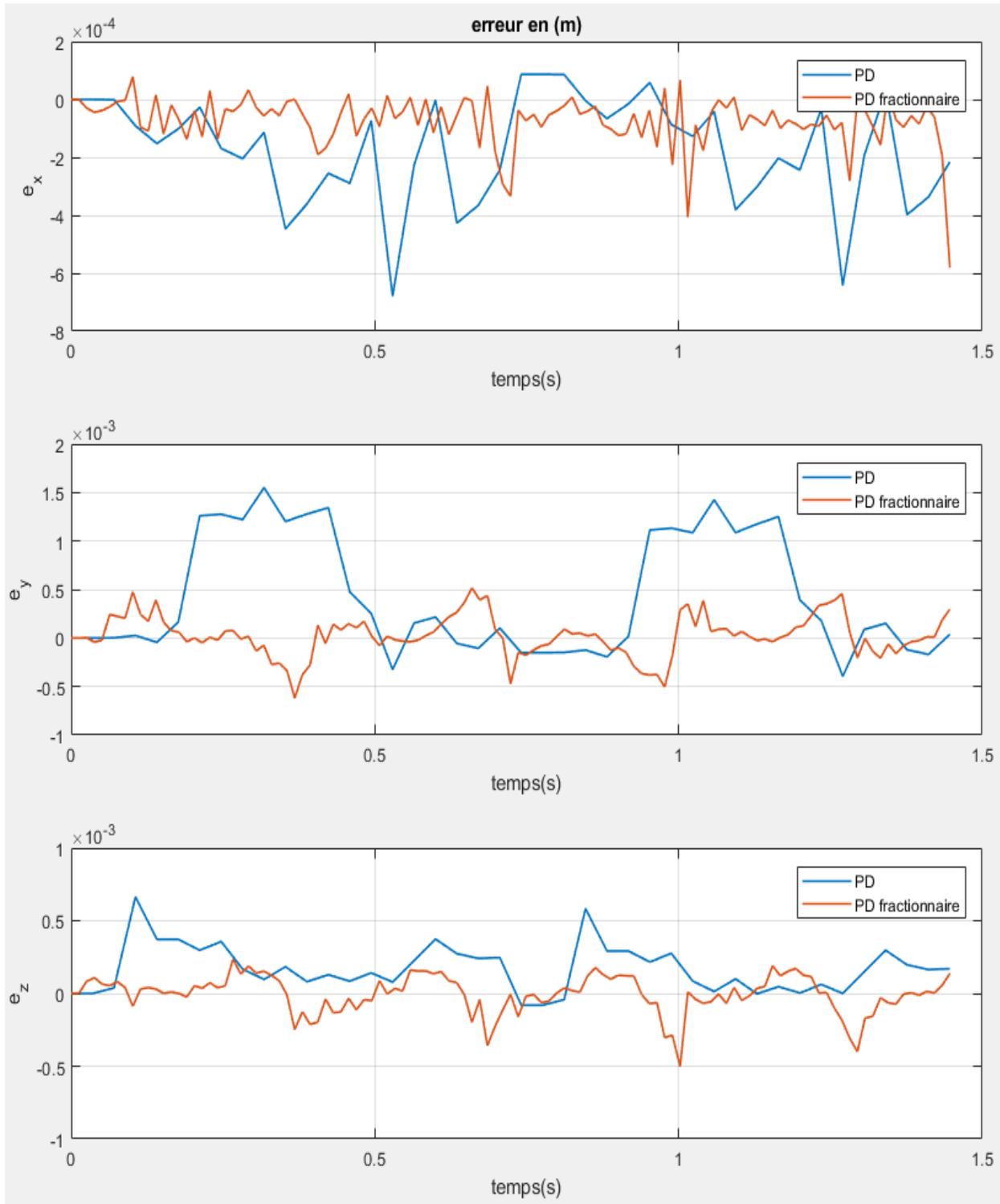


FIGURE 3.19: Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot pour une dynamique rapide

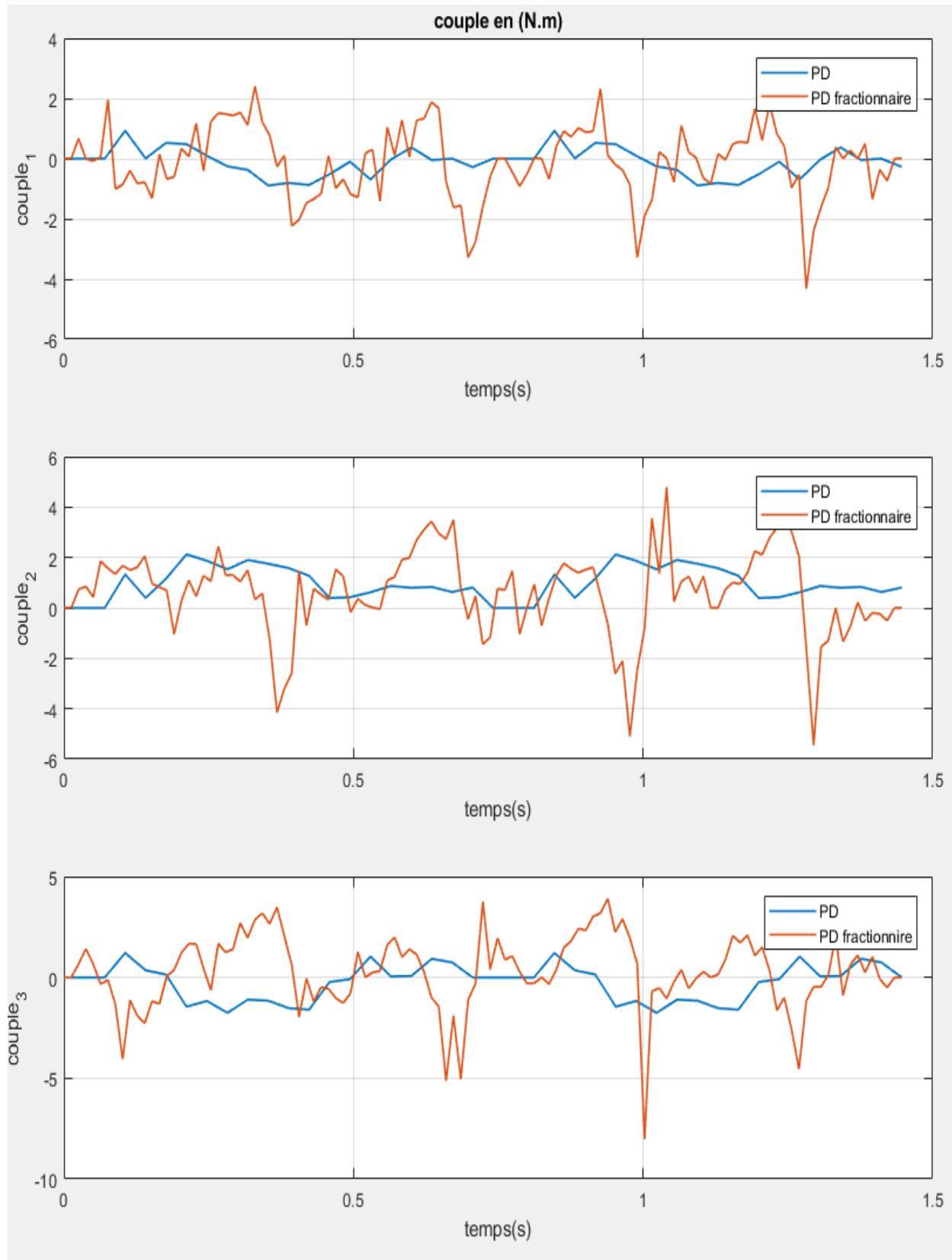


FIGURE 3.20: Les couples obtenus par l'implémentation des trois régulateurs sur le robot pour une dynamique rapide

### 3.7.3 Test de robustesse

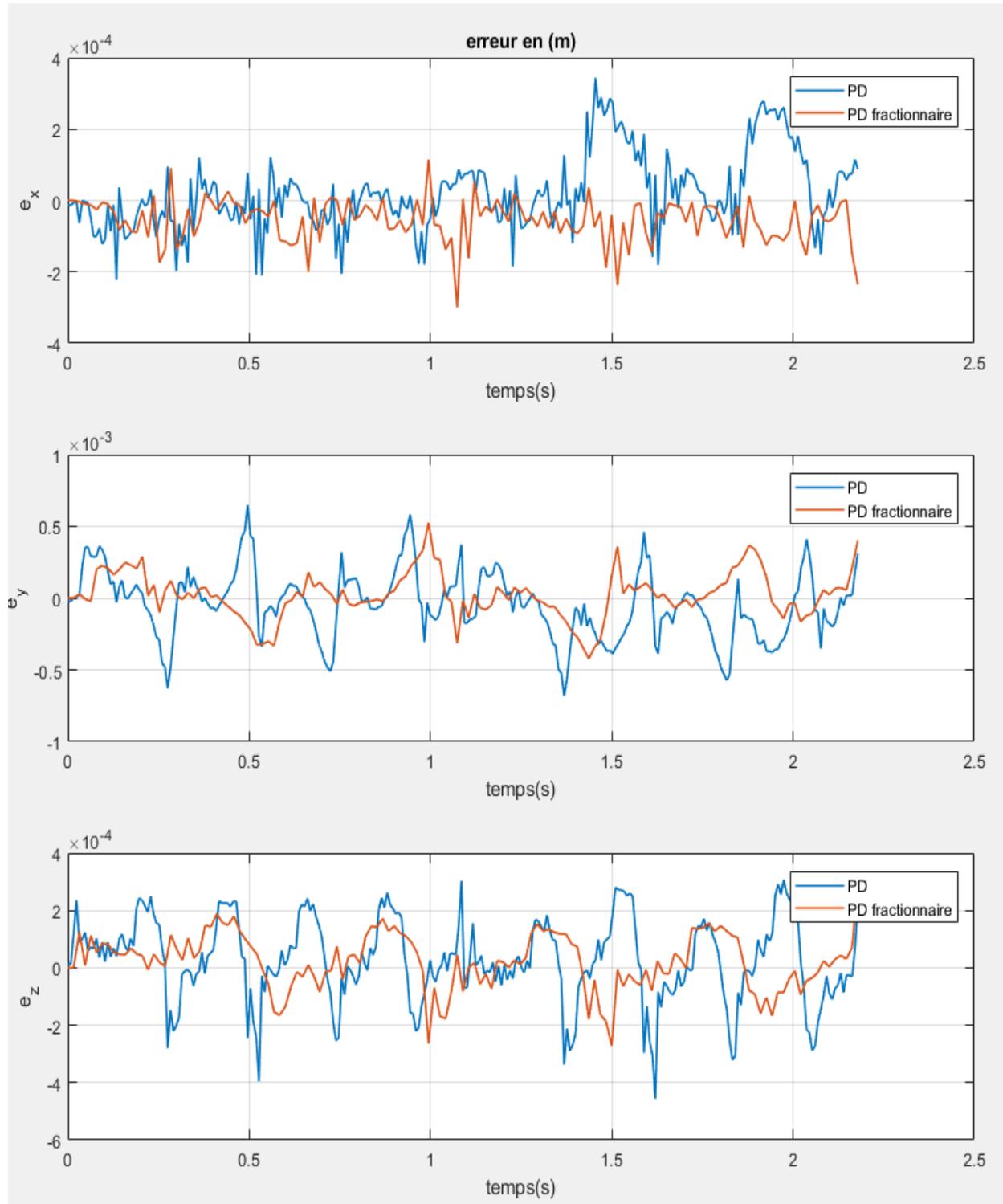


FIGURE 3.21: Les erreurs de position opérationnelle obtenus par l'implémentation des trois régulateurs sur le robot avec une masse de 46g

