

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Automatique
Laboratoire de Commande des Processus



Mémoire de projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état
en Automatique.

**Modélisation et Commande d'un Robot de Type Delta
de la Simulation à l'Expérimentation.**

Présenté par

Abdeldjalil BENSAID

Cherif MOKRAOUI

Sous la direction de

Pr. Djamel BOUKHETALA

M. Chems Eddine BOUDJEDIR

Présenté et soutenu publiquement le 01/07/2019

Composition du Jury :

Président :	M. Rachid ILLOUL	Professeur	ENP
Rapporteurs :	M. Djamel BOUKHETALA	Professeur	ENP
	M. Chems Eddine BOUDJEDIR	Doctorant	ENP
Examineur :	M. Hachemi CHEKIREB	Professeur	ENP

ENP 2019

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique
Département d'Automatique
Laboratoire de Commande des Processus



Mémoire de projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état
en Automatique.

**Modélisation et Commande d'un Robot de Type Delta
de la Simulation à l'Expérimentation.**

Présenté par

Abdeldjalil BENSAID

Cherif MOKRAOUI

Sous la direction de

Pr. Djamel BOUKHETALA

M. Chems Eddine BOUDJEDIR

Présenté et soutenu publiquement le 01/07/2019

Composition du Jury :

Président :	M. Rachid ILLOUL	Professeur	ENP
Rapporteurs :	M. Djamel BOUKHETALA	Professeur	ENP
	M. Chems Eddine BOUDJEDIR	Doctorant	ENP
Examineur :	M. Hachemi CHEKIREB	Professeur	ENP

ENP 2019

نبذة مختصرة:

الهدف من هذه المذكرة هو النمذجة والتحكم في الروبوت دلتا **ISIR 88**. هذا الروبوت من صنف الأليات المتوازية، يتميز هذا النوع بالسرعة والدقة من جهة وصعوبة النمذجة من جهة أخرى. لهذا ستم نمذجته باعتماد فرضيات تبسيطية، لكن قد تنتج عنها أخطاء. تعتبر قوانين التحكم التي سيتم تصميمها (**PDF, TDC**) متينة. بحث تتيح التحكم في الروبوت على الرغم من وجود الأخطاء النمذجية. تتضمن الدراسة كلا من المحاكاة والتجريب.

الكلمات المفتاحية: روبوت دلتا، الروبوت المتوازي، تتبع مسار، نمذجة والتحكم في روبوتات الصناعية، متحكمات متينة، التحكم بالاعتماد على المعلومات السابقة (**TDC**) ، التحكم بمتحكمات ذات درجة كسرية (**PDF**) .

Abstract:

The aim of this work is the modeling and control of the delta robot ISIR 88. This robot is of parallel structure, it is known to be fast and precise. Its modeling is done here using simplifying assumptions which may introduce uncertainties. The control laws to be designed (fractional order PD and Time Delay Control) are robust, that make it possible to control the robot despite these uncertainties. Robustness tests to parametric variations are then made. The study involves both simulation and experimentation.

Keywords: Delta robot, Parallel robot, Trajectory tracking, Modeling and control of manipulator robots, Robust controls, Time Delay Control, Fractional Order controller.

Résumé :

Le but de ce projet de fin d'étude est la modélisation et commande du robot delta ISIR 88. Ce robot est de structure parallèle, il est connu pour être rapide et précis. Sa modélisation se fait à l'aide d'hypothèses simplificatrices d'où la possibilité d'introduction d'incertitudes. Les commandes à synthétiser (PD d'ordre fractionnaire et Time Delay Control) sont robustes et permettent de commander le robot en dépit de ces incertitudes. Des tests de robustesse aux variations paramétriques sont aussi faits. L'étude comporte la simulation et l'expérimentation.

Mots-clés : Robot delta, Robot Parallèle, Poursuite de Trajectoire, Modélisation et Commande des Robots manipulateurs, Commandes Robustes, Time Delay Control (TDC), Commande d'Ordre Fractionnaire.

Dédicaces

A nos chers parents.

A nos frères et sœurs.

A nos familles.

A tous nos amis.

Remerciements

Nous remercions Dieu pour nous avoir donné la force et le courage nécessaires pour accomplir ce travail qui couronne nos études.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrions témoigner toute notre reconnaissance.

Notre sincère gratitude s'adresse particulièrement à nos deux promoteurs Pr. Djamel BOUKHETALA et M. Chems Eddine BOUDJEDIR pour avoir proposé le projet, et nous avoir encadré et orienté tout le long de ce travail.

Nous remercions vivement les membres du jury, Pr. Hachemi CHEKIREB et Pr. Rachid ILLOUL pour avoir accepté de faire partie du jury et pour les conseils et corrections apportés.

Nous tenons à remercier tous les enseignants qui nous ont prodigué des connaissances dont la valeur est incontestable.

Nous sommes reconnaissants à toute personne qui nous a aidé de près ou de loin à l'exemple de notre ami Boudjidj Abd El Ghani.

Nous remercions nos très chers parents qui ont toujours été là pour nous aider et encourager. Leur soutien inconditionnel depuis toujours nous a permis d'arriver là où nous sommes.

Table des Matières

Liste des Tableaux	
Liste des Figures	
Liste des Abréviations	
Introduction générale.....	13
Chapitre 1 Robot Delta.....	15
1.1 Introduction	16
1.2 Généralités sur la robotique.....	16
1.2.1 Le robot et ses catégories	18
1.2.2 Caractérisation des robots industriels.....	19
1.2.3 Classes des robots manipulateurs	19
1.2.4 Modélisation.....	21
1.2.5 Génération des trajectoires	23
1.2.6 Commande des robots	25
1.3 Robot Delta	26
1.3.1 Applications du robot Delta	27
1.3.2 Constituants et description du Delta.....	28
1.3.3 Robot Delta ISIR 88	30
1.4 Conclusion.....	34
Chapitre 2 Modélisation du Robot Delta.....	35
2.1 Introduction	36
2.2 Modélisation géométrique.....	36
2.3 Modélisation cinématique	38
2.4 Accélération	40
2.5 Modélisation dynamique	40
2.6 Conclusion.....	43
Chapitre 3 Commandes Robustes du Robot Delta	44
3.1 Introduction	45

Table des Matières

3.2	Commande par PD et PD d'ordre fractionnaire	46
3.2.1	Commande par PD	46
3.2.2	Commande PD d'ordre fractionnaire	47
3.2.2.1	Généralités sur le calcul fractionnaire	47
3.2.2.2	Régulateur PD d'ordre fractionnaire	50
3.3	Commande Time delay control (TDC)	53
3.3.1	Structure de la TDC.....	54
3.3.2	Technique d'estimation TDE	56
3.3.3	Synthèse de la loi de commande	60
3.3.3.1	TDC approche classique.....	60
3.3.4.2	TDC avec un élément de robustesse	66
3.4	Tests de Robustesse.....	74
3.5	Conclusion.....	76
Chapitre 4 Expérimentation		77
4.1	Introduction	78
4.2	Considérations pratiques	78
4.3	Environnement d'implémentation.....	79
4.4	Trajectoire de référence implémentée	79
4.5	L'implémentation des lois de commande.....	80
4.6	résultats expérimentaux.....	81
4.6.1	Commande PD	81
4.6.2	Commande PD d'ordre fractionnaire	83
4.6.3	Commande TDC avec PD	85
4.6.4	Commande TDC avec mode de glissement (surface de Slotine)	86
4.6.5	Commande TDC avec mode de glissement (surface de Jin).....	89
4.6.6	Tests de robustesse	91
4.7	Conclusion.....	94

Table des Matières

Conclusion Générale	95
Références Bibliographiques.....	97

Liste des Tableaux

Tableau	Titre
Tableau 1.1	Comparaison des deux structures
Tableau 1.2	Longueurs paramétriques
Tableau 1.3	Angles caractéristiques.
Tableau 1.4	Paramètres du volume de travail
Tableau 4.1	Les paramètres des régulateurs
Tableau 4.2	Résultats des tests de robustesse aux variations paramétriques

Liste des Figures

Figure	Titre
Fig. 1.1	Caractère interdisciplinaire de la robotique
Fig. 1.2	Canard de Jacques de Vaucanson
Fig. 1.3	Unimate, premier robot industriel
Fig. 1.4	Des robots de différentes catégories
Fig. 1.5	Classes des robots manipulateurs
Fig. 1.6	Exemples de robot sériel et parallèle
Fig. 1.7	Schéma de modélisation géométrique
Fig. 1.8	Schéma de modélisation dynamique
Fig. 1.9	Trajectoire de référence de type prise-dépose
Fig. 1.10	Schéma synoptique de la commande en boucle ouverte de robot manipulateur
Fig. 1.11	Schéma synoptique de la commande en boucle fermée de robot manipulateur
Fig. 1.12	Un des premiers prototypes du Delta
Fig. 1.13	Delta et son inventeur à droite
Fig. 1.14	Différentes réalisations du Delta
Fig. 1.15	Quelques applications du robot Delta
Fig. 1.16	Schéma de principe du robot Delta
Fig.1.17	Photo du Delta ISIR 88 du LCP
Fig. 1.18	Vue détaillée de la courroie
Fig. 1.19	Représentation schématique du Delta avec ses paramètres
Fig. 1.20	Le volume de travail du robot Delta
Fig. 1.21	Singularités du robot Delta
Fig. 2.1	Schéma synoptique des modèles géométriques directes et inverses
Fig. 2.2	Géométrie du Delta
Fig. 2.3	Les rotations entre repères
Fig. 2.4	La structure avant et après simplification
Fig.3.1	Schéma de commande du robot DELTA avec régulateur PD
Fig.3.2	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur PD
Fig. 3.3	Structure d'un régulateur PD d'ordre fractionnaire

Liste des Figures

Figure	Titre
Fig. 3.4	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur PD fractionnaire
Fig. 3.5	Structure de commande TDC
Fig. 3.6	L'erreur d'estimation de $N(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ par la technique TDE
Fig. 3.7	Structure de commande TDC simple basée sur servo-contrôleur PD
Fig. 3.8	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-PD
Fig. 3.9	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-PD d'ordre fractionnaire
Fig. 3.10	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-SM avec surface de Slotine et fonction signe.
Fig. 3.11	Erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-SM avec surface de Slotine et fonction sigmoïde.
Fig. 3.12	Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-TSMC avec surface de Jin
Fig. 3.13	La variations des RMS de l'erreur en fonction de masse de charge
Fig. 3.14	La variations des RMS du couple (commande) en fonction de masse de charge
Fig. 4.1	Trajectoire de référence
Fig.4.2	Résultats expérimentaux montrant la trajectoire effectuée et les erreurs de poursuite articulaires dans le cas du régulateur PD
Fig.4.3	Résultats expérimentaux montrant les commandes en couple du PD
Fig.4.4	Comparaison des erreurs de poursuite articulaires de PD et PDF
Fig.4.5	Comparaison entre les commandes en couple de PD et PDF
Fig.4.6	Comparaison entre PD et TDC (PD) : les erreurs de poursuite articulaires
Fig.4.7	Comparaison entre les commandes en couple de PD et TDC (PD).
Fig.4.8	Comparaison des erreurs de poursuite articulaires entre TDC(PD) et TDC (SM, Slotine):
Fig.4.9	Comparaison des commandes en couple entre TDC(PD) et TDC (SM, Slotine)
Fig.4.10	Comparaison entre les erreurs de poursuite articulaires de TDC (SM, Slotine) et TDC (SM, Jin) et trajectoire effectuée TDC (SM, Jin)

Liste des Figures

Figure	Titre
Fig.4.11	Comparaison des commandes en couple entre TDC (SM, Slotine) et TDC (SM, Jin)
Fig.4.12	Erreurs de poursuite articulaires pour les différents essais en masses (TDC-JIN)
Fig. 4.13	Couples pour les différents essais en masses (TDC-JIN)

Liste des Abréviations

Acronyme	Signification
BF	Boucle Fermée
BO	Boucle Ouverte
CRONE	Commande Robuste d'Ordre Non Entier
ENP	Ecole Nationale Polytechnique
EPFL	Ecole Polytechnique Fédérale de Lausanne
LCP	Laboratoire de commande de processus
MCD	Modèle Cinématique Directe
MCI	Modèle Cinématique Inverse
MDD	Modèle Dynamique Direct
MDI	Modèle Dynamique Inverse
MGD	Modèle Géométrique Direct
MGI	Modèle Géométrique Inverse
PD	Proportionnel Dérivée
PDF	Proportionnel Dérivée d'ordre Fractionnaire
PFE	Projet de Fin d'Etude
PID	Proportionnel Intégrale Dérivée
RMS	Root Mean Square
SM	Sliding Mode
SMC	Sliding Mode Control
TDC	Time Delay Control
TDE	Time Delay Estimation
TSMC	Terminal Sliding Mode Control

Introduction générale

La robotique est l'une des branches des sciences et techniques qui a connu dernièrement des avancées impressionnantes grâce aux développements réalisés dans les domaines auxquels elle fait recours à l'instar de l'informatique, l'intelligence artificielle, l'électronique et la théorie de la commande. Elle a envahi tous les secteurs, a changé le mode de vie des êtres humains sur tous les plans et devient de plus en plus un facteur de puissance incontournable chez les états maîtrisant sa technologie. D'où l'intérêt croissant porté par les laboratoires de recherche à cette science pointue.

Dans le cadre du projet de fin d'étude et pour les raisons qu'on vient d'évoquer, nous avons choisi de travailler sur la modélisation, la commande, la simulation et l'expérimentation du robot manipulateur Delta ISIR 88 du Laboratoire de Commande des Processus de l'Ecole Nationale Polytechnique d'Alger. Ce robot est très utilisé en industrie pour les multiples qualités dont il jouit et qui sont incontestablement très convoitées par les industriels à l'exemple de sa rapidité et sa haute précision. Ses qualités viennent principalement comme nous allons voir de sa structure parallèle particulière qui démontre le génie de son concepteur Reymond Clavel. Ce robot est utilisé dans un très large spectre d'applications allant du conditionnement des produits aux opérations de très haute précision comme la télé-chirurgie.