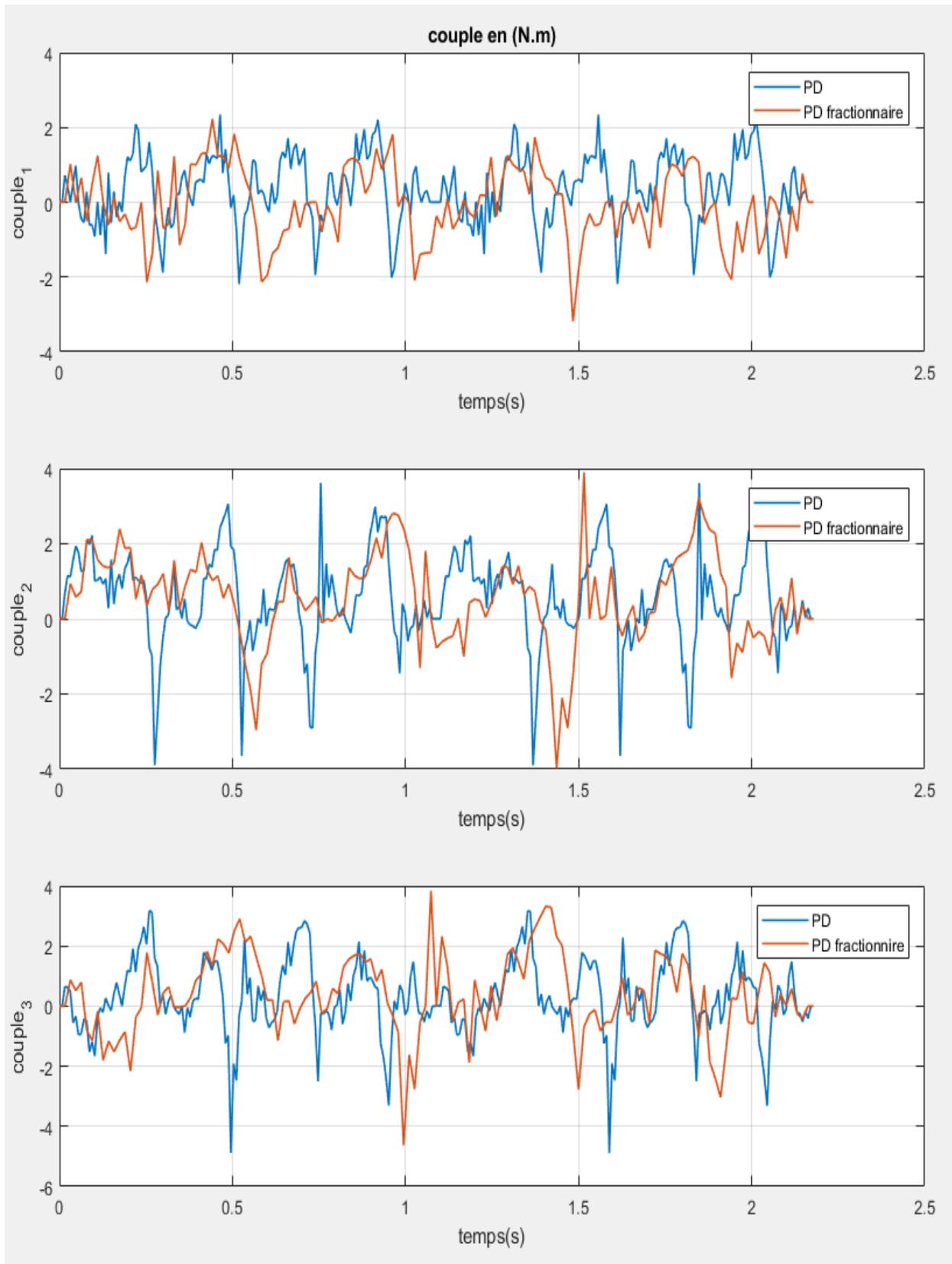


FIGURE 3.22: Les couples obtenus par l'implémentation des trois régulateurs sur le robot avec une masse de 46g

### 3.7.4 Comparaison et discussion

Pour pouvoir comparer les résultats expérimentaux nous avons utilisé un critère statistique qui est le "RMSE (Root Mean Square Error)" c'est-à-dire l'écart-type des erreurs de prédiction, sa formule est donnée par :

$$RMSE = \sqrt{\sum_{i=1}^3 RMSE_{P_i}^2}$$

Avec :

$$RMSE_{P_i} = \sqrt{\frac{1}{N} \sum_{j=1}^N e_i(j)^2}$$

Où :

$N$  : nombre d'échantillons.

$e_i(j)$  : l'erreur de poursuite pour l'articulation  $i$  pendant l'échantillon  $j$ .

Le deuxième critère utilisé est le "Maximum Absolute Error" qui calcule la moyenne de la différence absolue entre deux valeurs (prédite et réelle). Sa formule est donnée par :

$$MaxAE = \max(MaxAE_{p_1}, MaxAE_{p_2}, MaxAE_{p_3})$$

Où :

$$MaxAE_{p_i} = \max(|x_i - x_{di}|)$$

Si nous regardons les erreurs de position opérationnelle représentées par la figure (3.18), on peut facilement constater que le régulateur PD d'ordre fractionnaire est meilleur que le PD classique. Concrètement, le RMSE et le MaxAE du PD classique valent respectivement  $8.0952 \cdot 10^{-4}(m)$  et  $0.0013$  tandis que ceux du PD d'ordre fractionnaire valent  $1.8753 \cdot 10^{-4}(m)$  et  $4.2111 \cdot 10^{-4}(m)$ .

De plus, le PD d'ordre fractionnaire reste le meilleur même lorsqu'on augmente la dynamique du robot du fait qu'il a une valeur RMSE  $2.7320 \cdot 10^{-4}(m)$  et une valeur MaxAE  $6.2013 \cdot 10^{-4}(m)$  face au PD classique qui a une valeur RMSE  $8.5825 \cdot 10^{-4}(m)$  et une valeur MaxAE  $0.0016(m)$ .

D'après la figure (3.21), la robustesse du PD d'ordre fractionnaire est aussi meilleure que celle du PD classique, car il a un RMSE de  $2.0850 \cdot 10^{-4}(m)$  et un MaxAE de  $5.2593 \cdot 10^{-4}(m)$  tandis que les valeurs du RMSE et MaxAE du PD classique sont  $2.7874 \cdot 10^{-4}(m)$  et  $6.4776 \cdot 10^{-4}(m)$  respectivement.

En se basant sur ces résultats expérimentaux, on peut conclure que l'introduction de la dérivée d'ordre fractionnaire a donné une flexibilité considérable au régulateur PD classique en terme de performances et de robustesse.

## 3.8 Conclusion

Le travail de ce chapitre était principalement une amélioration de la commande PD en introduisant la dérivée d'ordre fractionnaire. Cela a nécessité une introduction à la théorie du calcul fractionnaire.

Dans une première partie on a exploité les résultats de la simulation des deux régulateurs, à savoir le PD classique et le PD d'ordre fractionnaire. Dans la deuxième partie on a implémenté ces derniers sur le robot ISIR88 et ça nous a permis de valider les résultats obtenus par la simulation et conclure que le PD d'ordre fractionnaire est meilleur que le classique en termes de robustesse et performances.

Pendant ces expériences on a remarqué que malgré le PD d'ordre fractionnaire est toujours meilleur que le classique, mais tous les deux perdent pas mal de performances avec des dynamiques élevées et des perturbations externes ; pour cela, l'introduction du modèle dynamique dans la loi de commande moyennant la CTC représente une solution très satisfaisante car il permet l'estimation de la non-linéarité et de l'incertitude de la dynamique du robot. Néanmoins, cette méthode est difficile à implémenter à cause de la complexité du modèle dynamique (le calcul de produit et de l'inverse de plusieurs matrices simultanément nécessite un calculateur performant), et pour cela qu'on vient de faire juste la simulation de cette méthode.

# La commande "Time Delay Control"

## 4.1 Introduction

Comme il est souvent difficile d'obtenir un modèle précis et simple pour la conception des commandes, des méthodes de contrôle sans modèle ont été développés au fil des ans, par exemple la commande par mode glissant (SMC) et la commande par logique floue (FLC). Néanmoins, la SMC a encore besoin des informations de la dynamique des robots, dont le calcul est compliqué. Le contrôle flou est une bonne solution pour l'estimation de la non-linéarité et de l'incertitude de la dynamique du robot, mais il souffre de la complexité et de la difficulté de mise en œuvre, en raison de nombreuses règles et paramètres flous. Pour cela, des nouvelles méthodes de commande robuste ont été développées comme la commande à retard où bien (Time Delay Control) qui est une approche très pratique en termes de calcul numérique.

Dans ce chapitre nous allons synthétiser théoriquement les commandes que nous allons appliquer sur le robot Delta ISIR88 à savoir la commande TDC simple, TDC avec mode glissant et TDC avec mode glissant d'ordre fractionnaire. Une interprétation des résultats expérimentaux est aussi donnée avec une étude comparative entre les 3 commandes [25].

## 4.2 "Time Delay Control" (TDC)

L'objectif principal de la TDC est d'utiliser l'observation passée de la réponse du système comme entrée de contrôle à l'instant actuel pour modifier immédiatement les actions de contrôle au lieu d'identifier les paramètres ou d'ajuster le gain du régulateur du système de contrôle, ce qui conduit à un régulateur indépendant de modèle indépendant, c'est-à-dire une compensation sans aucune utilisation de modèle dynamique[25].

Le modèle dynamique du robot est donné par l'équation (2.44), si on multiplie cette dernière par  $\bar{M}^{-1}(\alpha)$  et on résout pour  $\ddot{q}$ , on obtient :

$$\tau = C(\alpha_t, \dot{\alpha}_t)\dot{\alpha}_t + G(\alpha_t) + (M(\alpha_t) - \bar{M}(\alpha_t))\ddot{\alpha}_t + \bar{M}(\alpha_t)\ddot{\alpha}_t \quad (4.1)$$

Où  $t$  indique l'instant actuel et  $\bar{M} = \text{diag}(\bar{M}_1, \bar{M}_2, \bar{M}_3) \in \mathbb{R}^{3 \times 3}$  est une matrice diagonale positive supposée conforme à la dynamique du robot.

$\delta_m \leq M(\alpha_t) \leq \delta_M$ , pour certaines valeurs de  $\delta_M$  et  $\delta_m$ . C'est depuis la matrice d'inertie  $M(\alpha_t)$  est exprimée en termes de  $\sin(\alpha_t)$  et  $\cos(\alpha_t)$ . De l'équation (4.1) on écrit une forme compacte et simple de  $\ddot{\alpha}_t$  comme suit :

$$\ddot{\alpha}_t = N_t + \bar{M}^{-1} \tau_t \quad (4.2)$$

Où :

$$N_t = -\bar{M}^{-1} [C(\alpha_t, \dot{\alpha}_t) \dot{\alpha}_t + G(\alpha_t)] - \bar{M}^{-1} [(M(\alpha_t) - \bar{M}) \ddot{\alpha}] \quad (4.3)$$

Le but de l'approche de contrôle est de construire les angles articulaires  $\alpha_t$  avec précision, d'où l'erreur de poursuite ci-dessous doit tendre vers zéro :

$$e_t = \alpha_{ref,t} - \alpha_t \quad (4.4)$$

Comme  $N_t$  de l'équation (4.2) n'est pas disponible, on utilise son estimation  $N_{t-L}$ . Le régulateur TDC est exprimé par :

$$\bar{\tau}_t = -\bar{M} N_{t-L} + \bar{M} (\ddot{\alpha}_{ref,t} + K_d \dot{e}_t + K_p e_t) \quad (4.5)$$

Où  $K_d = \text{diag}(k_{d1}, k_{d2}, k_{d3})$  et  $K_p = \text{diag}(k_{p1}, k_{p2}, k_{p3})$  sont des matrices positives,  $\ddot{\alpha}_{ref,t} = [\ddot{\alpha}_{ref1,t}, \ddot{\alpha}_{ref2,t}, \ddot{\alpha}_{ref3,t}]$  est l'accélération articulaire désirée,  $\dot{e}_t$  est la dérivée de l'erreur de suivi,  $N_{t-L} = [N_{1,t-L}, N_{2,t-L}, N_{3,t-L}]$ . Avec  $N_{1,t-L}$  est l'estimation de  $N_t$  obtenue par une mesure retardée avec un pas d'exécution, dite estimation à retard (Time Delay Estimation).

L'expression de  $N_{1,t-L}$  est donnée par :

$$N_{1,t-L} = \ddot{\alpha}_{t-L} - \bar{M}^{-1} \tau_{t-L} \quad (4.6)$$

Où  $L$  est la période d'échantillonnage,  $t-L$  est un échantillon passé. Si on remplace l'équation (4.6) dans (4.5), on obtient la commande récursive suivante :

$$\bar{\tau}_t = -\bar{M} \ddot{\alpha}_{t-L} + \tau_{t-L} + \bar{M} (\ddot{\alpha}_{ref,t} + K_d \dot{e}_t + K_p e_t) \quad (4.7)$$

Qui est souvent appelée la commande à retard ou bien (TDC).

Il est à noter que la TDC est bornée, si le gain de contrôle  $\bar{M}$  est sélectionné pour satisfaire la condition suivante :

$$1 - M^{-1}(q_t) \bar{M} < 1 \quad (4.8)$$

Formellement,  $t \geq 0$ ; aussi, l'erreur de TDE est limitée par la constante  $N_i^*$  pour  $i=1,2,3$ . C'est-à-dire  $|N_{i,t} - N_{i,t-L}| \leq N_i^*$ . Ça implique que les gains de contrôle doivent être sélectionnés pour garantir cette limitation. Ainsi, en général, des petits gains de contrôle sont utilisés pour assurer l'inégalité. En outre, si les gains de contrôle sont incorrectement petits, la performance se dégrade. Contrairement, si ceux-ci deviennent trop grands pour une réponse rapide, ils ont tendance à produire une instabilité du système.

La structure de commande TDC est représentée par la figure (4.1) :

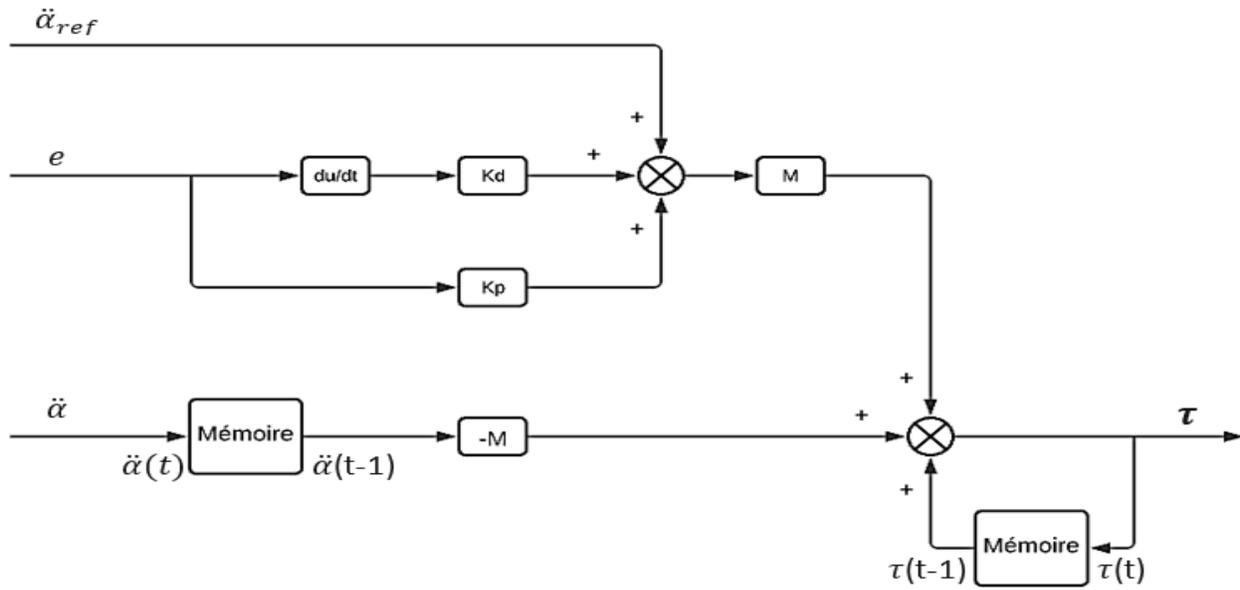


FIGURE 4.1: Le régulateur TDC

### 4.3 La commande TDC avec mode glissant

Afin d'éliminer les non-linéarités dures et augmenter la robustesse du régulateur tout en assurant un temps de convergence fini, un nouveau élément est généralement ajouté [26] [27]. Il est question du ('Terminal Sliding Mode' en anglais signifiant 'mode de glissement terminal') avec différentes surfaces de glissement. Dans cette section nous allons utiliser la surface de glissement donnée par Slotine [28].

#### 4.3.1 Le mode glissant et le phénomène de Chattering

L'approche de mode glissant consiste à amener la variable qu'on désire annuler (l'erreur de poursuite dans notre cas) vers la surface de glissement ( $s$ ). Lorsqu'on atteint la surface de glissement, la convergence de l'erreur est faite à l'aide d'une logique de commutation appropriée autour de celle-ci jusqu' au point d'équilibre, d'où le phénomène de glissement [23]. Cette commutation est due à l'introduction de la fonction  $sign(s)$  dans la loi de commande tel que :

$$sign(s) = \begin{cases} -1 & \text{si } s < 0 \\ 0 & \text{si } s = 0 \\ 1 & \text{si } s > 0 \end{cases}$$

La commutation très rapide de la commande par mode glissant provoque un phénomène appelé "le phénomène de Chattering". Ce dernier est indésirable car il ajoute au spectre de la commande des composantes de haute fréquence. Le broutement (phénomène de Chattering) peut être réduit en remplaçant la fonction «sign» par une

fonction de saturation adéquate qui filtre les hautes fréquences. On donne ci-dessous la fonction qu'on a utilisée :

$$sat(s) = \begin{cases} -1 & si \quad s < -\varepsilon \\ \frac{s}{\varepsilon} & si \quad |s| < \varepsilon \\ 1 & si \quad s > \varepsilon \end{cases}$$

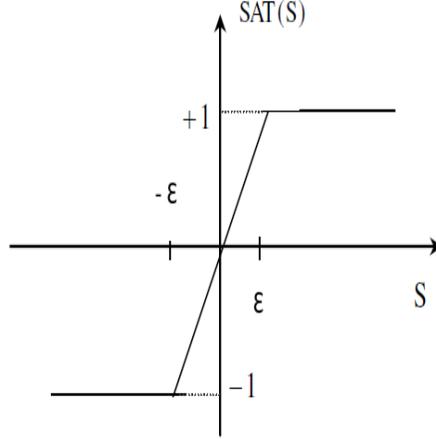


FIGURE 4.2: Fonction "saturation"

### 4.3.2 Conception de la commande et analyse de convergence

Pour atteindre les objectifs de contrôle décrits dans la section précédente, nous devons d'abord déterminer la surface de glissement de Slotine [28] donnée par l'équation (4.9) :

$$s_{1t} = \dot{e}_t + K_d e_t \quad (4.9)$$

Où  $s_{1t} = [s_{1t,1}, s_{1t,2}, s_{1t,3}] \in \mathbb{R}^3$ ,  $K_d = \text{diag}(k_{d1}, k_{d2}, k_{d3}) \in \mathbb{R}^3$  et  $e_t = [e_{t,1}, e_{t,2}, e_{t,3}] \in \mathbb{R}^3$ .

On reprend le modèle dynamique du robot donné par l'équation (2.44) mais cette fois ci en ajoutant une perturbation  $d$  :

$$\tau = M(\alpha)\ddot{\alpha} + C(\alpha, \dot{\alpha})\dot{\alpha} + G(\alpha) + d \quad (4.10)$$

Considérons que nous avons l'erreur des capteurs ( $\dot{\alpha}_t = \dot{\alpha} + \Delta\dot{\alpha}$ ). Par conséquent, le modèle dynamique devient :

$$\tau = M(\alpha_t)\ddot{\alpha}_t + C(\alpha_t, \dot{\alpha}_t)\dot{\alpha}_t + G(\alpha_t) + d_1 \quad (4.11)$$

Où  $d_1$  est la perturbation globale qui est exprimée par :

$$d_1 = M(\alpha)\ddot{\alpha} + C(\alpha, \dot{\alpha})\dot{\alpha} + G(\alpha) - M(\alpha_t)\ddot{\alpha}_t - C(\alpha_t, \dot{\alpha}_t)\dot{\alpha}_t - G(\alpha_t) + d \quad (4.12)$$

Par conséquent,  $N_t$  est donné par :

$$N_t = -\bar{M}^{-1}(C(\alpha_t, \dot{\alpha}_t)\dot{\alpha}_t + G(\alpha_t) + d_1) - \bar{M}^{-1}[(M - \bar{M})\ddot{\alpha}] \quad (4.13)$$

Ensuite, le TDE est utilisé pour estimer  $N_t$ , ce qui nous permet d'évaluer le modèle dynamique sans nécessiter une connaissance préalable ni sur le système ni sur les erreurs des capteurs. De plus, la TDC peut utiliser cette estimation pour compenser l'effet des perturbations globales et atteindre une bonne performance de poursuite.