Les travaux ayant le même thème que le nôtre foisonne. Toutefois, notre travail se focalise essentiellement sur le problème de commande robuste du robot Delta. Le présent mémoire suit le plan de travail suivant :

Au chapitre 1, en premier lieu seront traitées des généralités sur la robotique et sur les robots manipulateurs industriels, commençant par une brève histoire, puis une classification et étude des principales caractéristiques des robots industriels et enfin une introduction à la modélisation, la génération des trajectoires de référence et la commande des robots. En second lieu, on s'intéresse au robot Delta, sa description et ses caractéristiques en général puis au robot ISIR 88 en particulier.

Quant au chapitre 2, on y abordera la modélisation géométrique, cinématique et dynamique du Delta ISIR 88. Cette dernière modélisation se fera sous des hypothèses simplificatrices vue la complexité de la structure du Delta.

Au chapitre 3, on s'intéressera au problème de commande robuste du robot en question. Pour ce faire, plusieurs commandes ont été proposées, simulées, analysées et comparées. Ces

commandes sont le PD (Proportionnel dérivée) classique, le PD d'ordre fractionnaire et la commande TDC (Time Delay Control) sous plusieurs formes.

Enfin, le chapitre 4 portera sur l'expérimentation des commandes sur le robot réel disponible au LCP (Laboratoire de Commande des Processus) et à l'analyse des résultats obtenus.

Chapitre 1

Robot Delta

1.1 Introduction

Ce chapitre est divisé en deux principales parties. Dans la première, nous allons aborder des généralités sur la robotique, elle comportera une brève introduction à ce fascinant domaine et traitera les principaux sujets auxquelles la robotique fait appel d'une manière assez générale. Dans la seconde partie, nous allons introduire le robot Delta et son histoire, ensuite traiter ses applications en industrie, puis le décrire et donner ses constituants et enfin nous allons aborder les spécificités du robot sur lequel porte notre étude à savoir le Delta ISIR 88 se trouvant au sein du laboratoire de commande de processus (LCP) de l'école nationale polytechnique d'Alger (ENP).

1.2 Généralités sur la robotique

De nos jours, la robotique qui était il n'y a pas longtemps pur concept de science-fiction occupe une place importante et indéniable dans la vie de l'homme et continue à la changer sur plusieurs plans (socio-économique surtout). Elle est intrinsèquement interdisciplinaire et fait appel aux connaissances et expertises d'une large panoplie de domaines tels que l'automatique, la mécanique, l'électronique, l'informatique et bien d'autres (figure 1.1).

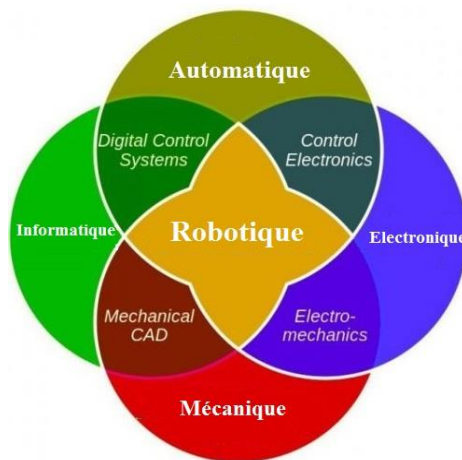


Fig. 1.1 Caractère interdisciplinaire de la robotique

Le terme robot tire son étymologie du tchèque "robota" qui signifie "travaux forcés", "corvées" ou "servitude". Le premier à l'avoir introduit était l'écrivain tchèque Karel Capek dans la pièce de théâtre R. U. R. (Rossum's Universal Robots) en 1920 pour désigner des serviteurs de métal. Une autre figure emblématique qui a marqué les débuts de la robotique (en tant que concept) est incontestablement l'écrivain Isaac Asimov. Il est connu pour ses abondantes œuvres de science-fiction portant sur le sujet où il a établi les fameuses lois de l'éthique de la robotique.

La robotique a connu depuis ses débuts beaucoup d'évolution. Elle est marquée par des dates dont nous allons citer quelques-unes :

1739 : Le canard de Jacques de Vaucanson (figure 1.2) ;

1774 : Les trois automates de Pierre Jaquet-Droz ;

1801 : Le métier à tisser Jacquard ;

1920 : Invention du mot « robot » (pièce de théâtre de Karel Capek) ;

1946 : ENIAC, le premier ordinateur ;

1950 : Les robots (I, Robot), recueil de nouvelles d'Isaac Asimov (les trois lois de la robotique);

1956 : Conférence sur l'intelligence artificielle ;

1961 : Unimate, premier robot industriel aux Etats-Unis (figure 1.3) ;

1967 : Shakey, premier robot mobile contrôlé par ordinateur, Stanford Research Institute ;

1973 : Wabot-1, premier robot humanoïde à marche bipède, université de Waseda;

1993: Cog, Humanoid Robotics Group, MIT (Massachusetts Institute of Technology – Etats-Unis);

1999: AIBO - ERS-110, Sony;

2000: Asimo, Honda;

2003 : Les rovers Spirit et Opportunity sur Mars, NASA ;

2004 : Le robot serpent amphibie (Japon) ;

2005 : AIBO Mind 3 – ERS-7, Sony ;

2018 : Le robot humanoïde Atlas de Boston Dynamics fait du "parkour " .

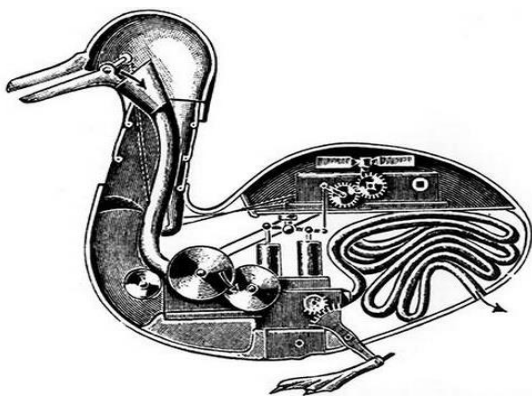


Fig. 1.2 Canard de Jacques de Vaucanson



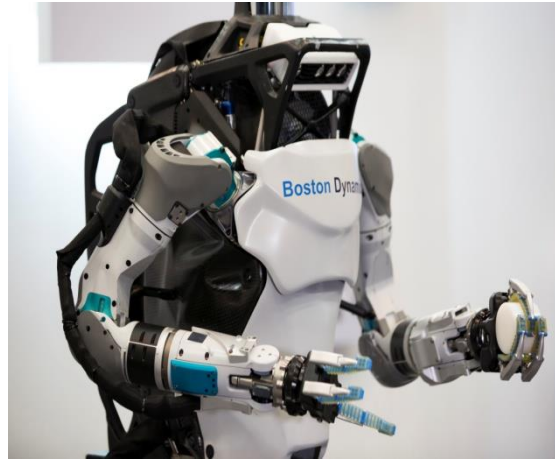
Fig. 1.3 Unimate, premier robot industriel

1.2.1 Le robot et ses catégories

Le robot est un système mécanique poly-articulé, actionné par le biais de moteurs et contrôlé via un ordinateur. Il effectue des tâches très variées s'avérant pénibles ou ennuyeuses pour l'humain du fait de leur caractère répétitif ou de l'hostilité du milieu. Cette définition englobe plusieurs catégories de robots, notamment les Robots mobiles, Robots sous-marins, Robots volants, Robots humanoïdes et les robots manipulateurs industriels.



a- Le robot mobile Rover



b - L'humanoïde de Boston Dynamics



c- Robot volant UAV



d- Bras manipulateur Kuka

Fig. 1.4 Des robots de différentes catégories

Dans tout ce qui suit nous n'allons considérer qu'une classe particulière à savoir les robots industriels. Un robot industriel est un manipulateur automatique asservi en position, polyvalent, reprogrammable, capable de positionner et d'orienter des matériaux, des pièces, des outils ou des dispositifs spécialisés au cours de mouvements variables et programmés pour l'exécution de tâches variées.

1.2.2 Caractérisation des robots industriels

- **Le Volume de travail**

Désigne l'ensemble des points de l'espace accessibles par l'effecteur du robot. Il dépend principalement de la géométrie du robot ainsi que des longueurs des segments. Un volume de travail grand signifie plus d'interaction avec l'environnement extérieur. C'est donc l'une des caractéristiques les plus importantes à prendre en considération lors du choix du robot pour effectuer une certaine tâche.

- **La Répétabilité**

La répétabilité d'un robot est l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail.

- **Performances dynamiques**

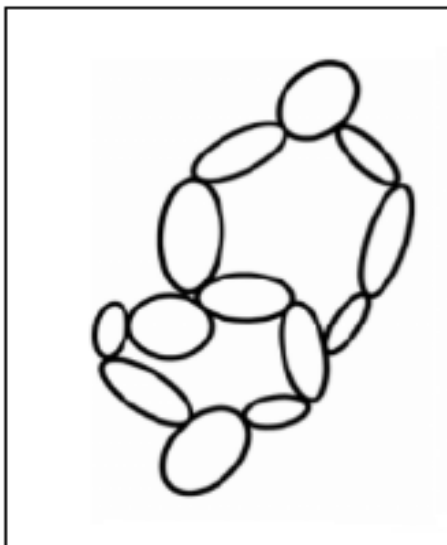
- Vitesse maximale de translation ou de rotation de chaque axe.
- Accélération maximale : Est donnée pour chaque axe dans la configuration la plus défavorable (inertie maximale, charge maximale).

- **Charge utile**

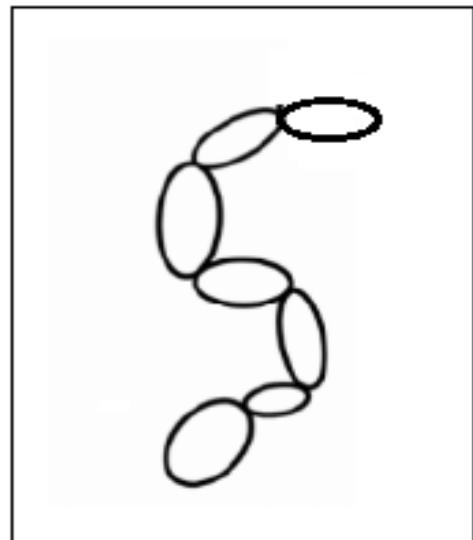
Désigne la charge maximale que peut porter le robot sans dégrader les performances dynamiques et la répétabilité.

1.2.3 Classes des robots manipulateurs

Il existe deux grandes familles de robots manipulateurs : Les manipulateurs sériels et les manipulateurs parallèles (figure 1.5).



a- robot parallèle

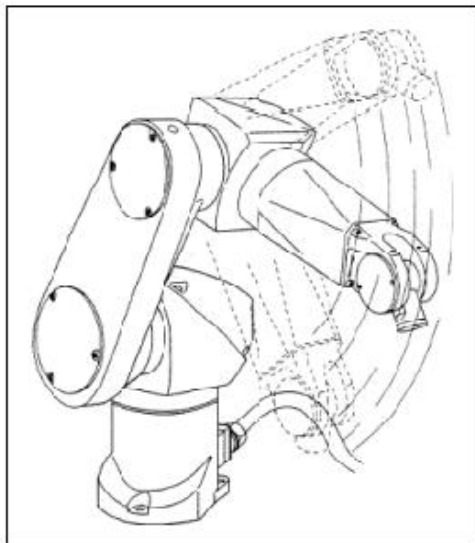


b- robot sériel

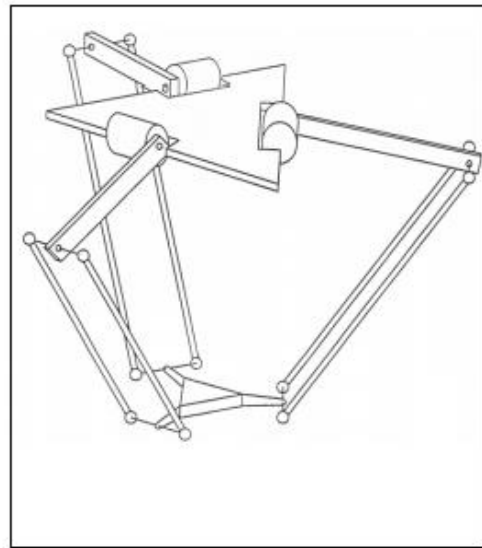
Fig. 1.5 Classes des robots manipulateurs

Robots sériels : Cette classe des robots est caractérisée, du point de vue architecture, par le fait qu'une seule chaîne cinématique relie la base à l'effecteur. La chaîne cinématique étant une succession ouverte de segments et d'articulations.

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 ; on parle également de cinématique bouclée. Certains robots peuvent être une combinaison de cinématiques sérielles et parallèles ; on parle dans ce cas de robots hybrides [1].



a- Staubli RX (sériel)



b- Delta (parallèle)

Fig. 1.6 Exemples de robot sériel et parallèle

Comparaison entre les deux structures

Le tableau 1.1 montre les principales différences entre les deux structures.

Caractéristique	Robot Sériel	Robot parallèle
Volume du travail	Grand	Petit et complexe
Erreur de position	Accumulative	Moyenne
Modélisation	Simple	Relativement complexe
Rigidité	Basse	Haute
Charge maximale	Petite	Grande
Vitesse et accélération	Petites	Grandes
Précision	Petite	Grande
Calibration	Simple	Complexe
Rapport volume de travail/taille du robot	Grande	Petite

Tableau 1.1 Comparaison des deux structures

1.2.4 Modélisation

- **Modélisation géométrique**

Le modèle géométrique permet d'exprimer la situation de l'organe terminal (espace opérationnel) en fonction des coordonnées articulaires (espace articulaire) et inversement.