La loi de commande proposée est la suivante :

$$\tau_t = \tau_{t-L} - \bar{M}\ddot{\alpha}_{t-L} + \bar{M}(\ddot{\alpha}_{ref,t} + K_d\dot{e}_t + k_1\text{sign}(s_1)) \quad (4.14)$$

Cette loi doit garantir la convergence asymptotique de l'erreur de poursuite et sa dérivée par rapport au temps. Pour cela on utilise le théorème de Lyapunov, avec la fonction de Lyapunov est donnée par :

$$V_1 = \frac{1}{2}s_{1t}^T s_{1t} = \frac{1}{2} \sum_{i=1}^3 s_{1t(i)}^2 \quad (4.15)$$

Sa dérivée par rapport au temps est donnée par :

$$\dot{V}_1 = s_{1t}^T \dot{s}_{1t} \quad (4.16)$$

En introduisant les équations (4.9) et (4.16) :

$$\dot{V}_1 = s_{1t}^T (\dot{e}_t + k_d \dot{e}_t) = s_{1t}^T (\ddot{\alpha}_{ref,t} - \ddot{\alpha}_t + k_d \dot{e}_t) \quad (4.17)$$

On remplace  $\ddot{\alpha}_t$  par son expression donnée par l'équation (4.2), on obtient :

$$\dot{V}_1 = s_{1t}^T (\ddot{\alpha} + k_d \dot{e}_t - \bar{M}^{-1} \tau_t - N_t) \quad (4.18)$$

En appliquant la loi de commande donnée par l'équation (4.14) on obtient :

$$\dot{V}_1 = s_{1t}^T \left[ \ddot{\alpha}_{ref,t} + k_d \dot{e}_t - N_t - \bar{M}^{-1} (\tau_{t-L} - \bar{M}\ddot{\alpha}_{t-L} + \bar{M}(\ddot{\alpha}_{ref,t} + K_d\dot{e}_t + k_1\text{sign}(s_1))) \right] \quad (4.19)$$

Donc :

$$\dot{V}_1 = s_{1t}^T \left( -N_t - \bar{M}^{-1} (\tau_{t-L} - \bar{M}\ddot{\alpha}_{t-L}) - K_1 \text{sign}(s_1) \right) \quad (4.20)$$

D'après l'équation (4.2), on peut obtenir la suivante :

$$\dot{V}_1 = s_{1t}^T (-N_t + N_{t-L} - K_1 \text{sign}(s_1)) \quad (4.21)$$

Comme  $\|N_t - N_{t-L}\|_\infty \leq N^*$ , donc :

$$\dot{V}_1 \leq \sum_1^3 |s_{1i}| (N^* - K_{1,m}) \quad (4.22)$$

Où  $K_{1,m}$  est la valeur propre minimale de la matrice  $K_1$ . Alors, si on choisie  $K_{1,m} > N^*$ , donc  $\dot{V}_1$  devient définie négative. Par conséquent, à partir de la définition de  $V_1$ , nous pouvons conclure que  $s_t$  converge asymptotiquement vers zéro :

$$s_{1t} = \dot{e}_t + K_d e_t \rightarrow 0, \quad t \rightarrow \infty$$

En conséquence, l'erreur de suivi et sa dérivée convergent asymptotiquement vers zéro :

$$e_t \rightarrow 0, \dot{e}_t \rightarrow 0, \quad t \rightarrow \infty$$

Finalement, on donne la structure de commande utilisée par la figure (4.3) :

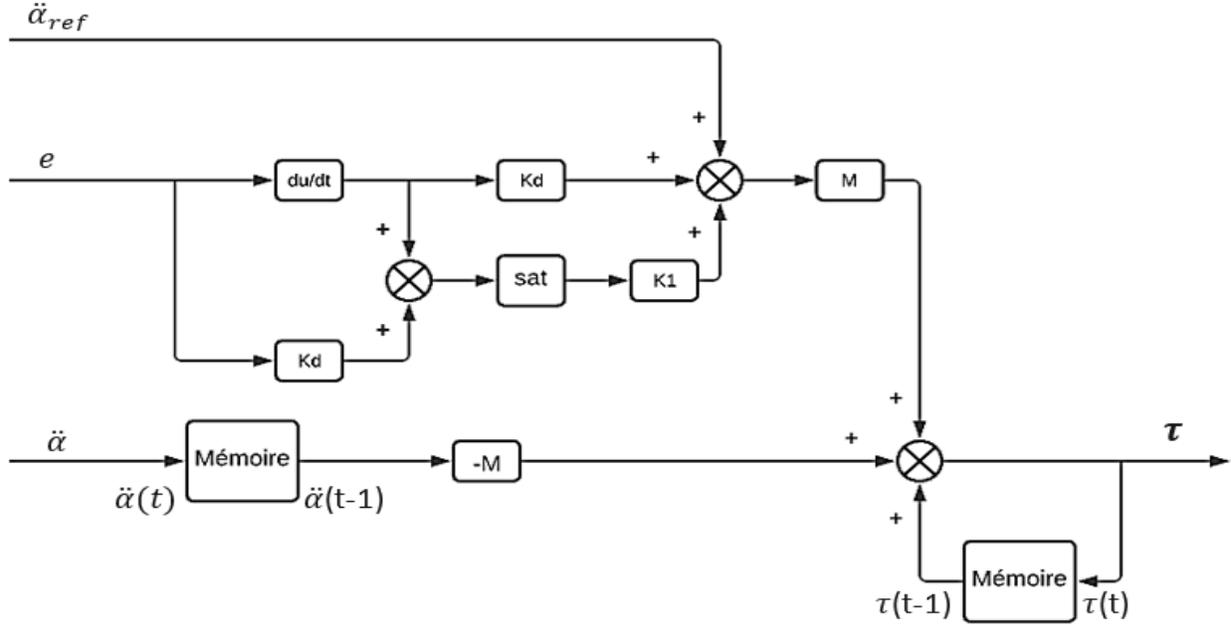


FIGURE 4.3: Le régulateur TDC avec mode glissant

## 4.4 La commande TDC avec mode glissant d'ordre fractionnaire

Dans cette section, une commande similaire à la précédente sera exploitée sauf que la surface de glissement cette fois-ci est d'ordre fractionnaire.

### 4.4.1 Conception de la commande et analyse de convergence

Afin d'améliorer les performances déjà atteintes par la commande précédente, on choisit la surface de glissement d'ordre fractionnaire proposée par Osman Eray et Sezai Tokat [29], donnée par :

$$s_{2t} = K_d e_t + D^{(1-\lambda)} e_t \quad (4.23)$$

La loi de commande proposée est donc la suivante :

$$\tau_t = \tau_{t-L} - \bar{M} \ddot{\alpha}_{t-L} + \bar{M} (\ddot{\alpha}_{ref,t} + D^{(1-\lambda)} K_d e_t + k_2 \text{sign}(s_2)) \quad (4.24)$$

Pour étudier la convergence de cette loi nous suivons les mêmes étapes que pour la précédente :

$$\dot{V}_2 = s_{2t}^T (K_d \dot{e}_t + D^{-\lambda} (\ddot{\alpha}_{ref,t} - \ddot{\alpha}_t)) \quad (4.25)$$

On remplace  $\ddot{\alpha}_t$  par son expression donnée par l'équation (4.2) :

$$\dot{V}_2 = s_{2t}^T (K_d \dot{e}_t + D^{-\lambda} \ddot{\alpha}_{ref,t} - D^{-\lambda} (N_t + \bar{M}^{-1} \tau_t)) \quad (4.26)$$

On remplace l'expression du couple donnée par l'équation (4.24) :

$$\dot{V}_2 = s_{2t}^T \left( K_d \dot{e}_t + D^{-\lambda} \ddot{\alpha}_{ref,t} - D^{-\lambda} N_t - D^{-\lambda} \bar{M}^{-1} \left( \tau_{t-L} - \bar{M} \ddot{\alpha}_{t-L} + \bar{M} (\ddot{\alpha}_{ref,t} + D^{(1-\lambda)} K_d e_t + k_2 \text{sign}(s_2)) \right) \right) \quad (4.27)$$

Après simplification, on trouve :

$$\dot{V}_2 = s_{2t}^T \left( -D^{-\lambda} N_t + -D^{-\lambda} N_{t-L} - D^{-\lambda} K_2 \text{sign}(s_2) \right) \quad (4.28)$$

Comme  $\|N_t - N_{t-L}\|_\infty \leq N^*$ , donc :

$$\dot{V}_2 \leq \sum_1^3 |s_{2i}| (D^{(-\lambda)} N^* - D^{(-\lambda)} K_{2,m}) \quad (4.29)$$

Tel que  $K_{2,m}$  est la valeur propre minimale de la matrice  $K_2$ . Alors, si on choisie  $K_{2,m} > N^*$ , donc  $\dot{V}_2$  devient définie négative. Par conséquent, à partir de la définition de  $V_2$ , nous pouvons conclure que  $s_t$  converge asymptotiquement vers zéro.

La structure de cette commande est représentée dans la figure (4.4) :

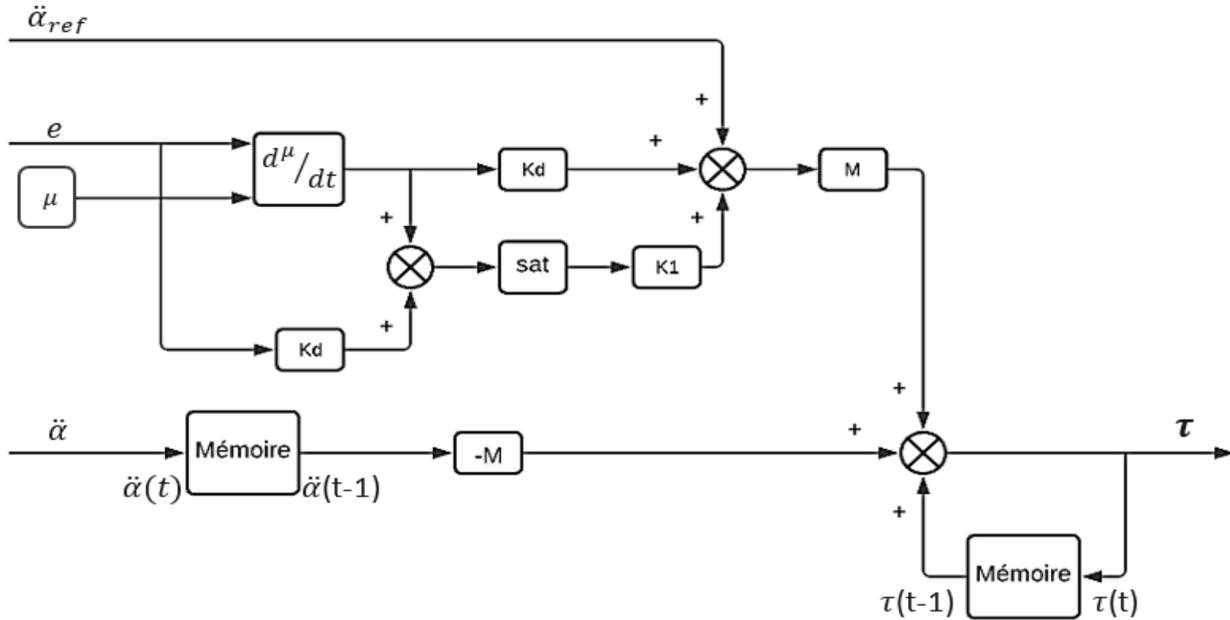


FIGURE 4.4: Le régulateur TDC avec mode glissant fractionnaire

## 4.5 Simulation

Dans cette section nous présenterons les résultats de la simulation des trois lois de commande sur le robot ISIR88 à l'aide de "Matlab SIMULINK", tel que les paramètres de chacune sont donnés comme suit :

- **La commande TDC :**  
 $\bar{M} = 2.416 \cdot 10^{-5}$ ;  $K_p = 2942.4744$ ;  $K_d = 267.4649$ .
- **La commande TDC avec mode glissant :**  
 $\bar{M} = 2.416 \cdot 10^{-5}$ ;  $K_1 = 1.46146$ ;  $K_d = 588.49487$ .
- **La commande TDC avec mode glissant d'ordre fractionnaire :**  
 $\bar{M} = 2.416 \cdot 10^{-5}$ ;  $K_1 = 1.46146$ ;  $K_d = 588.49487$ ;  $\lambda = 0.97$

### 4.5.1 Test de poursuite

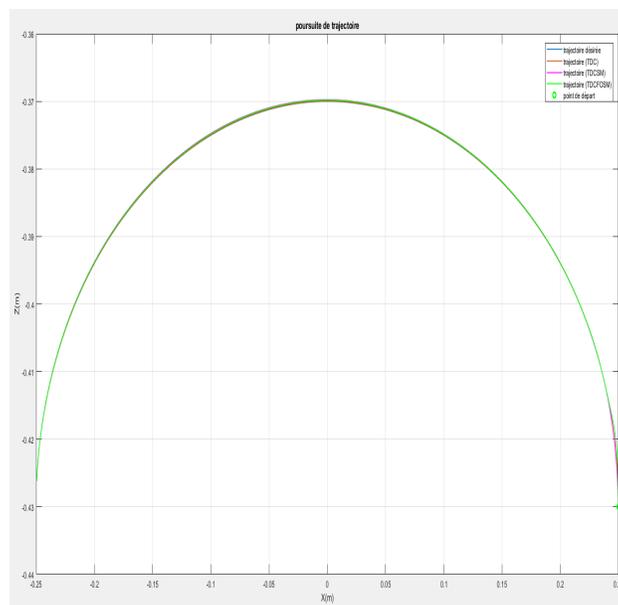


FIGURE 4.5: Résultats obtenus par la simulation des trois commandes TDC (suivie de trajectoire)

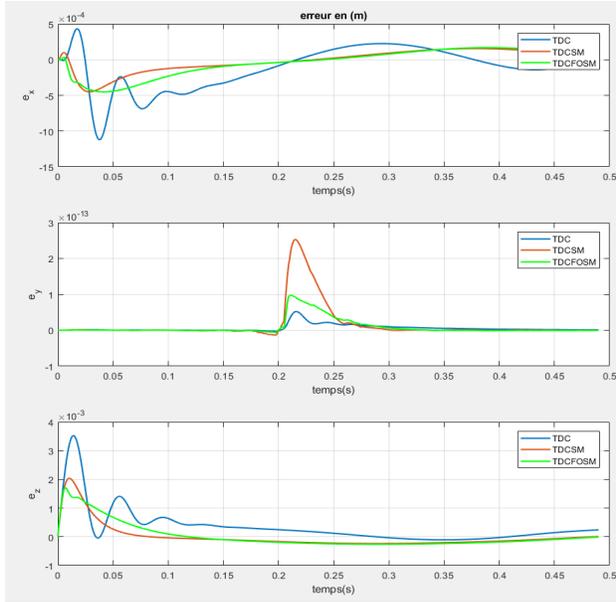


FIGURE 4.6: Les erreurs de position opérationnelle obtenus par les trois commandes TDC

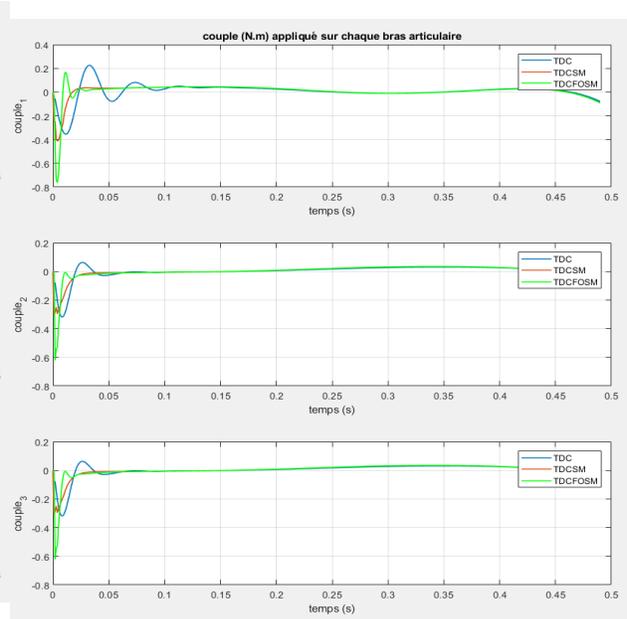


FIGURE 4.7: Les couples obtenus par la simulation des trois commandes TDC

### 4.5.2 Test de robustesse

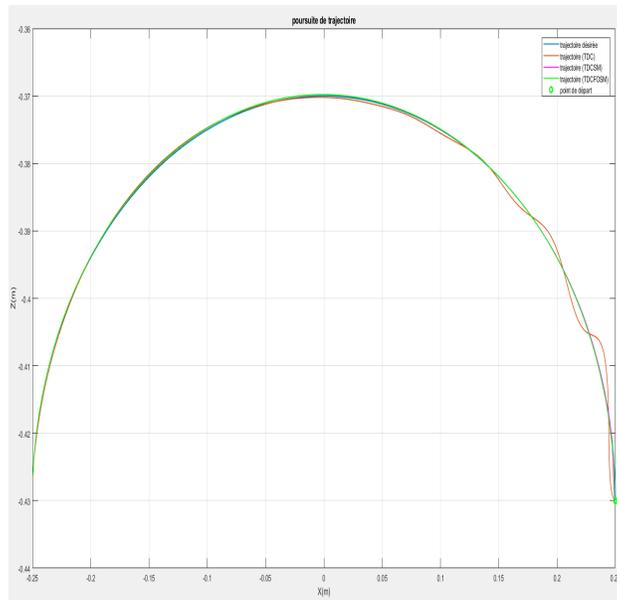


FIGURE 4.8: Résultats obtenus par la simulation des trois commandes TDC (suivie de trajectoire) avec une masse de 200g

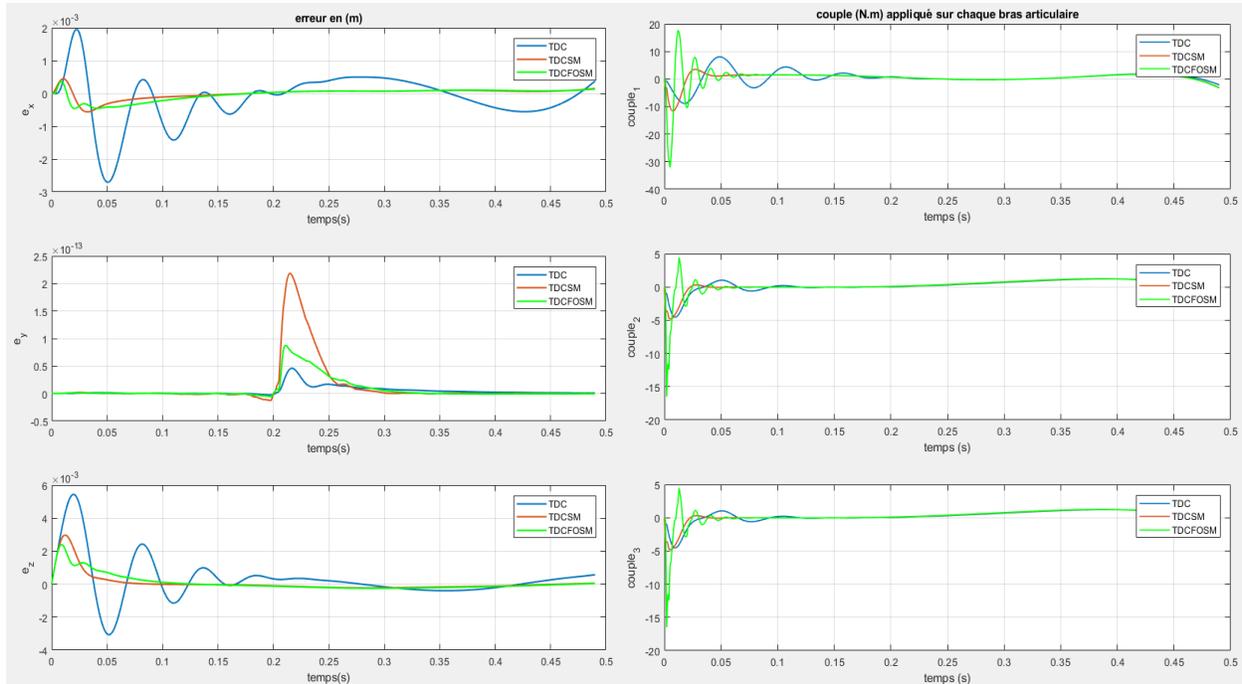


FIGURE 4.9: Les erreurs de position articulaire obtenus par les trois commandes TDC avec une masse de 200g

FIGURE 4.10: Les couples obtenus par la simulation des trois commandes TDC avec une masse de 200g

Cette-fois-ci la figure (4.5) nous ne permettons pas de bien visualiser la différence entre les trois commandes, mais en se référant à la figure (4.6) on peut voir que la combinaison de la TDC avec le mode glissant améliore remarquablement la poursuite de trajectoire. De plus, l'introduction de la dérivée d'ordre fractionnaire contribue aussi à l'amélioration des performances mais d'une façon légère. Les courbes de la commande en couple de la figure (4.7) montre que la commande TDC avec mode glissant et la TDC avec mode glissant d'ordre fractionnaire sont optimales en terme d'énergie.

Les tests de robustesse ont donné des bons résultats du fait que la commande TDC en général est dépendante du modèle dynamique et donc rejette les perturbations y compris la masse supplémentaire qu'on a ajoutée, mais la figure (4.9) nous donne habilité de favoriser un peu la commande TDC avec mode glissant d'ordre fractionnaire car les erreurs de suivi sont légèrement petites par rapport à la TDC avec mode glissant entier qui est à son tour mieux que la TDC simple.

## 4.6 Expérimentation

Dans cette section on va implémenter sur le robot les trois commandes présentées précédemment, à savoir la commande TDC, TDC avec mode glissant et TDC avec le mode glissant fractionnaire. Les essais sont effectués avec 3 environnements différents, le premier avec une dynamique faible (une accélération maximale de  $10m/s^2$  et une vitesse maximale de  $5m/s$ ); le deuxième avec une dynamique plus élevée (une accélération

maximale de  $15m/s^2$  et une vitesse maximale de  $7.5m/s$ ) et un dernier test en ajoutant une masse de (46g) en vue de tester la robustesse. Les trois tests ont été effectués avec la même trajectoire de la fiugre (3.16) et les mêmes constantes utilisés dans la partie simulation;