Le modèle géométrique direct (MGD) du robot peut être représenté par la relation :

$$X = f(q) \quad (1.1)$$

q étant le vecteur des variables articulaires tel que :

$$q = [q_1 q_2 \dots q_n]^T \quad (1.2)$$

X est le vecteur des coordonnées opérationnelles définies par :

$$X = [x_1 x_2 \dots x_m]^T \quad (1.3)$$

Le modèle géométrique inverse (MGI) s'exprime ainsi :

$$q = f^{-1}(X) \quad (1.4)$$

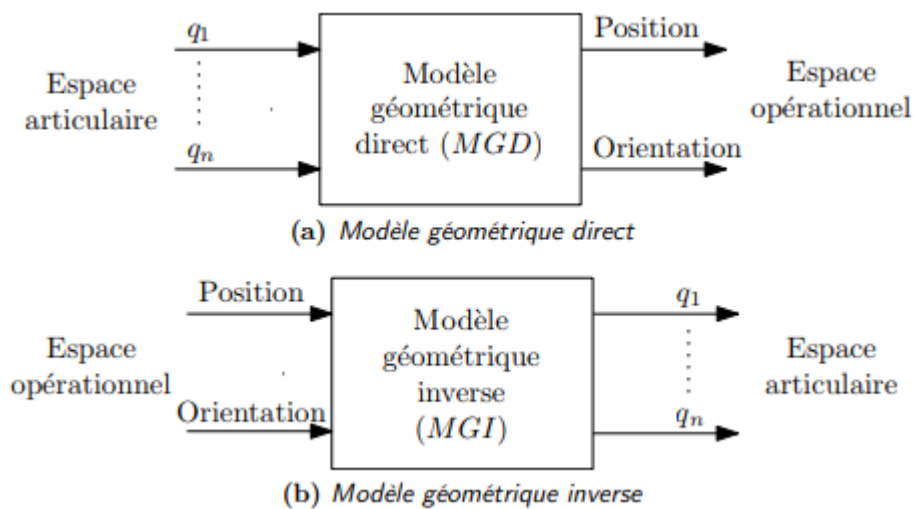


Fig. 1.7 Schéma de modélisation géométrique. [8]

- **Modélisation cinématique**

Elle a pour objet l'établissement des modèles exprimant la relation entre les vitesses articulaires et les vitesses opérationnelles. Elle fait intervenir généralement une matrice J dite jacobienne. Elle s'exprime ainsi : Dans le cas direct :

$$\dot{X} = J \cdot \dot{q} \quad (1.5)$$

Et dans le cas inverse :

$$\dot{q} = J^{-1} \cdot \dot{X} \quad (1.6)$$

Avec :

\dot{X} : vecteur des vitesses opérationnelles ;

\dot{q} : vecteur des vitesses articulaires ;

L'étude de la matrice jacobienne J permet d'avoir les singularités du robot ainsi que d'autres importantes informations [1].

- **Modélisation dynamique**

Cette modélisation, quant à elle, est la plus importante du moment qu'elle occupe une place cruciale et incontournable en commande. Elle donne la relation entre les accélérations, vitesses et positions des articulations, et les couples développés par les moteurs donc la dynamique du robot en fonction des tensions fournies aux entraînements . En effet, en commande on agit sur les tensions à fournir pour imposer au robot une certaine dynamique désirée. Cette dernière technique sera abordée plus en détail dans la suite du mémoire.

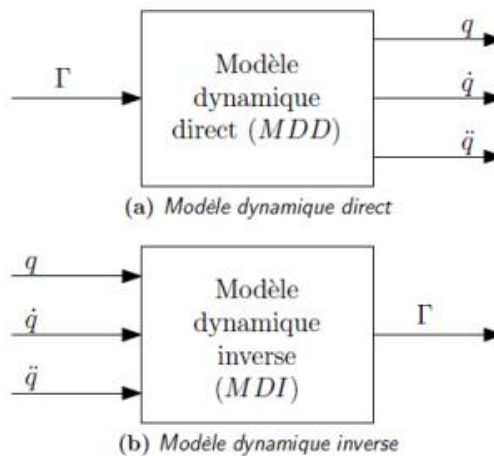


Fig. 1.8 Schéma de modélisation dynamique [8]

Le modèle dynamique s'obtient en faisant recours aux formalismes de mécanique suivants : Le formalisme de Lagrange-Euler ainsi que les équations de Newton-Euler. Le modèle dynamique des robots manipulateurs peut généralement s'exprimer de la sorte :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_d \quad (1.7)$$

Avec :

q, \dot{q}, \ddot{q} : les vecteurs position, vitesse et accélération articulaires respectivement.

τ : le vecteur des couples appliqués ;

$M(q)$: la matrice d'inertie du robot ;

$C(q, \dot{q})$: la matrice des forces centrifuges et de Coriolis ;

$G(q)$: vecteur des forces gravitationnelles ;

τ_d : vecteur des couples de perturbation.

1.2.5 Génération des trajectoires

Dans une tâche de robotique industrielle, L'effecteur du robot doit impérativement suivre une trajectoire spécifique. Cette dernière définissant un chemin le long duquel l'effecteur du robot doit se déplacer. Ce chemin peut être défini par une courbe géométrique ou une séquence de points définie soit en coordonnées opérationnelles (coordonnées de l'effecteur), soit en coordonnées articulaires. Le problème de la génération de trajectoire consiste à calculer, pour le système de commande, les variables d'articulation ou d'effecteur final de référence souhaitées en fonction du temps, de sorte que le robot suive le trajet souhaité.

Les trajectoires d'un robot peuvent être classées comme suit :

- trajectoire entre deux points avec un chemin libre entre eux ;
- trajectoire entre deux points via une séquence de points intermédiaires souhaités, également appelés points de passage, avec des chemins libres entre ces points de passage ;
- trajectoire entre deux points séparés par un chemin contraint (droit segment de ligne par exemple) ;
- trajectoire entre deux points via des points intermédiaires avec des chemins contraints entre ces points de passage.

Pour les deux premières classes, la trajectoire est généralement créée dans l'espace articulaire. Dans les deux dernières classes, il est généralement préférable de générer la trajectoire dans l'espace opérationnel.

La génération de trajectoires dans l'espace opérationnel permet de prédire la géométrie du chemin donc de s'assurer que les consignes restent dans l'espace de travail du robot. Elle représente toutefois un certain nombre d'inconvénients comme la nécessité de conversions en espace articulaire via le modèle géométrique inverse dans la consigne de commande ainsi que la possibilité de passage par des configurations singulières.

Ces trajectoires s'expriment dans la quasi-totalité des cas par des fonctions mathématiques à base d'interpolation polynomiale. En l'occurrence, la trajectoire prise-dépose avec laquelle nous allons travailler dans la suite s'exprime à l'aide d'un polynôme de degré cinq dont l'équation est:

$$r(t) = 10 \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \quad (1.8)$$

Où t_f est le temps écoulé entre la position initiale et finale de la variable opérationnelle.

Les variables opérationnelles sont données par

$$\left\{ \begin{array}{l} x(t) = x_i + \Delta x r(t) \quad 0 \leq t \leq t_f \\ y(t) = y_i + \Delta y r(t) \quad 0 \leq t \leq t_f \\ \left\{ \begin{array}{l} z(t) = z_i + \Delta z r(t) \quad 0 \leq t \leq \frac{t_f}{2} \\ z(t) = z_f - \Delta z r(t) \quad \frac{t_f}{2} \leq t \leq t_f \end{array} \right. \end{array} \right. \quad (1.9)$$

Avec $\Delta x = x_f - x_i$; $\Delta y = y_f - y_i$ et $\Delta z = z_f - z_i$;

Et :

(x_i, y_i, z_i) : La position initiale de l'organe terminal dans l'espace opérationnel.

(x_f, y_f, z_f) : La position finale de l'organe terminal dans l'espace opérationnel.

Le choix de ce polynôme est justifié par la nécessité d'assurer la continuité des accélérations pour les robots à grande vitesse ou transportant des charges importantes afin d'éviter d'exciter la mécanique [1].

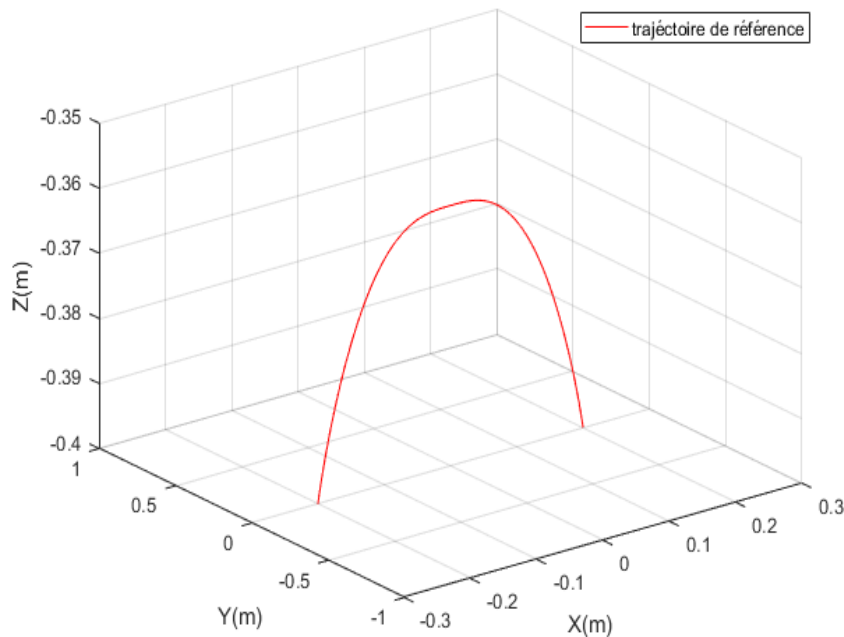


Fig. 1.9 Trajectoire de référence de type prise-dépose

1.2.6 Commande des robots

La quasi-totalité des applications qu'effectuent les robots industriels fait surgir la nécessité de résoudre un problème de poursuite de trajectoire. Ce dernier est l'un des problèmes les plus classiques auxquels s'intéresse la théorie de commande (automatique). Pour exécuter une certaine tâche, l'effecteur du robot, qui est son élément opérationnel, doit suivre une trajectoire bien définie. Cette trajectoire dans l'espace opérationnel se traduit par des trajectoires dans l'espace articulaire par l'intermédiaire du modèle géométrique inverse. Nous avons uniquement le control des couples développés par les moteurs (actionneurs) via des tensions à fournir. Cependant, les couples sont reliés aux positions, vitesses et accélérations des articulations par le modèle dynamique. Nous allons donc agir sur les tensions à fournir (couples des moteurs) appelées commandes pour imposer une certaine dynamique au robot (le système à commander) et par conséquent son effecteur suit la trajectoire désirée. Pour ce faire, il existe deux grandes classes de techniques : La commande en boucle ouverte et la commande en boucle fermée.

Commande en boucle ouverte BO

Cette commande se distingue par sa simplicité du moment qu'elle ne requiert pas un retour d'information sur l'état du robot. Toutefois, elle souffre de beaucoup de problèmes en performances, elle est incapable de compenser les perturbations et ne fonctionne pas s'il y a erreur de modélisation. Elle est donc rarement utilisée en robotique.

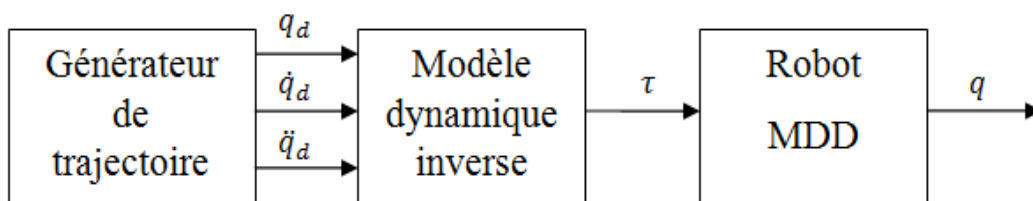


Fig. 1.10 Schéma synoptique de la commande en boucle ouverte de robot manipulateur

Commande en boucle fermée BF

Contrairement à la commande précédente, celle-ci prend en considération l'état du robot, obtenu via des capteurs, lors de la synthèse de la loi de commande. Ce retour d'information s'appelle feedback et permet de contourner les problèmes dont souffre la commande en boucle ouverte. Elle est, dès lors très utilisée en robotique. La figure 1.11 en est une représentation graphique.

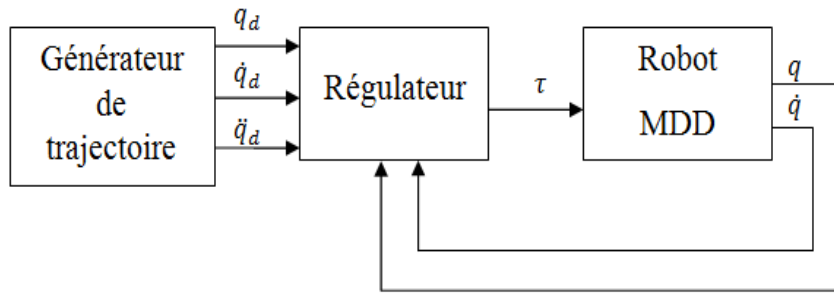


Fig. 1.11 Schéma synoptique de la commande en boucle fermée de robot manipulateur

Le régulateur chargé de calculer la commande peut être analogique (à base d'amplificateurs opérationnels généralement) ou numérique (en utilisant un ordinateur doté d'un processeur). Toutefois, la tendance ces derniers temps est vers le numérique vu les avantages qu'il offre notamment la possibilité de reprogrammation et la simplicité d'implémentation.

Il existe plusieurs méthodes pour la synthèse de la loi de commande du régulateur. On cite :

- La commande PD (Proportionnel Dérivée)
- La commande par couple calculé (Computed Torque) : commande linéarisante;
- La commande Backstepping;
- La commande par mode glissant ;
- La commande adaptative ;
- La commande H infini ;
- La commande TDC (Time Delay Control);
- La commande d'ordre fractionnaire.

1.3 Robot Delta

Le robot Delta montré dans la figure 1.12 est un robot manipulateur de structure parallèle (ayant des chaînes cinématiques fermées) inventé par le professeur Reymond Clavel (figure 1.13) de l'École Polytechnique Fédérale de Lausanne (EPFL) en 1985.

En effet, après une visite avec son équipe du Laboratoire de Systèmes Robotiques à une usine de chocolat, l'idée d'un robot assurant des tâches de prise-dépose à grande vitesse et à faible charge lui est venue à l'esprit. Un travail de recherche fut ensuite couronné par l'invention du robot en question. Une invention qui lui a valu la réception en 1999 du Golden Robot Award, parrainé par ABB Flexible.



Fig. 1.12 Un des premiers prototypes du Delta

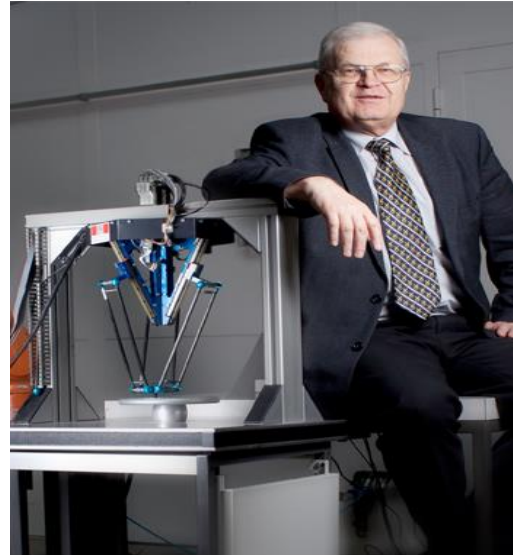


Fig. 1.13 Delta et son inventeur à droite

Le premier brevet fut accordé à R. Clavel en 1985, puis il a été vendu en 1988 à la compagnie suisse Demarex SA (actuellement Bosch Packaging Technology Unit, Romanel). En 2007, le brevet s'ouvre au domaine public et dès lors plus de 100 entreprises ont adopté cette structure.



a- ABB FlexPicker



b- Delta de Yaskama



c- Delta de Omron

Fig. 1.14 Différentes réalisations du Delta.

Constamment amélioré et perfectionné, le robot Delta a gagné au fil des années une véritable domination dans l'industrie, on en compte environ 10000 dans le monde.

1.3.1 Applications du robot Delta

Le robot Delta est utilisé dans un très large spectre d'applications. En plus des applications de conditionnement où il excelle, Il se distingue aujourd'hui dans la micro position, télé-chirurgie, l'usinage de matériaux, l'horlogerie, l'impression 3D et bien d'autres.



a- Delta effectuant la prise-dépose



b- Delta dans l'impression 3D



c- Delta en chirurgie



d- Delta en conditionnement

Fig. 1.15 Quelques applications du robot Delta

1.3.2 Constituants et description du Delta

La figure 1.16 sert de support pour la description du Delta. Ce robot est constitué d'une "base" fixe (1) et d'une plaque mobile (5) désignée sous le nom "nacelle". La liaison entre ces deux constituants est faite par le biais de trois chaînes cinématiques identiques. Chaque chaîne est constituée d'un "bras" (2) et deux "barres parallèles" (3). Le bras est lié de son extrémité supérieure à la base via une articulation rotoïde. Les deux barres parallèles munies chacune d'articulation rotule (4) les reliant avec la nacelle. L'ensemble des deux barres parallèles ainsi que ses deux éléments de liaison au bras et à la nacelle est nommé "parallélogramme". Chaque bras est entraîné à l'aide d'un moteur (7) fixé à la base. Le préhenseur (10) peut suivant la nécessité être monté sur la nacelle et relié au moteur (6) par l'intermédiaire de l'arbre télescopique (8) doté d'une articulation cardan (9) [21].

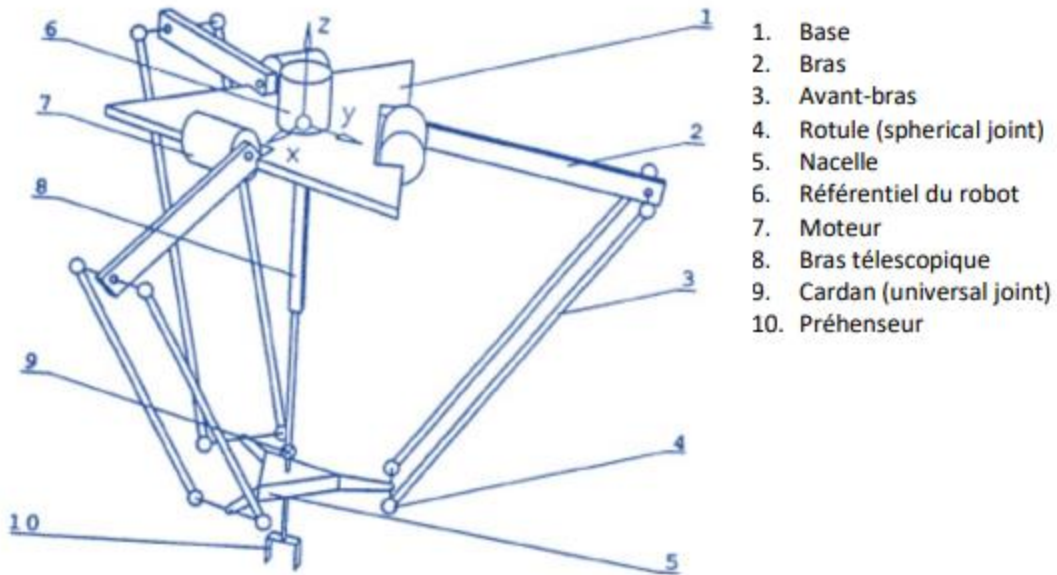


Fig. 1.16 Schéma de principe du robot Delta

Le delta est muni de trois degrés de liberté de translation (XYZ) plus un quatrième degré de rotation (thêta). Du moment que ce dernier est assuré par un mécanisme indépendant qu'on peut amputer du robot et du fait qu'on s'intéresse dans ce travail au problème de suivi de trajectoire, on ne prendra pas en considération la rotation, autrement dit on se contentera de mouvements dans l'espace opérationnel qui peuvent être décrits uniquement par les translations (XYZ).

L'idée ingénieuse dans la conception du robot parallèle Delta est l'utilisation de parallélogrammes. Un parallélogramme permet à une liaison de sortie de rester dans une orientation fixe. L'utilisation de trois parallélogrammes de ce type restreint complètement l'orientation de la plate-forme mobile (nacelle), qui ne conserve que trois degrés de liberté de pure translation. Les articulations des trois parallélogrammes qui sont montées sur les bras sont sans actionnement (passives).

Les articulations rotoïdes des bras tournants sont actionnés de deux manières différentes : avec des moteurs rotatifs (servomoteurs CC ou CA) ou avec des actionneurs linéaires. Ceci fait qu'il existe deux principales catégories de robots Delta : Le delta linéaire et le delta rotatif.

Une autre caractéristique importante du Delta est le fait de monter ses actionneurs sur la base qui supporte dès lors leurs poids et d'utiliser des liaisons de faible masse permettant ainsi à la plate-forme mobile (nacelle) d'accéder des accélérations jusqu'à 50 G dans des environnements expérimentaux et 12 G dans des applications industrielles.

1.3.3 Robot Delta ISIR 88

Dans la suite nous allons nous intéresser au robot Delta ISIR 88 qui se trouve au laboratoire de commande des processus (LCP) de l'école nationale polytechnique (ENP) d'Alger et qui représente l'objet de notre étude. Il est, comme le montre la figure 1.17, contrôlé par l'intermédiaire d'un ordinateur.



Fig.1.17 Photo du Delta ISIR 88 du LCP

Ses entrainements sont des moteurs à courant continu brushless, il se range donc dans la classe des Deltas rotatifs (non linéaires). Ce robot a la particularité d'être doté au niveau de chacun des trois moteurs d'un réducteur à courroie dentée (figure 1.18).

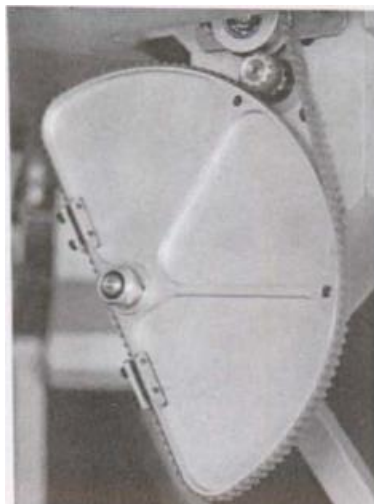


Fig. 1.18 Vue détaillée de la courroie

La transmission entre le moteur et le bras n'est donc pas directe, un coefficient de réduction 12:1 est son effet (couple au niveau du bras= $12 \times$ couple au niveau du moteur) .

• Longueurs paramétriques et angles caractéristiques

La figure 1.19 montre une représentation schématique d'une chaîne cinématique parmi les trois existantes et qui sont identiques :

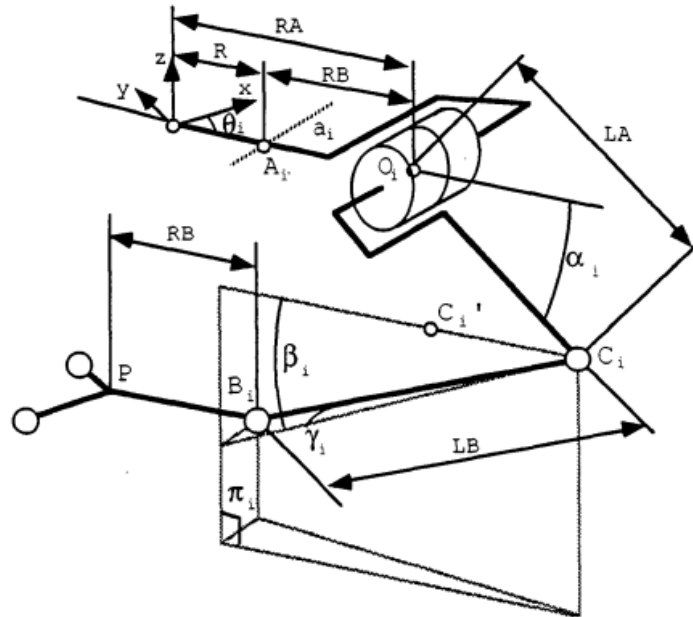


Fig. 1.19 Représentation schématique du Delta avec ses paramètres [21]

Longueurs paramétriques

Paramètre	Description	Valeur (ISIR 88)
L_A	Longueur du bras	200 mm
L_B	Longueur d'une barre parallèle	380 mm
R_A	La distance entre le centre de la base fixe et l'axe de rotation du bras	150 mm
R_B	Distance entre le centre de la nacelle et le côté du parallélogramme.	25 mm
R	$R_A - R_B$	125 mm

Tableau 1.2 Longueurs paramétriques

Angles caractéristiques :

Paramètre	Description
α_i	Angle entre le $i^{\text{ème}}$ bras et le plan de la base fixe, il sera considéré positif lorsque le bras est situé du côté de la nacelle
β_i	Angle entre le plan du i parallélogramme et le plan horizontal
γ_i	Angle entre le plan vertical π_i et une des barres du parallélogramme i .
θ_i	Angle entre le plan π_i et le plan Oxz

Tableau 1.20 Angles caractéristiques.

- **Volume de travail**

La connaissance du volume de travail, dont la définition est abordée dans la partie généralités, est primordial. En effet, la trajectoire qu'on désire être suivie par la nacelle doit impérativement appartenir à ce volume.

Le volume de travail du robot Delta a la forme indiquée dans la figure suivante :

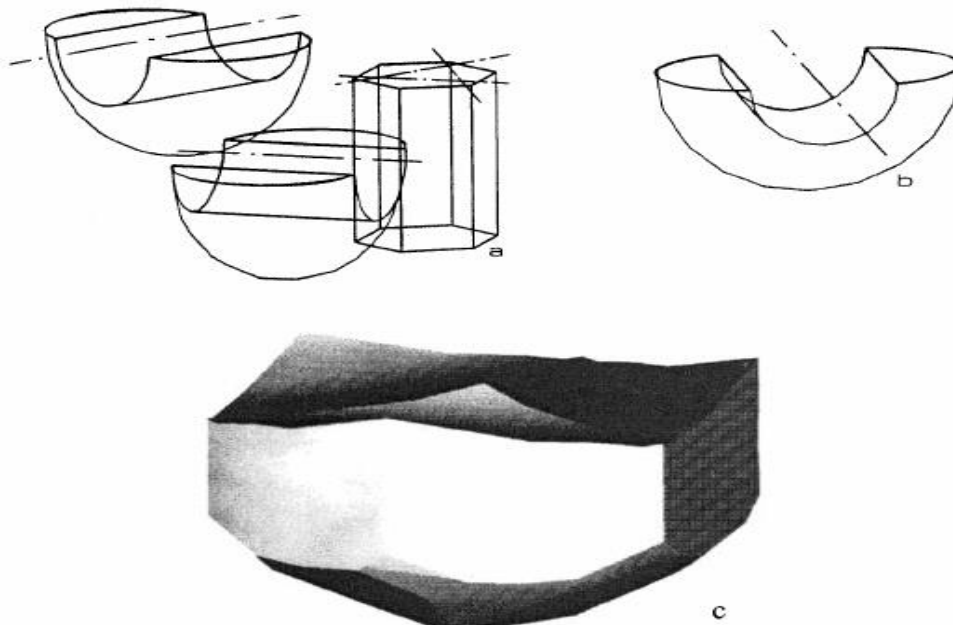


Fig. 1.20 Le volume de travail du robot Delta en (c) est donné par l'intersection du prisme hexagonal en (a) avec les trois corps de révolution en (b) dont les axes sont ramenés au prisme en (a).

En plus des restrictions du volume et afin d'éviter les ambiguïtés lors du calcul des transformations de coordonnées et des singularités, des limites géométriques s'imposent:

Limite 1 : $-\gamma_{max} \leq \gamma \leq \gamma_{max}$ avec $\gamma_{max}=40^\circ$;

Limite 2: $30^\circ \leq \alpha_i + \beta_i \leq 180^\circ$.