#### 4.6.1 Test avec une dynamique faible

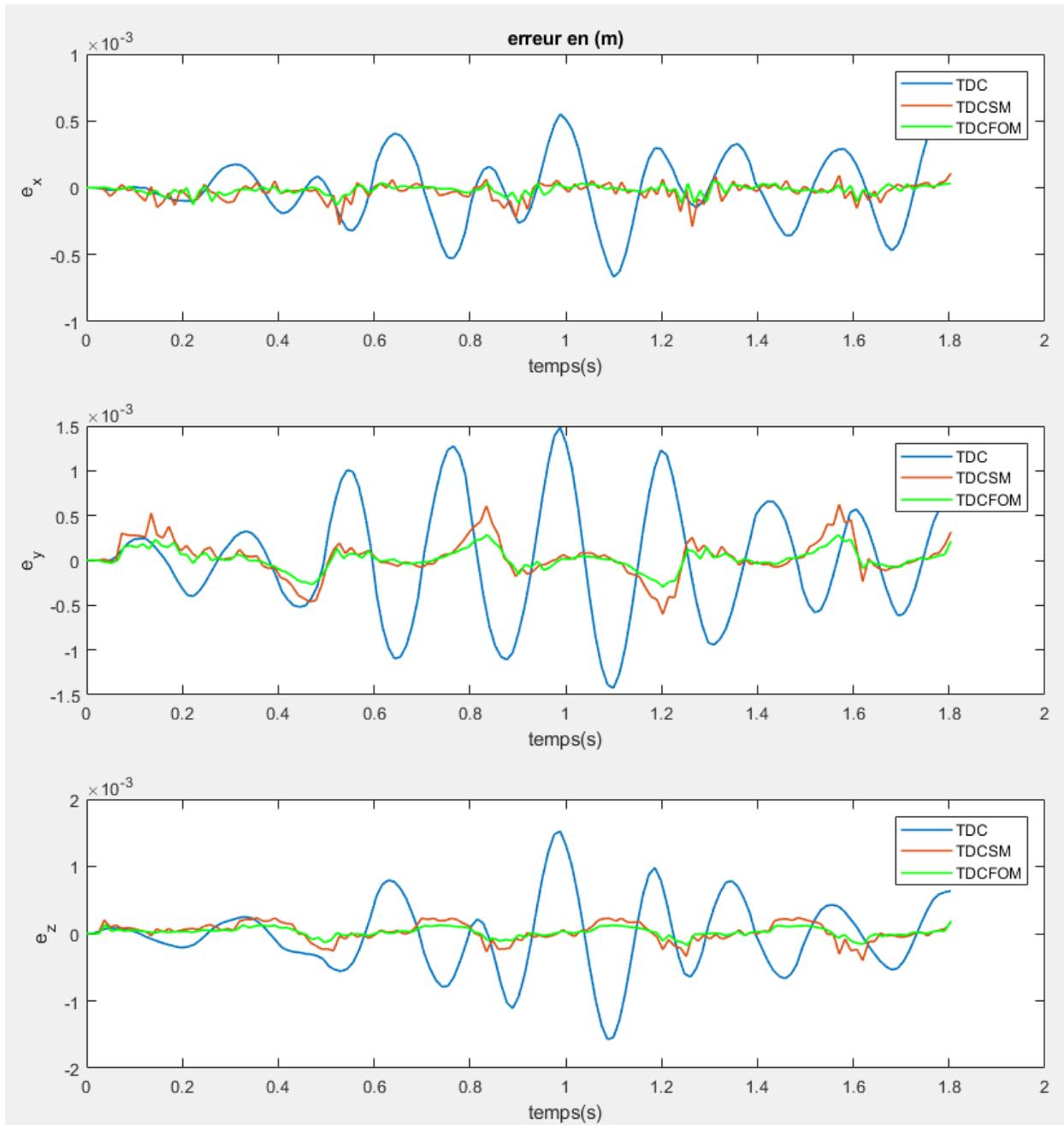


FIGURE 4.11: Erreurs de position opérationnelle des trois commandes TDC pour une dynamique faible

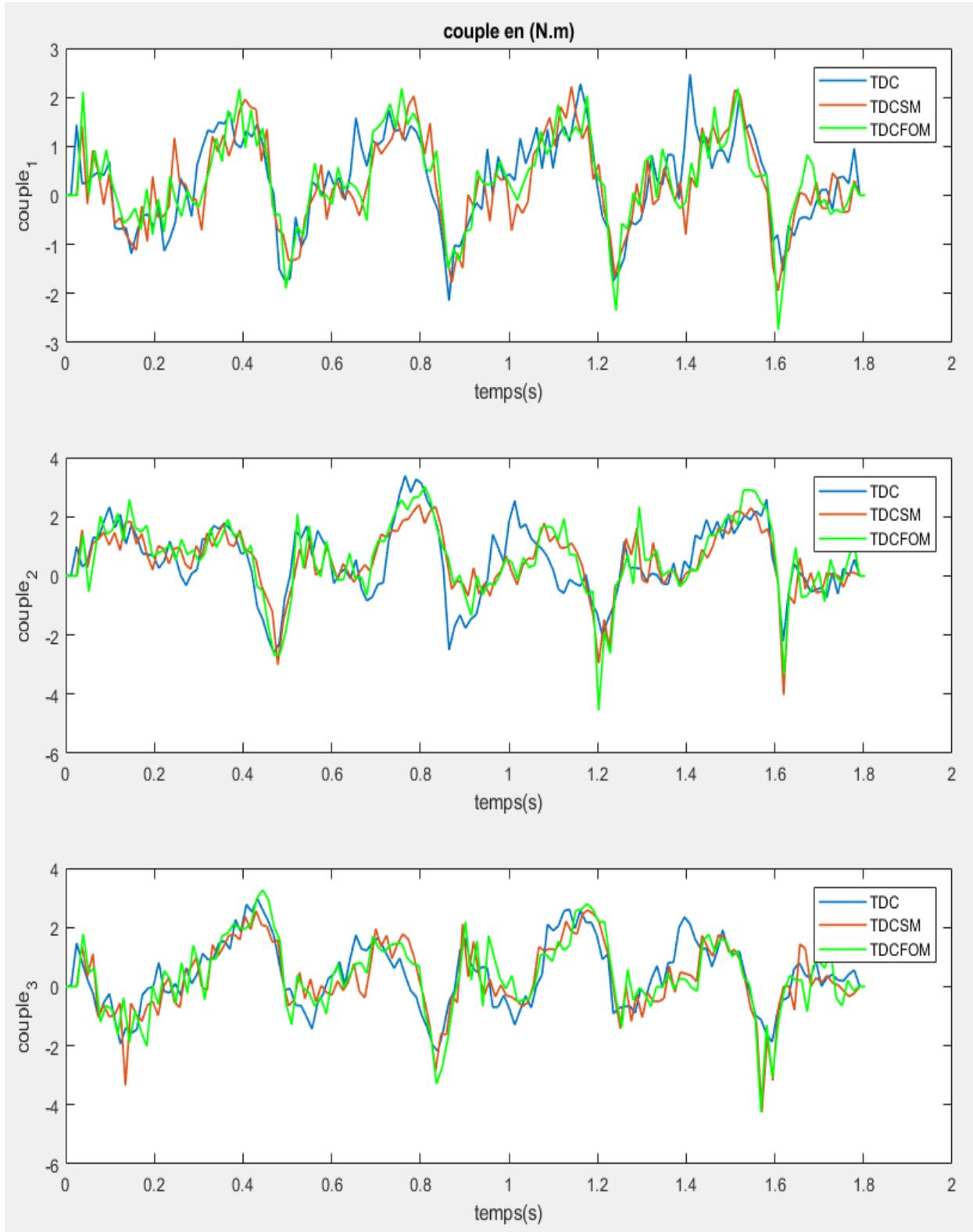


FIGURE 4.12: Couples appliqués par les trois commandes TDC pour une dynamique faible

### 4.6.2 Test avec une dynamique plus élevée

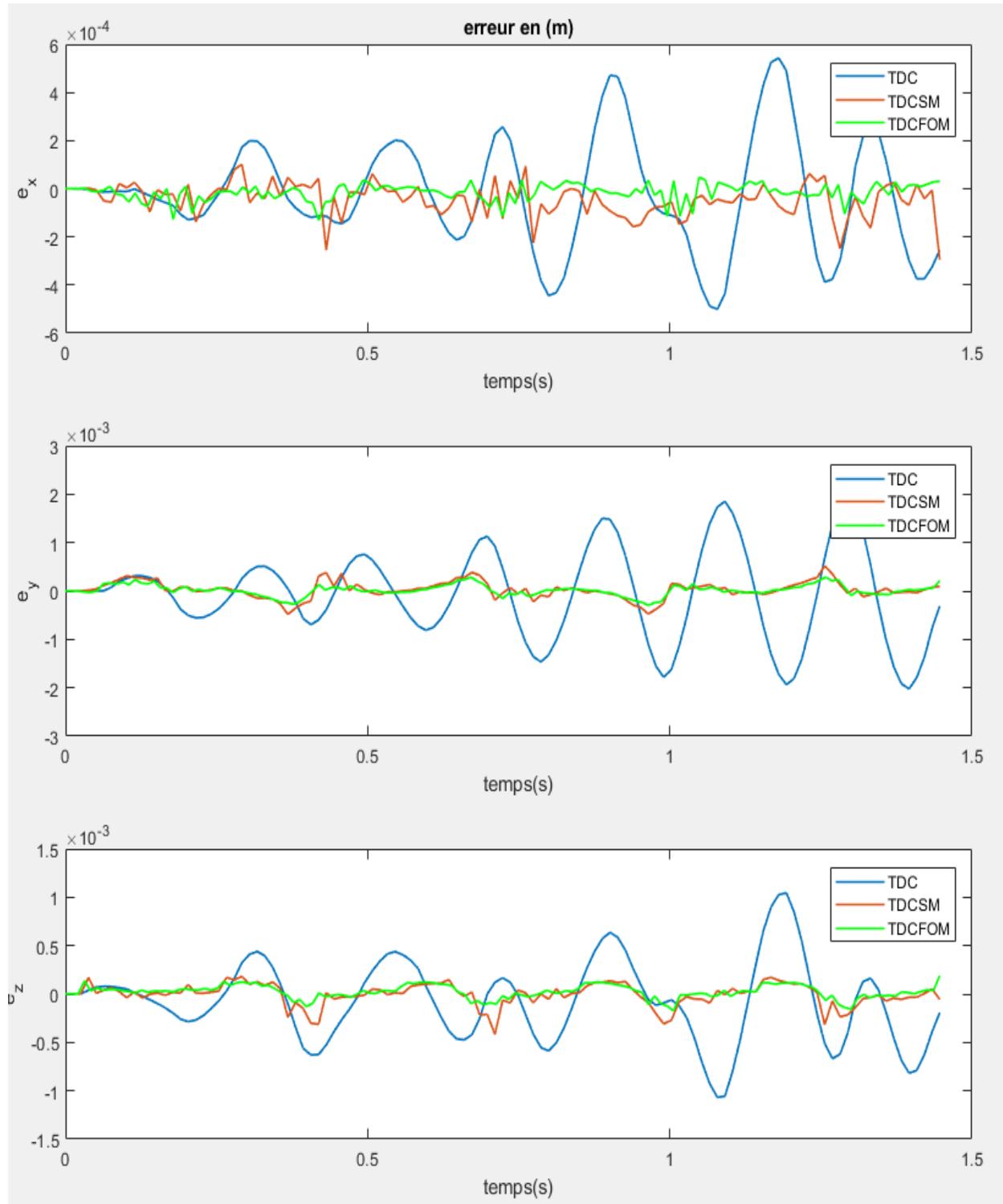


FIGURE 4.13: Erreurs de position opérationnelle des trois commandes TDC pour une dynamique plus élevée