La justification de ces limites ainsi que du volume de travail est donnée dans [21]

Le tableau suivant donne des paramètres du volume de travail, leurs définitions ainsi que leurs valeurs pour le robot Delta sur lequel notre étude porte.

Paramètre	Définition	Valeur (ISIR 88)
D	Diamètre du cylindre inscrit dans le volume.	489 mm
S	La surface de la projection du volume sur un plan horizontal.	$20,7 \cdot 10^4 \text{ mm}^2$
H	Hauteur du volume.	H=183 mm
V	Le volume.	$37,9 \cdot 10^6 \text{ mm}^3$

Tableau 1.4 Paramètres du volume de travail

On remarque que le volume de travail est très petit par rapport à la longueur de la chaîne cinématique qui est de 580 mm environ ($L_A + L_B$) cela est dû au fait que la fermeture de la chaîne qui conduit à la restriction des mouvements de la nacelle et par conséquent le volume de travail se restreint.

Singularités du robot Delta :

L'étude des singularités d'un robot est cruciale pour bien le commander. En effet, le passage par une singularité entraîne la perte de contrôle d'un ou plusieurs degrés de liberté.

Les singularités du robots Delta sont au nombre de quatre et sont données dans la figure 1.20.

La définition totale de la position de la nacelle et sa stabilité sont les indices de non singularité et cela se traduit par deux conditions [21] :

Les lignes d'action des forces (les barres parallèles dans ce cas) sont situées dans 3 plans différents non parallèles ;

2 lignes d'action au maximum sont parallèles.

A partir de ces deux conditions, nous tirons les configurations singulières suivantes :

- a) lorsque les 3 paires sont toutes parallèles à une même direction (Fig. 1.20-a);
- b) si 2 paires sont parallèles (Fig. 1.20-b);
- c) si 2 paires sont dans un même plan ou dans 2 plans parallèles (Fig. 1.20-c);
- d) lorsque les 3 parallélogrammes appartiennent à 3 plans parallèles ou confondus (Fig. 1.20-d).

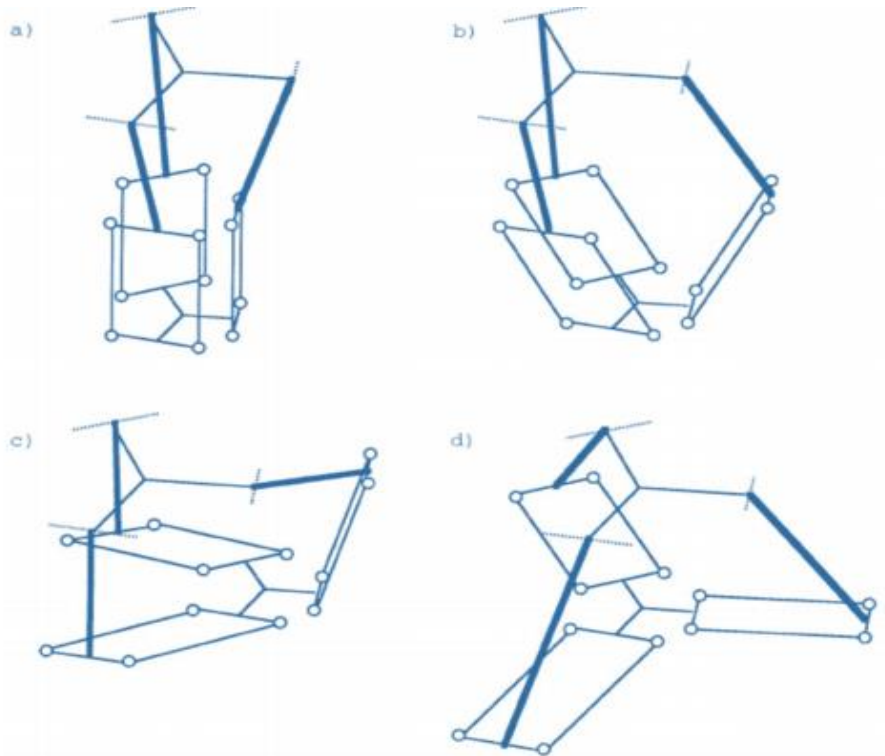


Fig. 1.21 Singularités du robot Delta [21]

1.4 Conclusion

Dans ce chapitre nous avons abordé une introduction sur la robotique en général et sur les robots manipulateurs en particulier. Nous avons traité les principales questions intimement liées à ce domaine comme la modélisation des robots, la génération de trajectoire de référence et la commande des mouvements, des tâches qui constituent le cœur de ce présent projet de fin d'étude.

Dans un second lieu nous avons introduit le robot delta et ses applications, décrit sa structure et ses qualités. Ensuite nous nous sommes intéressés au cas spécial du robot Delta ISIR 88 et nous avons traité sa géométrie, son volume de travail et ses singularités. Nous pouvons dès lors passer à sa modélisation.

Chapitre 2

Modélisation du Robot Delta

2.1 Introduction

L'étude d'un système physique, de n'importe qu'elle nature, nécessite en général la connaissance de son comportement. Cette dernière notion est techniquement connue sous l'appellation "modélisation". La modélisation vise donc à décrire la manière dont se comporte un système. Dans ce chapitre, nous allons traiter la modélisation de notre système d'étude qui est le robot delta ISIR 88 pour ensuite s'en servir pour les besoins de commande. Ce travail comporte la modélisation géométrique, cinématique, modélisation des accélérations et enfin la modélisation dynamique.

2.2 Modélisation géométrique

La modélisation géométrique du robot Delta consiste à trouver les relations qui lient les coordonnées de la nacelle (x,y,z) et les coordonnées généralisées des articulations des bras $(\alpha_1,\alpha_2,\alpha_3)$.



Fig. 2.1 Schéma synoptique des modèles géométriques directs et inverses

Le MGD donne (x,y,z) en fonction de $(\alpha_1,\alpha_2,\alpha_3)$.

Le MGI donne $(\alpha_1,\alpha_2,\alpha_3)$ en fonction de (x,y,z) .

L'établissement du modèle géométrique se fait en faisant recours à des considérations géométriques simplificatrices.

Le repère de base $\{R\}$ est choisi comme montré à la figure 2.2 i.e. au centre du triangle formé par les axes des 3 moteurs, z, dirigé vers le haut, est perpendiculaire à ce triangle, x est perpendiculaire à l'axe du moteur 1. La triple-symétrie du robot nous permet de traiter chaque chaîne séparément. L'index i ($i=1,2,3$) est utilisé pour identifier chaque chaîne. Les chaînes sont séparées l'une de l'autre par un angle de 120° . Pour chaque chaîne, on associe un repère $\{R_i\}$ confondu avec $\{R\}$ mais tourné autour de (oz) par un angle $\varphi_i = 0^\circ, 120^\circ, 240^\circ$ pour $i=1,2,3$ respectivement (figure 2.3).

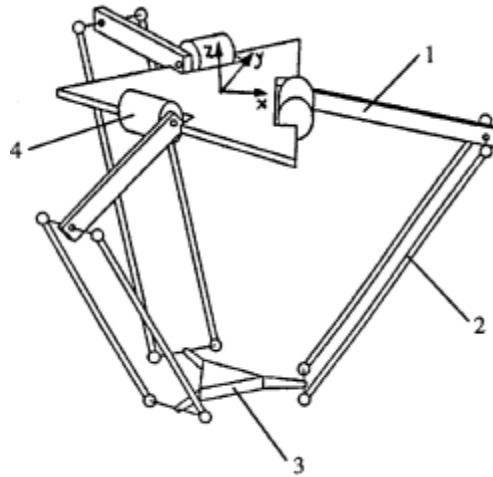


Fig. 2.2 géométrie du Delta

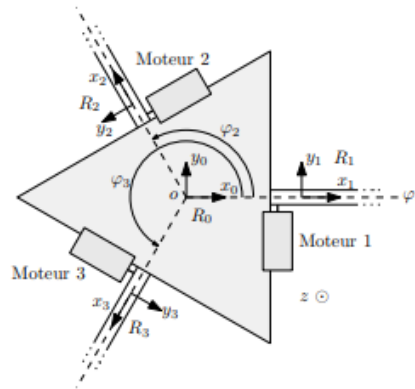


Fig. 2.3 Les rotations entre repères

La matrice de transformation (rotation) entre le repère $\{R_i\}$ et le repère de base $\{R\}$ est donnée par :

$$R_i^R = \begin{bmatrix} \cos(\varphi_i) & -\sin(\varphi_i) & 0 \\ \sin(\varphi_i) & \cos(\varphi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Comme la nacelle effectue seulement des mouvements de translation, elle gardera toujours la même orientation que $\{R\}$. Ceci nous mène à considérer la distance entre $\{R\}$ et le moteur comme étant $R = R_A - R_B$. On considère aussi la nacelle comme un point P avec :

$$P = B_1 = B_2 = B_3$$

Ces considérations (figure 2.4) permettent de simplifier l'établissement du modèle sans trop affecter les résultats.

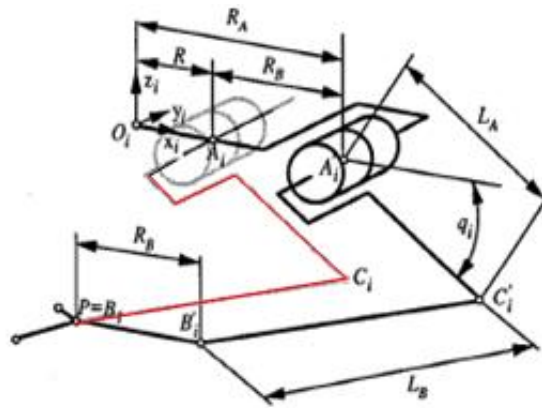


Fig. 2.4 La structure avant et après simplification

Avec ces définitions et hypothèses, les modèles géométriques direct et inverse peuvent être établis comme proposé dans [21] ou [23]. Il convient toutefois de noter que, dans les deux formulations, le modèle géométrique inverse souffre de singularités mathématiques du type $\frac{0}{0}$ qui entraînent des perturbations dans le contrôle du robot à faible vitesse. Pour éliminer ces singularités, un nouveau modèle est proposé dans [24].

2.3 Modélisation cinématique

Détermination de la matrice jacobienne **J** du Delta :

La matrice jacobienne permet d'exprimer la relation entre les vitesses opérationnelles et les vitesses articulaires du Delta comme suit :

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J \cdot \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{bmatrix} = J \cdot \dot{\alpha} \quad (2.2)$$

Pour les robots sériels, il existe des approches systématiques pour déterminer **J** mais malheureusement ce n'est pas le cas concernant les robots parallèles. Pour le robot Delta, la matrice **J** est déterminée pour la première fois par Codourey par un calcul numérique des dérivées partielles du modèle géométrique [22].

Une autre façon de calculer le jacobien des robots parallèles est à considérer un ensemble d'équations de contraintes reliant les variables d'espace opérationnel aux variables d'espace articulaire. Nous décrivons ci-après une version simplifiée de cette dernière formulation.

Les trois équations de contrainte dans le cas du robot DELTA peuvent être choisies comme :

$$\|\overrightarrow{C_i B_i}\|^2 - L_B^2 = 0 \quad i = 1,2,3 \quad (2.3)$$

C'est-à-dire que la longueur des avant-bras doit être constante.

Soit S_i le vecteur $\overline{C_i B_i}$. L'équation précédente peut alors s'écrire :

$$S_i^T S_i - L_B^2 = 0 \quad i = 1,2,3 \quad (2.4)$$

Avec :

$$S_i = \overline{C_i B_i} = \overline{O_i B_i} - (\overline{O_i A_i} + \overline{A_i C_i}) \quad (2.5)$$

Donc :

$$S_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - R_i^R \left(\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_A \cos(\alpha_i) \\ 0 \\ -L_A \sin(\alpha_i) \end{bmatrix} \right) \quad i = 1,2,3 \quad (2.6)$$

La dérivée temporelle de l'équation (2.4) donne:

$$S_i^T \dot{S}_i + \dot{S}_i^T S_i = 0 \quad i = 1,2,3 \quad (2.7)$$

En vertu de la propriété de commutativité du produit scalaire, la dernière équation devient :

$$S_i^T \dot{S}_i = 0 \quad i = 1,2,3 \quad (2.8)$$

La dérivée temporelle du vecteur S_i est donnée par :

$$\dot{S}_i = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + R_i^R \begin{bmatrix} L_A \sin(\alpha_i) \\ 0 \\ L_A \cos(\alpha_i) \end{bmatrix} \dot{\alpha}_i = \dot{X} + b_i \dot{\alpha}_i \quad i = 1,2,3 \quad (2.9)$$

Avec :

$$b_i = R_i^R \begin{bmatrix} L_A \sin(\alpha_i) \\ 0 \\ L_A \cos(\alpha_i) \end{bmatrix} \quad i = 1,2,3 \quad (2.10)$$

En combinant les équations (2.8) et (2.9) On obtient :

$$S_i^T \dot{X} + \begin{bmatrix} S_1^T b_1 & 0 & 0 \\ 0 & S_2^T b_2 & 0 \\ 0 & 0 & S_3^T b_3 \end{bmatrix} \dot{\alpha} = 0 \quad (2.11)$$

On réécrit cette dernière équation comme suit :

$$\dot{X} = - \begin{bmatrix} S_1^T \\ S_2^T \\ S_3^T \end{bmatrix}^{-1} \begin{bmatrix} S_1^T b_1 & 0 & 0 \\ 0 & S_2^T b_2 & 0 \\ 0 & 0 & S_3^T b_3 \end{bmatrix} \dot{\alpha} \quad (2.12)$$

En l'identifiant avec (2.2) on obtient :

$$J = - \begin{bmatrix} S_1^T \\ S_2^T \\ S_3^T \end{bmatrix}^{-1} \begin{bmatrix} S_1^T b_1 & 0 & 0 \\ 0 & S_2^T b_2 & 0 \\ 0 & 0 & S_3^T b_3 \end{bmatrix} \quad (2.13)$$

Il est à noter que l'expression de la matrice jacobienne dépend en plus des variables articulaires de la position de la nacelle, d'où la nécessité d'utiliser le modèle géométrique directe.

2.4 Accélération

En dérivant l'équation (2.12) on obtient :

$$\ddot{X} = - \begin{bmatrix} S_1^T \\ S_2^T \\ S_3^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{S}_1^T \\ \dot{S}_2^T \\ \dot{S}_3^T \end{bmatrix} J + K \right) \dot{\alpha} + J \ddot{\alpha} \quad (2.14)$$

Avec :

$$K = \begin{bmatrix} \dot{S}_1^T b_1 + S_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{S}_2^T b_2 + S_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{S}_3^T b_3 + S_3^T \dot{b}_3 \end{bmatrix} \quad (2.15)$$

Et :

$$\dot{b}_i = R_i^R \begin{bmatrix} L_A \sin(\alpha_i) \\ 0 \\ L_A \cos(\alpha_i) \end{bmatrix} \dot{\alpha}_i \quad i = 1,2,3 \quad (2.16)$$

Le terme multipliant $\dot{\alpha}$ dans l'équation est identifié comme étant la dérivée de la matrice jacobienne (\dot{J}) dont l'utilisation est nécessaire pour la détermination du modèle dynamique

2.5 Modélisation dynamique

Au cours des dernières décennies, Le développement de modèles dynamiques de robots a été un sujet d'intense recherche. La principale difficulté réside dans la nécessité de trouver une solution suffisamment représentative du système réel tout en étant facile à être calculé en temps réel pour la mise en œuvre dans l'algorithme de commande. Pour les structures parallèles, le problème est encore plus complexe que pour les robots sériels, principalement en raison de la difficulté analytique présentée par l'interdépendance des variables articulaires.

La plupart des chercheurs sont unanimes quant au fait qu'une modélisation qui prend en considération toutes les masses et inerties mène à des solutions très compliquées. Par conséquent, pour des objectifs de commande, des hypothèses simplificatrices ont lieu d'être prises d'autant plus que les commandes que nous allons développer dans ce présent PFE (Projet de fin d'étude) sont connues pour être intrinsèquement robustes.

Hypothèses simplificatrices :

Pour le robot DELTA, la complexité du modèle découle principalement du mouvement des avant-bras (barres parallèles). Nous pouvons simplifier ce problème si nous choisissons de négliger leurs inerties de rotation. Cette hypothèse n'est pas très restrictive en raison de l'utilisation de fibres de carbone dans leur construction. Ainsi, la force entre la plateforme mobile (nacelle) et le bras est dans une direction donnée par l'orientation de l'avant-bras. Le

modèle développé pour le robot DELTA est basé sur la méthode de Newton-Euler avec les hypothèses simplificatrices suivantes :

- Les inerties de rotation des avant-bras sont négligées ;
- Les effets de frottement et l'élasticité sont négligés.
- Les masses d'avant-bras sont séparées en deux parties et placées à leurs deux extrémités, $\frac{2}{3}$ à l'extrémité supérieure et $\frac{1}{3}$ à l'extrémité inférieure. Ceci est facilement justifiable par le fait que l'inertie I d'une barre rigide de longueur L et de masse m par rapport à l'une des extrémités vaut : $I = \frac{1}{3}mL^2$ ce qui est équivalent à placer $\frac{1}{3}$ de la masse à l'extrémité mobile.

Paramètres dynamiques

Dans la suite, plusieurs paramètres dynamiques sont utilisés pour établir le modèle dynamique du robot. Ils sont énumérés ici pour plus de commodité.

Au niveau de la nacelle, seule la masse totale m_{nt} est considérée, à savoir :

$$m_{nt} = m_n + m_{payload} + 3(1 - r)m_{ab} \quad (2.17)$$

Qui est la somme de la masse de la nacelle m_n , la masse de la charge $m_{payload}$ et les 3 masses rapportées de chacun des 3 avant-bras où $r = \frac{2}{3}$ est le rapport de masse des avant-bras située à l'extrémité supérieure.

Au niveau des bras, la position du centre de masse du bras est calculée comme suit:

$$r_{Gb} = L_A \frac{\frac{1}{2}m_{br} + m_c + m_{ab}}{m_b} \quad (2.18)$$

Avec :

$$m_b = m_{br} + m_c + rm_{ab} \quad (2.19)$$

Où :

m_{br} : la masse du bras ;

m_c : la masse du coude ;

m_{ab} : la masse de l'avant-bras.

L'inertie à considérer au niveau du bras I_{bi} est la somme de l'inertie du moteur après réduction et rapporté au bras I_m plus l'inertie du bras lui même I_{br} plus l'inertie du système de réduction rapporté au bras I_{red} . Elle est donnée par :

$$I_{bi} = I_m + I_{br} + I_{red} \quad (2.20)$$

Avec :

$$I_{br} = L_A^2 \left(\frac{1}{3}m_{br} + m_c + rm_{ab} \right) \quad (2.21)$$

Le facteur de réduction étant égal à 12, l'inertie après réduction sera multipliée par un facteur de 12^2 .

Le théorème de Huygens est utilisé pour ramener les inerties au niveau du bras.

Modèle dynamique basé sur le principe du travail virtuel :

Avec l'hypothèse simplificatrice mentionnée précédemment, le robot peut être réduit à 4 corps seulement : la nacelle et les 3 bras supérieurs.

Deux types de forces agissent sur la nacelle : la force de gravité G_n et la force d'inertie F_n . Elles sont respectivement données par :

$$G_n = m_{nt}[0 \quad 0 \quad -g]^T \quad (2.22)$$

$$F_n = m_{nt}\ddot{X} \quad (2.23)$$

\ddot{X} et \ddot{q} sont liées par :

$$\ddot{X} = J\ddot{q} + \dot{J}\dot{q} \quad (2.24)$$

D'où

$$F_n = m_{nt}(J\ddot{q} + \dot{J}\dot{q}) \quad (2.25)$$

La contribution de ces deux forces à chacun des moteurs peut alors être calculée en les multipliant par la transposée de la matrice jacobienne :

$$\tau_n = J^T F_n = J^T m_{nt}(J\ddot{q} + \dot{J}\dot{q}) \quad (2.26)$$

$$\tau_{Gn} = J^T G_n = J^T m_{nt}[0 \quad 0 \quad -g]^T \quad (2.27)$$

Selon le principe du travail virtuel, la contribution de toutes les forces non inertielles doit être égale à la contribution de toutes les forces inertielles. Ce principe appliqué au niveau articulaire donne:

$$\tau + \tau_{Gn} + \tau_{Gb} = I_b \cdot \ddot{q} + \tau_n \quad (2.28)$$

τ est le vecteur des couples développés aux bras avec:

$$\tau = 12 \tau_m \quad (2.29)$$

τ_m étant le couple développé par les moteurs.

I_b est la matrice d'inertie des bras dans l'espace articulaire et s'exprime ainsi:

$$I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \quad (2.30)$$

τ_{Gb} est le vecteur des couples produits par les forces gravitationnelles des bras et il est donné par:

$$\tau_{Gb} = m_b r_{Gb} g [\cos(\alpha_1) \quad \cos(\alpha_2) \quad \cos(\alpha_3)]^T \quad (2.31)$$

De (2.26) et (2.28) :

$$\tau = (I_b + J^T m_{nt} J) \ddot{q} + J^T m_{nt} \dot{J} \dot{q} - \tau_{Gn} - \tau_{Gb} \quad (2.32)$$

En identifiant l'équation (2.32) avec celle du modèle général des robots manipulateurs :

$$\tau = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad (2.33)$$

On trouve :

$$M(q) = (I_b + J^T m_{nt} J) \quad (2.34)$$

$$C(q, \dot{q}) = J^T m_{nt} \dot{J} \quad (2.35)$$

$$G(q) = -\tau_{Gn} - \tau_{Gb} \quad (2.36)$$

2.6 Conclusion

Le robot Delta possède une structure parallèle complexe qui se traduit par une modélisation assez difficile. D'où la nécessité de faire recours à des hypothèses simplificatrices. Cette modélisation étant ainsi faite, nous pouvant passer au problème de commande avec l'idée à la tête que la commande doit être robuste.

Chapitre 3

Commandes robustes du Robot Delta

3.1 Introduction

Dans ce chapitre nous allons aborder le problème de commande du robot Delta ISIR 88. Nous allons développer théoriquement les commandes que nous allons utiliser à savoir la commande PD, PD d'ordre fractionnaire, TDC avec PD, TDC avec PD d'ordre fractionnaire et TDC avec modes de glissement. Les résultats de simulation sur MATLAB sont donnés et commentés en parallèle avec le développement théorique tout en comparant les commandes entre elles.

Afin de simuler les lois de commande développées dans la suite, on considère la trajectoire prise-dépose définie dans le plan XZ par les équations des variables opérationnelles (1.9). La trajectoire est divisée en deux phases :

- De $(-0.2, 0, -0.4)m$ à $(0.2, 0, -0.4)m$ dans un temps $t_f = 0.2$ s, avec une hauteur maximale de $0.04m$ suivant l'axe Z , sans masse de charge.
- De $(0.2, 0, -0.4)m$ à $(-0.2, 0, -0.4)m$ dans un temps $t_f = 0.2$ s, avec une hauteur maximale de $0.04m$ suivant l'axe Z , avec une masse de charge pour tester la robustesse des lois de commande synthétisées.

Le temps de l'opération sera $T = 0.4$ s ce qui implique une cadence de prise-dépose de 2.5 Hz. Pour pouvoir comparer les différentes lois de commande, on se base sur le critère RMS (Root Mean Square). Le RMS de l'erreur de poursuite (la formule reste valable pour le RMS de la commande) est donnée par la formule suivante :

$$RMSE = \sqrt{RMSE_{\alpha_1}^2 + RMSE_{\alpha_2}^2 + RMSE_{\alpha_3}^2} \quad (3.1)$$

$$\text{Avec } RMSE_{\alpha_i} = \sqrt{\frac{1}{n} \sum_{j=1}^n e_i(j)^2}$$

Où :

n : nombre d'échantillons ;

$e_i(j)$: l'erreur de poursuite pour l'articulation α_i dans l'échantillon j ;

$RMSE_{\alpha_i}$: est RMS de l'erreur de poursuite pour l'articulation α_i ;

3.2 Commande par PD et PD d'ordre fractionnaire

3.2.1 Commande par PD

Elle est l'une des commandes classiques les plus utilisées en asservissement de position des robots industriels vu sa maîtrise, simplicité d'implémentation et son faible coût de calcul. Sa loi se donne par :

$$u(t) = k_p e(t) + k_d \dot{e}(t) \quad (3.2)$$

Où :

$u(t) \in \mathbb{R}^3$: représente la commande (l'entrée du système) ;

$e(t) \in \mathbb{R}^3$: l'erreur de poursuite (écart entre position désirée et position mesurée) ;

$K_p, K_d \in \mathbb{R}^{3 \times 3}$: sont les matrices des paramètres du régulateur, déterminés de façon à satisfaire un cahier de charges.

La procédure de détermination de ces paramètres dans le cas de la robotique industrielle est abordée dans [1]. Pour le cas du robot Delta ISIR 88, les paramètres par défaut implémentés dans le robot sont $k_p = \text{diag}(2.2; 2.2; 2.2)$ et $k_d = \text{diag}(0.0145; 0.0145; 0.0145)$ dans l'approche décentralisée où l'articulation est commandée séparément et le couplage considéré comme une perturbation.

La structure de commande est illustrée dans la figure 3.1

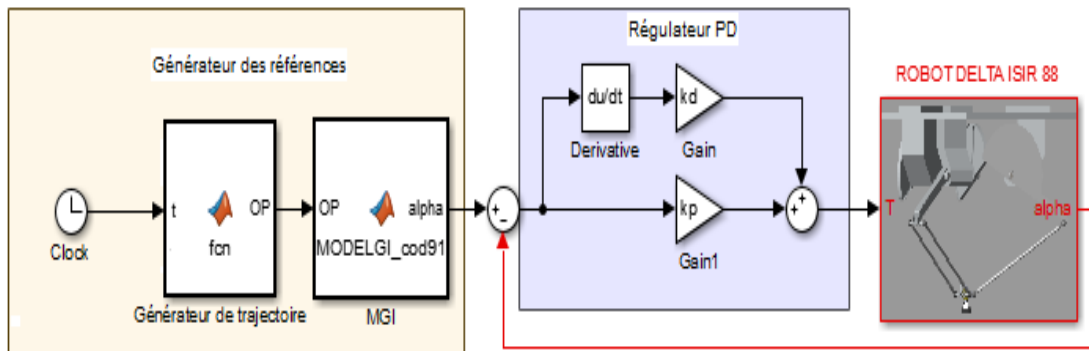


Fig. 3.1 Schéma de commande du robot DELTA avec régulateur PD

Résultats de simulation avec PD

La simulation sous MATLAB/SIMULINK du robot Delta réglé par un régulateur proportionnel-dérivée (l'approche décentralisée) avec les mêmes paramètres que ceux implémentés dans le robot réel a donné les résultats illustrés dans la figure 3.2.

Nous remarquons (Fig.3.2) qu'il y a effectivement suivi de trajectoire dans la phase aller (0s-0.2s) où la nacelle ne porte pas de charge avec une erreur de poursuite maximale de 0.04 rad (espace articulaire). Toutefois, dans la phase retour (0.2s-0.4s) où la nacelle porte une charge de 0.1 kg il n'y a plus suivi de trajectoire. L'erreur de poursuite dans l'espace articulaire a augmenté, elle a pour valeur maximale 0.07 rad, ceci s'est traduit par la remarquable dégradation de la poursuite dans l'espace opérationnel. Donc cette commande est sensible aux variations paramétriques considérables autrement dit sa robustesse est limitée. La valeur RMS de l'erreur de poursuite est de 0.0216 rad. De plus la commande présente un grand pic (15 N.m) à t=0s ce qui est déconseillé du fait que ça risque de détériorer l'état des moteurs. La valeur RMS de la commande est de 1.2916 N.m.