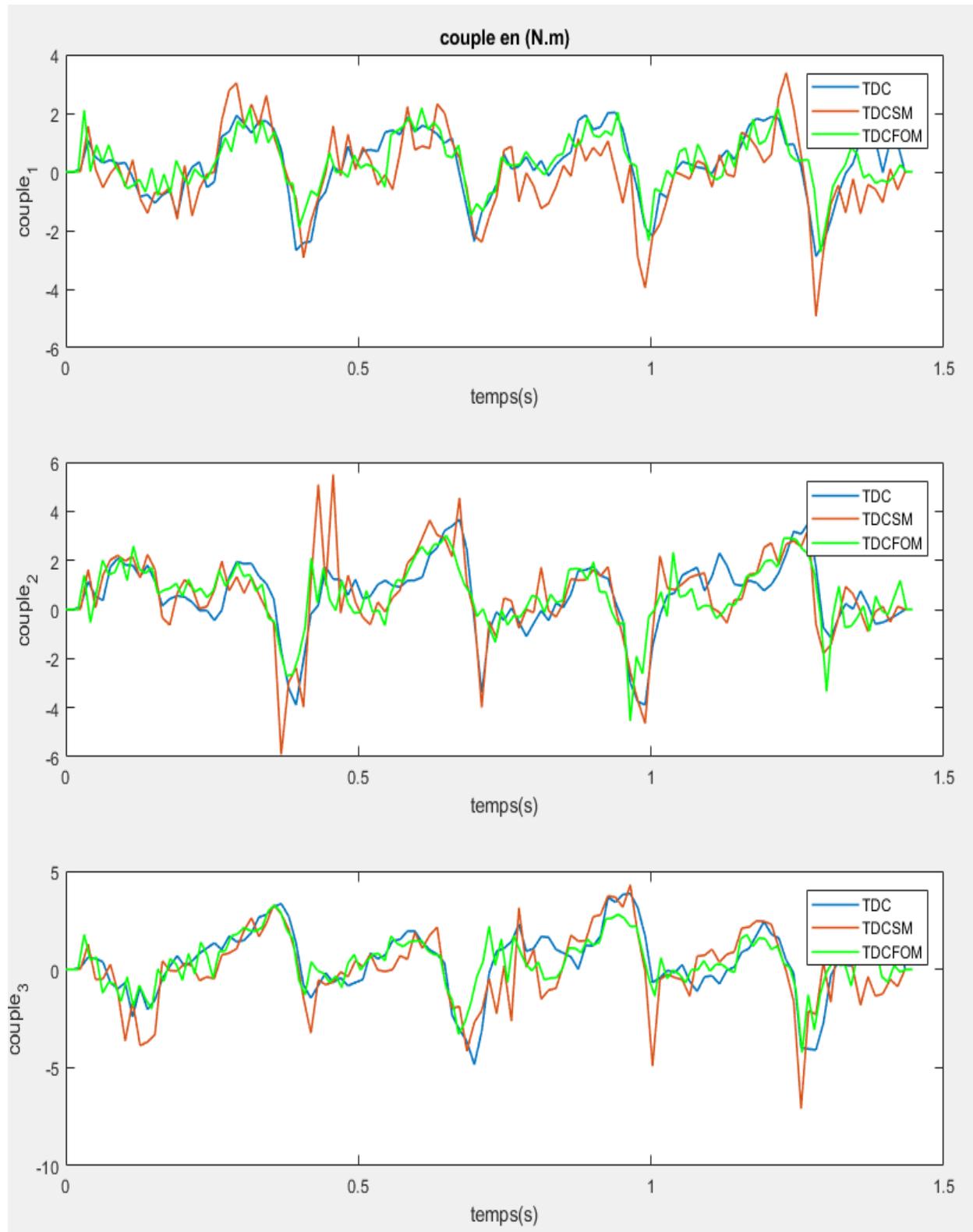


FIGURE 4.14: Couples appliqués par les trois commandes TDC pour une dynamique plus élevée

### 4.6.3 Test de robustesse

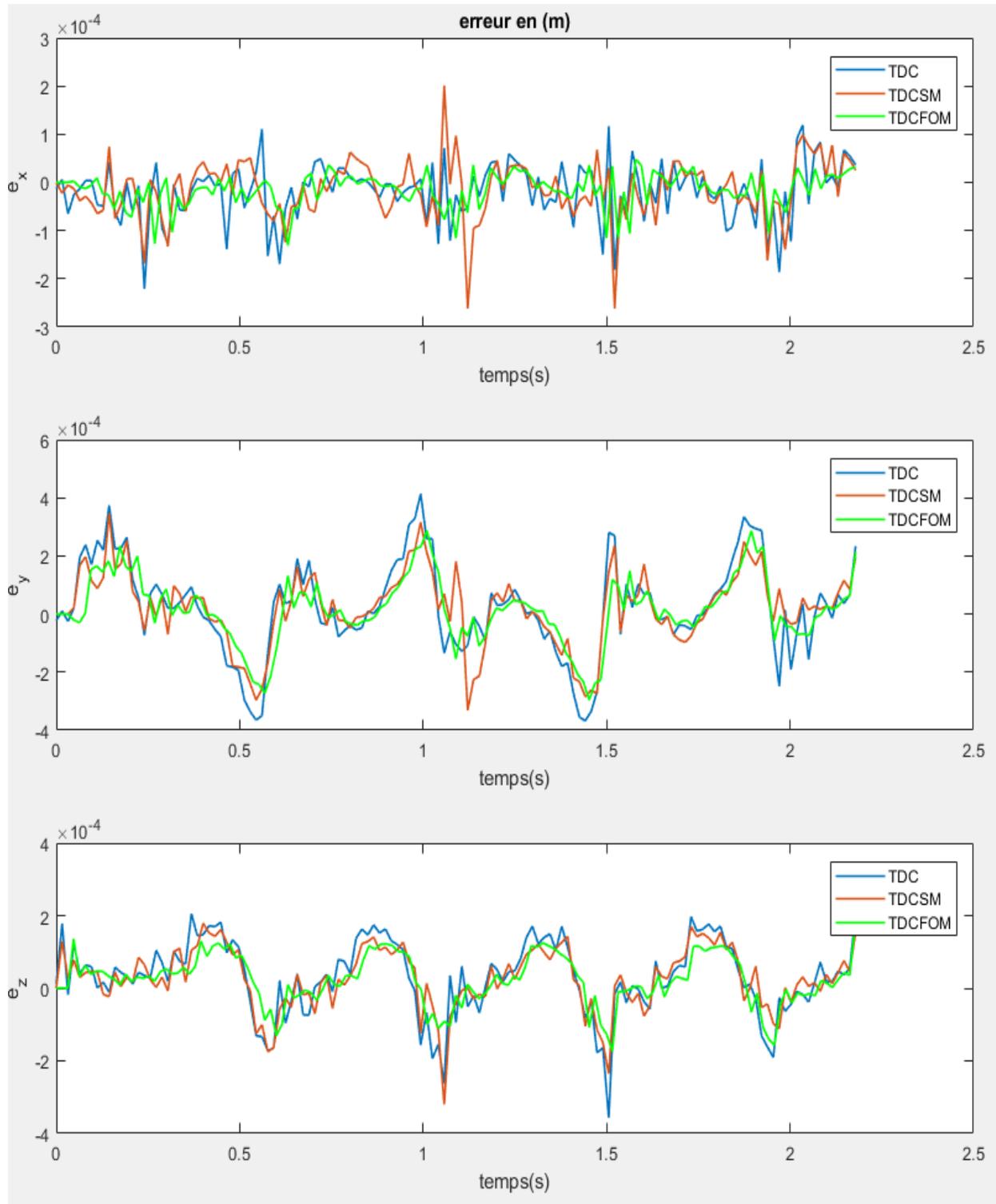


FIGURE 4.15: Erreurs de position opérationnelle des trois commandes TDC avec une masse de 46g

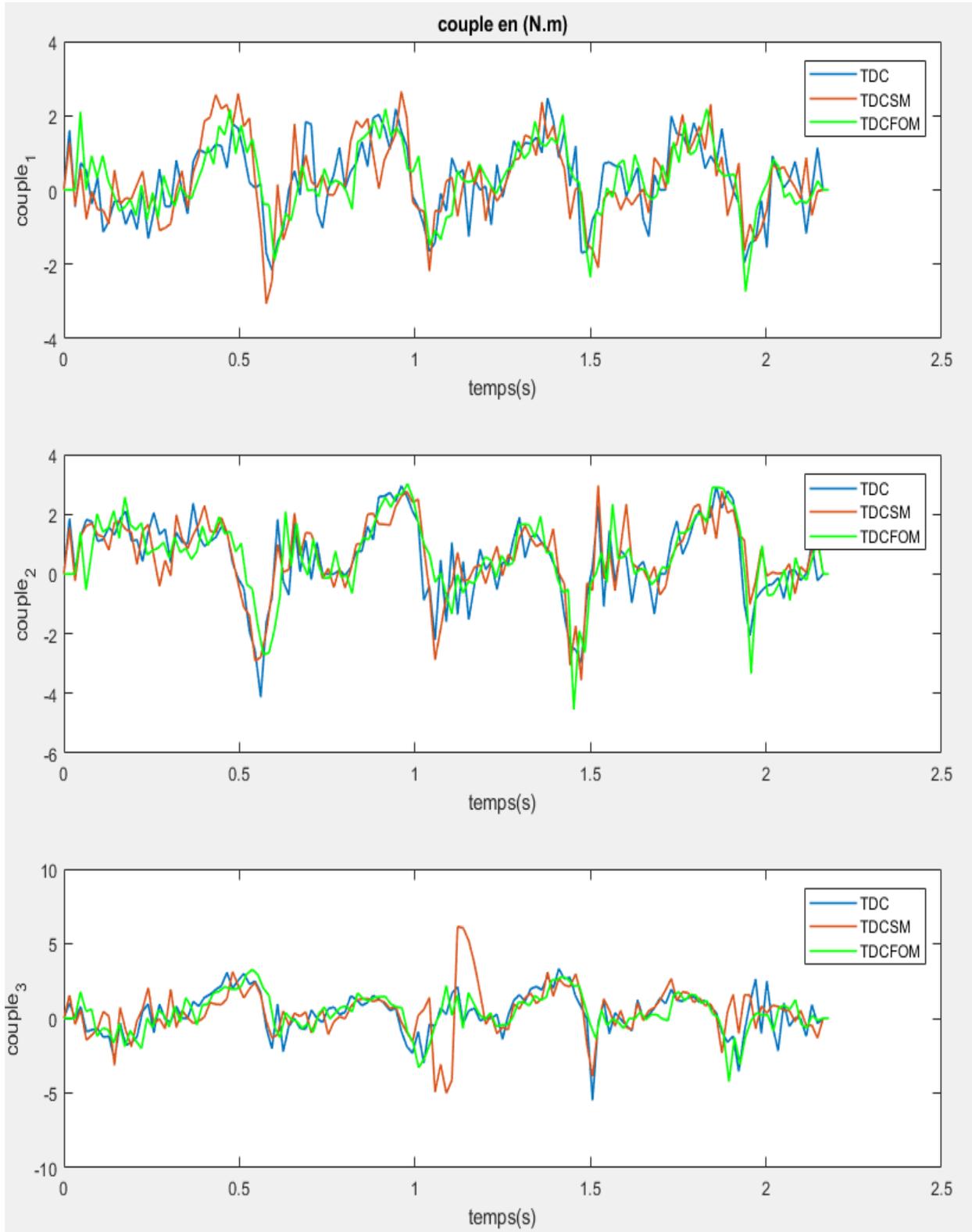


FIGURE 4.16: Couples appliqués par les trois commandes TDC avec une masse de 46g

#### 4.6.4 Comparaison et discussion

Les résultats de l'expérimentation viennent de valider ceux de la simulation. En effet, les erreurs de poursuite de la figure (4.11) ont diminué lorsqu'on a introduit le TSM. Autrement dit, les valeurs de (RMSE/MaxAE) de la commande TDC et ceux de la TDC avec mode glissant et la TDC avec mode glissant d'ordre fractionnaire valent  $(2.772 \cdot 10^{-4}(m)/6.012 \cdot 10^{-4}(m))$ ,  $(2.593 \cdot 10^{-4}(m)/6.226 \cdot 10^{-4}(m))$  et  $(6.425 \cdot 10^{-5}(m)/1.660 \cdot 10^{-4}(m))$  respectivement.