Toutes ces remarques nous mènent à la nécessité de trouver d'autres régulateurs. Toutefois, le régulateur PD est très simple à implémenter et a pu, plus ou moins, commander un système non linéaire aussi complexe.

3.2.2 Commande PD d'ordre fractionnaire

3.2.2.1 Généralités sur le calcul fractionnaire

La généralisation du calcul différentiel et intégral sur l'ordre non entier (dit fractionnaire) a été proposée pour la première fois par Leibniz et l'Hopital. Des recherches ont été ensuite faites sur ce domaine par de nombreux chercheurs à l'instar de Liouville 1832, Holmgren 1864 et Riemann 1953[2].

On trouve dans la littérature différentes définitions de l'opérateur intégro-différentiel telles que celles de Cauchy, Grunwald Leitnikov, Caputo et bien d'autres [3].

Cet opérateur est :

$${}_a D_t^r = \begin{cases} \frac{d^r}{dt^r} & : r > 0 \\ 1 & : r = 0 \\ \int_a^t (d\tau)^{-r} & : r < 0 \end{cases} \quad (3.3)$$

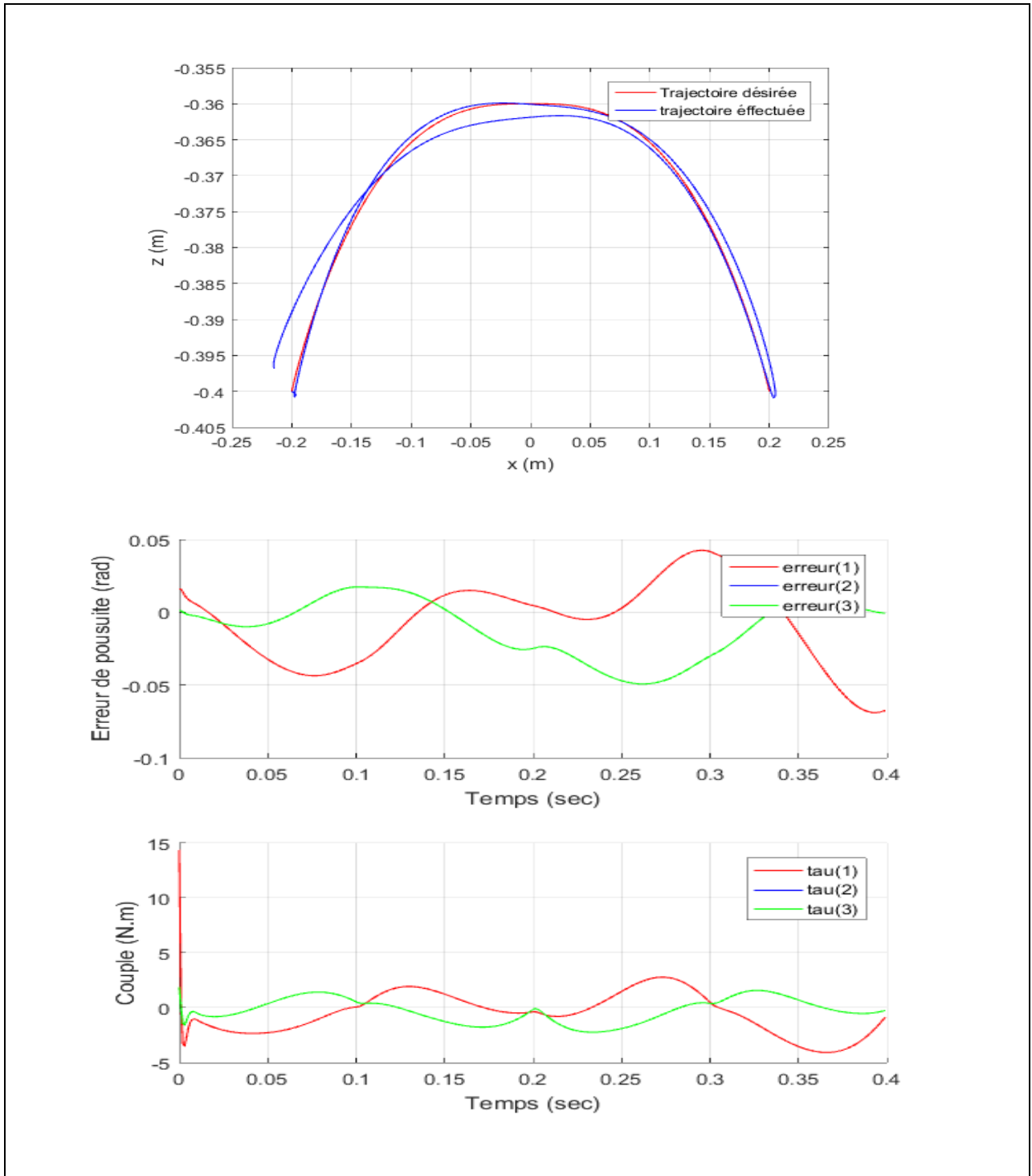


Fig. 3.2 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur PD.

La transformée de Laplace des dérivées d'ordre non entier s'avère nécessaire pour une étude optimale des systèmes modélisés par des équations différentielles non entières et pour une synthèse plus aisée des lois de commande. En effet, cet outil mathématique a été élargi pour

rester applicable pour le cas fractionnaire et ne présente pas beaucoup de différences par rapport au cas classique. La transformée de Laplace de la dérivée fractionnaire dans le cas de conditions initiales nulles garde la même formule :

$$L\left\{\frac{d^m f(t)}{dt^m}\right\} = s^m L\{f(t)\} \quad (3.4)$$

Avec $m \in \mathbb{R}$ l'ordre de différentiation.

Nous pouvons dès lors définir des fonctions de transfert fractionnaires et tirer parti de ses avantages. Elles sont exprimées sous la forme :

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0}} \quad (3.5)$$

Il y a plusieurs méthodes de passage entre le cas fractionnaire et le cas entier à travers des outils appelés filtres. Le filtre récursif d'Oustaloup [4] en est l'une des possibilités. Il donne une très bonne approximation des opérateurs fractionnaires dans une bande de fréquence spécifique. C'est pourquoi il est très utilisé pour l'implémentation pratique des systèmes et régulateurs fractionnaires (ce qui est notre cas). Son expression mathématique est la suivante :

$$s^\alpha \approx K \prod_{k=1}^N \frac{s + \omega'_k}{s + \omega_k} \quad (3.6)$$

Avec :

$$\omega_k = \omega_b \cdot \omega_u^{(2k-1+\alpha)/N}$$

$$\omega'_k = \omega_b \cdot \omega_u^{(2k-1-\alpha)/N}$$

$$K = \omega_h^\alpha, \omega_u = \sqrt{\omega_h / \omega_b}$$

N : étant l'ordre de l'approximation dans la bande de fréquence $[\omega_b, \omega_h]$;

Et $\alpha \in [-1, 1]$ l'ordre de différentiation.

Pour l'implémentation digitale, sur micro-processeur en l'occurrence, d'une fonction de transfert fractionnaire, on l'exprime tout d'abord sous forme de filtres d'Oustaloup, puis on discrétise la formule obtenue. La discrétisation se fait avec une période d'échantillonnage T_s ($T_s = 0.001s$ dans notre cas) donc ω_h est choisie de façon à satisfaire la condition $\omega_h = 2/T_s$. Le résultat sera une fonction de transfert en z exprimée sous la forme pôles-zéros [4]. Ensuite,

on passe à l'écriture de cette dernière sous forme d'équation aux différences dont l'implémentation est facilement faisable.

Il existe plusieurs toolbox Matlab permettant l'étude des systèmes d'ordre non entier. Parmi les plus avérées la toolbox FOMCON. Elle propose plusieurs fonctionnalités telles que l'identification, la modélisation, la simulation et même l'implémentation des régulateurs fractionnaires sous différentes formes. Elle jouit remarquablement d'une simplicité d'utilisation grâce aux interfaces graphiques dont elle est dotée. On en trouve "impid", "iopid_tune" et "fomcon" .

Des algorithmes de commande à base d'ordre fractionnaire ont été développés et ont démontré leurs grandes utilité et efficacité. Le régulateur CRONE, introduit par Oustaloup[3], a connu un grand succès et jouit d'une robustesse incontestable. Une généralisation des régulateurs PID a été aussi réalisée.

3.2.2.2 Régulateur PD d'ordre fractionnaire

Ce régulateur, qui est un cas particulier des régulateurs PID d'ordre fractionnaire proposés par Podlubny, est une généralisation du régulateur PD, sa loi de commande est définie ainsi

$$u(t) = K_p e(t) + K_d D^r e(t) \tag{3.7}$$

Avec $0 \leq r \leq 2$: l'ordre de dérivation.

Sa fonction de transfert est :

$$C(s) = k_p + k_d s^r \tag{3.8}$$

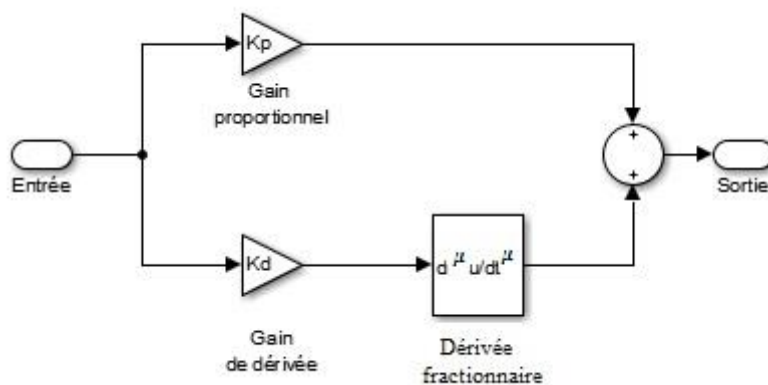


Fig. 3.3 Structure d'un régulateur PD d'ordre fractionnaire

On remarque que ce régulateur possède trois paramètres, un de plus par rapport au PD classique, donc un degré de liberté de plus. Ce dernier est utilisé généralement pour satisfaire aux conditions de robustesse en spécifiant la marge de phase à imposer.

L'utilisation de commandes robustes est très recommandée dans le cas de présence d'incertitudes de modélisation ou de variations paramétriques, ce qui est notre cas. La modélisation dynamique du robot DELTA a été faite en faisant recours à des hypothèses simplificatrices d'où l'introduction d'incertitudes. De surcroît, la masse de la charge portée par la nacelle change, ce qui entraîne fatalement une variation paramétrique.

Il existe dans la littérature plusieurs méthodes de détermination des paramètres du régulateur PD d'ordre fractionnaire. Parmi elles, la détermination des paramètres K_p et K_d comme pour le cas du PD classique puis la détermination de r ou bien par ajustement ou bien par un algorithme d'optimisation (l'optimisation par essaim de particules par exemple [27]) pour satisfaire une contrainte donnée qui est généralement la marge de phase, la marge de gain ou les sensibilités.

Dans le but de comparer les performances du régulateur PD classique et celles du PD d'ordre fractionnaire, on garde les mêmes valeurs des gains K_p, K_d que la simulation précédente. La valeur de r est déterminée par ajustement ($r = 0.91$).

Résultats de simulation avec PD d'ordre fractionnaire :

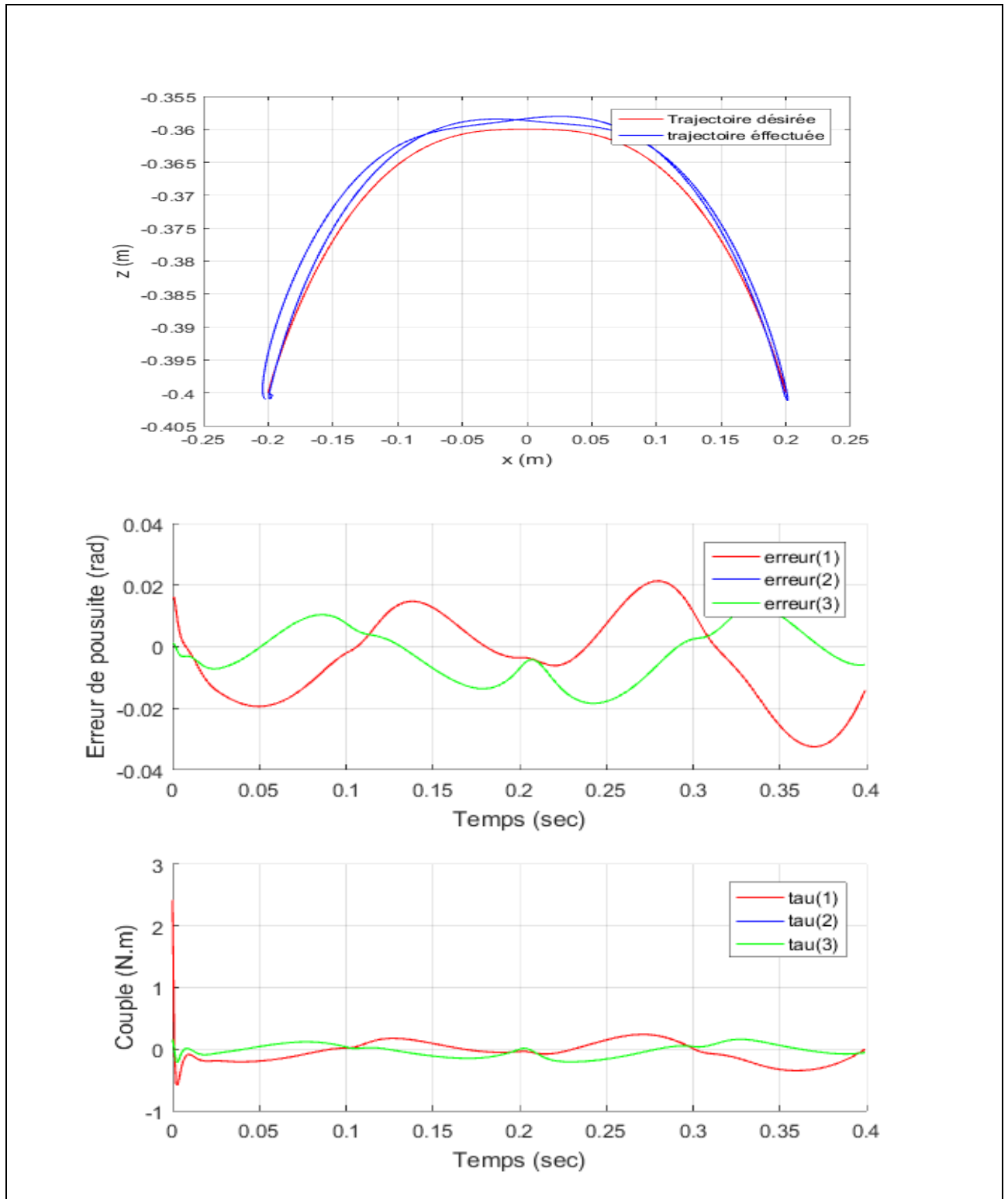


Fig. 3.4 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur PD fractionnaire.