Lorsqu'on augmente la dynamique du robot, la commande TDC perd pas mal de ses performances tel que  $RMSE = 9.105 \cdot 10^{-4}$  et  $MaxAE = 0.00274$ , par contre la TDC avec mode glissant et celle avec mode glissant d'ordre fractionnaire sont meilleures avec une supériorité légère du mode glissant d'ordre fractionnaire traduite par les valeurs (RMSE/MaxAE) égales à  $(1.2482 \cdot 10^{-4}(m)/2.9100 \cdot 10^{-4}(m))$  pour la fractionnaire et  $(2.2618 \cdot 10^{-4}(m)/5.1732 \cdot 10^{-4}(m))$  pour le mode glissant d'ordre entier.

Même en présence d'une perturbation (masse supplémentaire dans notre cas) la TDC avec mode glissant d'ordre fractionnaire reste toujours en première place suivie par la TDC avec mode glissant en deuxième place du fait que l'apport d'ajout d'un élément de robustesse (TSM) est la compensation des non-linéarités dures causées par les changements brusques et qui dégradent les performances de l'estimation TDE. La TDC est moins robuste à cause de l'absence de cet élément. Statistiquement, le (RMSE/MaxAE) de chacune des 3 commandes valent :  $(2.0341 \cdot 10^{-4}(m)/4.1384 \cdot 10^{-4}(m))$  pour la TDC;  $(1.6824 \cdot 10^{-4}(m)/3.4794 \cdot 10^{-4}(m))$  pour la TDC avec mode glissant et  $(1.4060 \cdot 10^{-4}(m)/2.9643 \cdot 10^{-4}(m))$  pour la TDC avec mode glissant d'ordre fractionnaire.

Énergiquement parlant, et en se référant aux figures (4.12), (4.14) et (4.16), on constate que la TDC avec mode glissant d'ordre fractionnaire garde sa supériorité suivie par la TDC avec mode glissant et en dernière place on trouve la TDC simple. D'après les résultats obtenus, on peut conclure que l'introduction de TSM améliore remarquablement les performances de la commande TDC. D'autre part, le choix d'une surface de glissement d'ordre fractionnaire donne plus de flexibilité à la commande (plus de paramètres à ajuster) et donc plus de robustesse et plus de performances.

## 4.7 Conclusion

Bien que la commande TDC soit une commande robuste indépendante du modèle grâce à l'estimateur TDE, on lui a pu provoquer des améliorations grâce au travail effectué dans ce chapitre.

Dans un premier temps, on a combiné la TDC avec le mode glissant en choisissant à la fois une surface simple qui est celle de Slotine et une autre surface d'ordre fractionnaire proposée par [29]. Ces commandes ont été simulées avec Matlab SIMULINK et c'est ce qui a conduit à conclure que le mode glissant fournit plus de performances et de robustesse et que l'introduction de l'opérateur d'ordre fractionnaire fournit à son rôle plus de flexibilité à la commande TDC.

Ensuite, ces résultats ont été validés par l'expérimentation.

# Conclusion générale

Comme le titre de ce mémoire indique, notre travail s'est focalisé sur deux aspects du robot Delta ISIR88 disponible au niveau du "LCP" : la modélisation et la commande.

Concernant la modélisation, le robot Delta présente pas mal de difficultés pour identifier ses différents modèles, à savoir le modèle géométrique, cinématique et dynamique. La modélisation géométrique est assez compliquée à cause de l'absence d'une méthode systématique, mais le fait qu'à chaque instant et quelque soit la configuration articulaire, les coudes des trois chaînes appartiennent à une même sphère nous a permis de le déterminer. Pour la modélisation cinématique on a utilisé un ensemble d'équations de contraintes reliant les coordonnées articulaires aux coordonnées cartésiennes ; et par dérivation on a pu obtenir le modèle des accélérations. Une modélisation dynamique réaliste est très importante pour avoir des simulations crédibles et pour l'amélioration de la précision des lois de commande, alors on a choisi de le calculer en se basant sur le principe de travail virtuel.

Pour ce qui est de la commande, on a commencé par un régulateur très basique qui est le proportionnel dérivé ; mais vu que notre système est très compliqué et comporte pas mal de non-linéarités, ce dernier n'est pas le meilleur choix, pour cela on a essayé de l'améliorer tout d'abord en remplaçant l'action dérivée par une dérivée d'ordre fractionnaire. Le régulateur obtenu nous a posé un problème de synthèse donc on s'est orienté vers une méthode qui repose sur la toolbox Matlab "Simulink Design Optimization" qui permet l'optimisation des paramètres d'un système à partir d'un schéma-bloc Simulink. Cette méthode on l'a ensuite utilisée pour l'ajustement des paramètres de tous les commandes utilisées dans ce travail. Ensuite, on a apporté une deuxième modification sur le régulateur PD qui est l'introduction du modèle dynamique à priori moyennant la méthode de couple calculé. Ces trois régulateurs ont été simulés à l'aide de Matlab SIMULINK dans un premier temps puis implémentés sur le robot, sauf que la méthode du couple calculé n'est pas programmable sur l'ordinateur de notre robot vu qu'elle nécessite un calculateur très performant pour pouvoir effectuer les calculs matriciels demandés. Les résultats obtenus indiquent que le PD d'ordre

fractionnaire fournit plus de flexibilité à la commande et donc il est toujours meilleur que le PD classique.

Dans un deuxième temps, on s'est intéressé par une autre commande robuste qui est la TDC ou bien "Time Delay Control". Cette commande est indépendante du modèle de système grâce à son estimateur TDE qui nécessite le bon choix d'un retard temporel dans lequel les signaux à estimer ne varient pas grandement. Les non-linéarités posent toujours des problèmes même avec ce genre de commandes, pour cela on a ajouté à la TDC un élément de robustesse qui est le TSM ou bien "Terminal Sliding Mode" avec un choix approprié de deux surfaces de glissement; la première est celle de Slotine et la deuxième est d'ordre fractionnaire proposée par [29]. Les trois lois de commande ont été simulées et implémentées sur le robot ISIR88 ce qui nous a permis de conclure que le mode glissant en général améliore la commande TDC en terme de performances et surtout en terme de robustesse; mais en comparant les résultats obtenus avec les deux surfaces de glissement on peut dire que la surface d'ordre fractionnaire est meilleure que celle de Slotine.

En guise de conclusion, notre travail présente une base pour les prochains projets du LCP touchant la commande d'ordre fractionnaire et la commande TDC. De plus, nous avons eu de la chance de pouvoir implémenter les commandes sur le robot ISIR88 et c'est ce qui a donné des résultats et des améliorations inattendus.

# Bibliographie

- [1] Reymond Clavel. Conception d'un robot parallèle rapide à 4 degrés de liberté. Technical report, EPFL, 1991.
- [2] Wisama Khalil and Etienne Dombre. *Modélisation identification et commande des robots*. Hermes, 1999.
- [3] Philippe Lelias. *Modélisation, identification et commande dynamique d'un robot d'architecture parallèle*, 2013.
- [4] Hamied Taha Dhia El Hak and Bakkouche Ala Eddine. Conception et réalisation d'un robot parallèle à trois degrés de liberté. *University of El Oued*, 2018.
- [5] M Rachedi. *Modélisation et Commande Robuste de Robots Parallèles : Application au Robot Delta*. PhD thesis, 2016.
- [6] Reymond Clavel. A fast robot with parallel geometry. In *Proc. Int. Symposium on Industrial Robots*, pages 91–100, 1988.
- [7] Damien Chablat and Philippe Wenger. A new three-dof parallel mechanism : milling machine applications. *arXiv preprint arXiv :0705.0982*, 2007.
- [8] H Nadour and N I Mokhtari. Contribution à la modélisation et à la commande du robot delta isir88. *Mémoire de projet de fin d'études, ENP*, 2016.
- [9] H I Kadri and W Hamouche. Modélisation et commande du robot delta isir88. *Mémoire de projet de fin d'études, ENP*, 2017.
- [10] André Olsson. Modeling and control of a delta-3 robot. *MSc Theses*, 2009.
- [11] Alain Codourey. Contribution à la commande des robots rapides et précis. Technical report, EPFL, 1991.
- [12] O Khelifi and L Azzoug. "modélisation et commande d'un robot parallèle de type delta". *Mémoire de projet de fin d'études, ENP*, 2010.
- [13] Alain Codourey. Dynamic modelling and mass matrix evaluation of the delta parallel robot for axes decoupling control. In *Proceedings of IEEE/RSJ international conference on intelligent robots and systems. IROS'96*, volume 3, pages 1211–1218. IEEE, 1996.
- [14] C E Boudjedir. *Contribution à la Commande par Apprentissage Itératif des Robots Parallèles de Type Delta. Théorie, Simulation et Expérimentation*. PhD thesis, 2019.

- [15] K Astrom and T Haggglund. Pid controllers : theory, design and tuning. Technical report, Research Triangle Park, North Carolina, USA, 1995.
- [16] Kenneth S Miller and Bertram Ross. *An introduction to the fractional calculus and fractional differential equations*. Wiley, 1993.
- [17] Concepción A Monje, YangQuan Chen, Blas M Vinagre, Dingyu Xue, and Vicente Feliu-Batlle. *Fractional-order systems and controls : fundamentals and applications*. Springer Science & Business Media, 2010.
- [18] Mohamed Charef. Commande d'ordre fractionnaire basée sur la "fractionalisation" des correcteurs classiques pid. 2014.
- [19] Jerzy Baranowski, Waldemar Bauer, Marta Zagórowska, Tomasz Dziwiński, and Paweł Piątek. Time-domain oustaloup approximation. In *2015 20th international conference on methods and models in automation and robotics (MMAR)*, pages 116–120. IEEE, 2015.
- [20] Ivo Petráš. *Fractional derivatives, fractional integrals, and fractional differential equations in Matlab*. IntechOpen, 2011.
- [21] Matlab Optimization Toolbox. User's guide. *The MathWorks, Inc.–2019.–3788*, 1997.
- [22] Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on optimization*, 13(3) :889–903, 2002.
- [23] C Mokraoui and A Bensaid. Modélisation et commande d'un robot de type delta de la simulation à l'expérimentation. *Mémoire de projet de fin d'études, ENP*, 2019.
- [24] Mohamed Bouri and Reymond Clavel. A windows pc based robot controller, an open architecture. In *International Symposium on Robotics*, volume 36, page 59. unknown, 2005.
- [25] Dihya Maincer, Moufid Mansour, Amar Hamache, Chemseddine Boudjedir, and Moussaab Bounabi. Switched time delay control based on artificial neural network for fault detection and compensation in robot manipulators. *SN Applied Sciences*, 3(4) :1–13, 2021.
- [26] Maolin Jin, Sang Hoon Kang, and Pyung Hun Chang. Robust compliant motion control of robot with nonlinear friction using time-delay estimation. *IEEE Transactions on Industrial Electronics*, 55(1) :258–269, 2008.
- [27] Maolin Jin, Jinh Lee, Pyung Hun Chang, and Chintae Choi. Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control. *IEEE Transactions on Industrial Electronics*, 56(9) :3593–3601, 2009.
- [28] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [29] Osman Eray and Sezai Tokat. The design of a fractional-order sliding mode controller with a time-varying sliding surface. *Transactions of the Institute of Measurement and Control*, 2020.