Nous remarquons que les erreurs de poursuites articulaires ont nettement diminué par rapport au PD classique avec une valeur maximale de 0.02 rad dans la phase aller et de 0.035 rad dans la phase retour (avec une charge de 0.1 kg). La valeur RMS de l'erreur de poursuite est de 0.016 rad et elle est inférieure à celle du PD d'ordre entier. Les résultats qu'on vient de citer montrent que le PD d'ordre fractionnaire synthétisé est assez acceptable que ce soit en performance ou en robustesse face aux variations paramétriques. De plus, le couple (la commande) a considérablement diminué (2.4 N.m, pour le pic au début et un max de 0.5N.m, sur le reste avec une valeur RMS de 0.126 N.m), ce qui veut dire que cette dernière commande est plus économique énergétiquement.

Les valeurs des gains K_p et K_d étant les même pour le PD classique et le PD d'ordre fractionnaire, nous pouvons affirmer que c'est l'ordre de dérivation r qui a amélioré les performances et assuré la robustesse.

3.3 Commande Time delay control (TDC)

Dans la section précédente, nous avons étudié le régulateur PD d'ordre entier ainsi que son homologue d'ordre fractionnaire appliqués au robot DELTA réputé pour avoir une structure complexe. Dans la recherche d'un régulateur meilleur nous allons explorer d'autres stratégies de commande. La qualité la plus convoitée pour notre cas de figure étant la robustesse, notre choix est automatiquement porté sur l'une des commandes intrinsèquement robustes. Beaucoup d'entre elles ont été déjà appliquées sur le robot DELTA à l'instar de la commande par mode de glissement [6], la commande H^∞ [7], la commande par apprentissage itératif [8][9].

Généralement ces commandes nécessitent des algorithmes d'approximation très complexes ou de larges données d'expérimentation, ce qui s'ajoutent à la complexité de leur implémentation.

Dans le but de contourner toutes ses difficultés, une nouvelle stratégie de commande robuste est proposée pour le DELTA. C'est de la commande **TDC** (Time Delay Control) qu'il est question.

Le régulateur TDC présenté dans [10],[11], possède une structure remarquablement simple et sa caractéristique clé réside dans l'annulation directe des incertitudes sans avoir à augmenter un gain particulier.

3.3.1 Structure de la TDC

Ce nouveau régulateur est composé principalement de deux parties indépendantes [10], la partie TDE ('time delay estimation' en anglais, veut dire en français 'estimation à retard temporel') qui est l'élément chargé d'annuler les non linéarités continues (incluant la gravité , les couples centrifuges et de Coriolis ,la friction, les perturbations et les couples d'interaction) et les incertitudes du modèle dynamique, l'autre partie étant un élément injecté qui a pour tâche d'imposer une dynamique désirée pour l'erreur de poursuite (servo-régulateur).Les deux parties sont synthétisées séparément. La nécessité de faire appel à un troisième élément est abordée dans [12]. Cet élément est chargé de compenser les non linéarités discontinues (friction de Coulomb, etc) qui augmentent les erreurs résiduelles de l'estimation TDE. La structure générale de la commande TDC est illustrée dans la figure 3.5.

- **Time Delay estimation (TDE)**

Cette partie du régulateur sert à se passer de l'identification hors ligne et de la connaissance préalable de la dynamique du robot. Sa structure est simple et elle peut estimer à la fois facilement et efficacement les fonctions non linéaires englobant la dynamique inconnue ainsi que les perturbations. Ceci en faisant recours uniquement à l'information retardée temporellement des dérivées de la variable d'état et de l'entrée de commande. Autrement dit en utilisant leurs valeurs dans les échantillons précédents sous l'hypothèse que la dynamique non connue ne change pas considérablement pour un pas d'échantillonnage suffisamment petit.

En fait, Il existe deux versions pour l'implémentation de la TDE [10], celle avec des gains fixes et celle avec des gains adaptatifs. Dans notre cas nous allons utiliser la version avec gains fixes.

- **Servo-régulateur**

Comme précédemment évoqué, la technique TDE permet non seulement de se passer avec succès et de manière simple de la connaissance complète du modèle du système mais aussi de compenser toutes les parties inconnues. La synthèse du servo-régulateur se fera donc en se basant sur le modèle nominal résultant, ce qui est assez simple du moment que ça devient une simple imposition de la dynamique de l'erreur de poursuite pour satisfaire un cahier de charge donné. C'est cette dernière opération que nous désignons par servo-réglage.

Dans ce qui suit, deux servo-régulateurs seront utilisés. Le premier étant le PD établi dans [10] et développé ensuite dans [11]. Le second, qui est notre propre contribution fera appel au PD d'ordre fractionnaire comme servo-régulateur.

- **Élément de robustesse**

Afin d'éliminer les non linéarités dures et augmenter la robustesse du régulateur tout en assurant un temps de convergence fini, un troisième élément est généralement ajouté [13][12]. Il est question du TSM ('Terminal Sliding Mode' en anglais signifiant 'mode de glissement terminal') avec différentes surfaces de glissement. Parmi elles la surface d'ordre non entier utilisée dans [14].

Dans cette section nous allons utiliser deux surfaces de glissement à savoir la surface simple de Slotine[15], et la surface non linéaire de Jin [16].

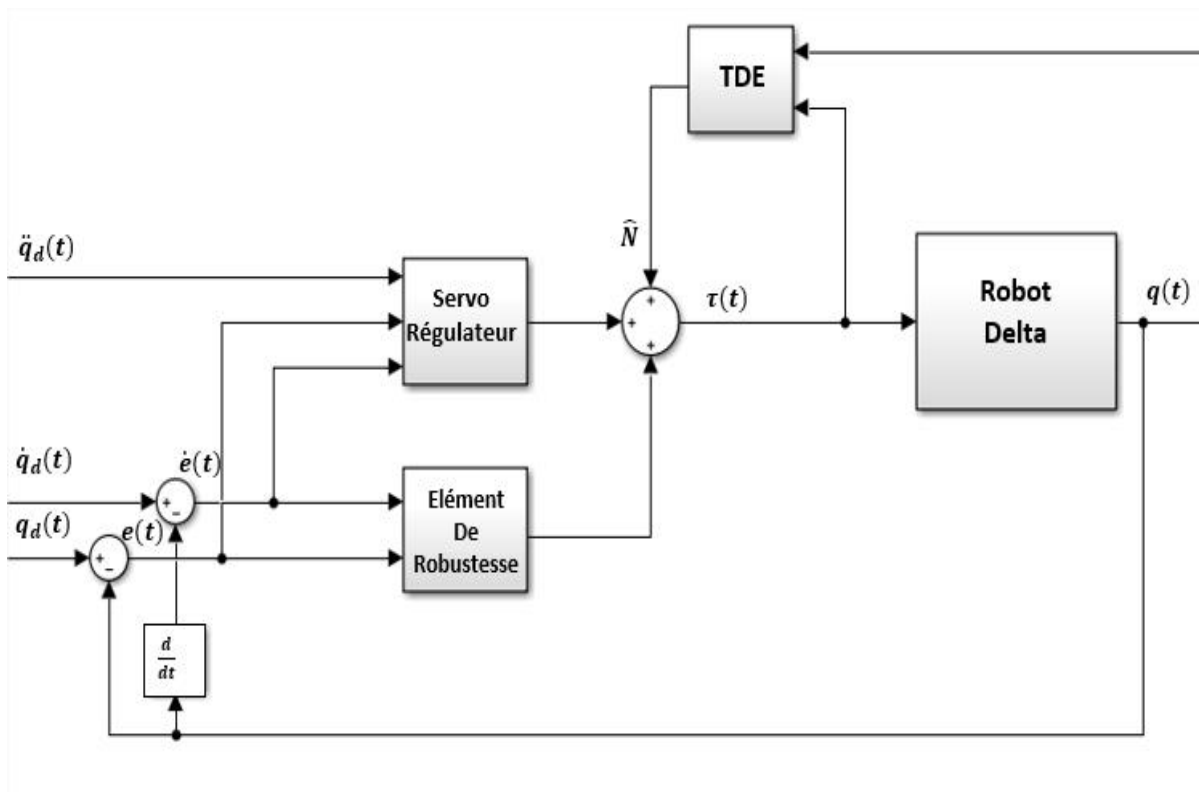


Fig. 3.5 Structure de commande TDC

3.3.2 Technique d'estimation TDE

Nous avons mentionné dans le chapitre précédent que la dynamique du robot Delta est donnée par l'équation :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.9)$$

Où :

$q, \dot{q}, \ddot{q} \in \mathbb{R}^3$: les vecteurs position, vitesse et accélération respectivement.

$\tau \in \mathbb{R}^3$: le vecteur des couples appliqués ;

$M(q) \in \mathbb{R}^{3 \times 3}$: la matrice d'inertie du robot ;

$C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$: la matrice des forces centrifuges et de Coriolis ;

$G(q) \in \mathbb{R}^3$: vecteur des forces gravitationnelles ;

La structure du Delta étant complexe, le modèle est obtenu sous des hypothèses simplificatrices.

La synthèse d'un régulateur en se basant sur un tel modèle risque de ne pas fonctionner sur le système réel d'où la nécessité de prendre un modèle plus significatif de la forme :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \tau_d = \tau \quad (3.10)$$

Où $F(q, \dot{q}) \in \mathbb{R}^3$ les forces de friction, $\tau_d \in \mathbb{R}^3$ couple perturbateur qui peut être considéré incluant les parties négligées lors de la modélisation.

L'étape suivante consiste à reformuler l'équation (3.10) en introduisant une matrice diagonale constante $\bar{M} \in \mathbb{R}^{3 \times 3}$:

$$\bar{M}\ddot{q} - \bar{M}\ddot{q} + M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \tau_d = \tau \quad (3.11)$$

On reformule de nouveau :

$$\bar{M}\ddot{q} + [M(q) - \bar{M}]\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \tau_d = \tau \quad (3.12)$$

On pose :

$$N(q, \dot{q}, \ddot{q}) = [M(q) - \bar{M}]\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \tau_d \quad (3.13)$$

En substituant (3.13) dans (3.12) on obtient

$$\bar{M}\ddot{q} + N(q, \dot{q}, \ddot{q}) = \tau \quad (3.14)$$

De (3.14), nous avons

$$N(q, \dot{q}, \ddot{q})_t = \tau_t - \bar{M}\ddot{q}_t \quad (3.15)$$

Où :

$N(q, \dot{q}, \ddot{q})_t \in \mathbb{R}^3$: désigne la valeur de $N(q, \dot{q}, \ddot{q})$ à l'instant t .

τ_t : désigne la valeur de τ à l'instant t .

\ddot{q}_t : désigne la valeur de \ddot{q} à l'instant t .

Il est clairement facile d'évaluer $N(q, \dot{q}, \ddot{q})_t$ par $\tau_t - \bar{M}\ddot{q}_t$. Toutefois, cette évaluation ne peut pas être implémentée physiquement car non causale (deux évènements ayant lieu simultanément). Pour remédier à cette restriction on considère un temps L suffisamment petit pour lequel $N(q, \dot{q}, \ddot{q})_t$ peut être approximé par $N(q, \dot{q}, \ddot{q})_{t-L}$ (la valeur de $N(q, \dot{q}, \ddot{q})$ à l'instant $(t - L)$).

D'où :

$$\hat{N}(q, \dot{q}, \ddot{q})_t = N(q, \dot{q}, \ddot{q})_{t-L} \quad (3.16)$$

L : est le retard temporel d'estimation. Dans l'implémentation digitale, Le plus petit L atteignable est le temps d'échantillonnage.

$\hat{N}(q, \dot{q}, \ddot{q})_t$: la valeur estimée à l'instant t de la fonction $N(q, \dot{q}, \ddot{q})$

De (3.15) et (3.16) , on tire :

$$\hat{N}(q, \dot{q}, \ddot{q})_t = \tau_{t-L} - \bar{M}\ddot{q}_{t-L} \quad (3.17)$$

En l'occurrence, l'accélération \ddot{q}_{t-L} (valeur de \ddot{q} à l'instant $(t - L)$) se calcule numériquement comme suit :

$$\ddot{q}_{t-L} = \frac{q_t - 2q_{t-L} + q_{t-2L}}{L^2} \quad (3.18)$$

L'erreur d'estimation de $N(q, \dot{q}, \ddot{q})$ est montrée dans la figure 3.6 avec $N(q, \dot{q}, \ddot{q})$ calculé de l'équation (3.13) en utilisant le modèle dynamique et $\hat{N}(q, \dot{q}, \ddot{q})$ calculé par estimation TDE. L est pris égal au pas d'échantillonnage comme recommandé précédemment ($L = 0.001$ s). Il est clair que l'estimation TDE est efficace dans l'évaluation de $N(q, \dot{q}, \ddot{q})$ contenant toutes les non linéarités (Fig.3.6).

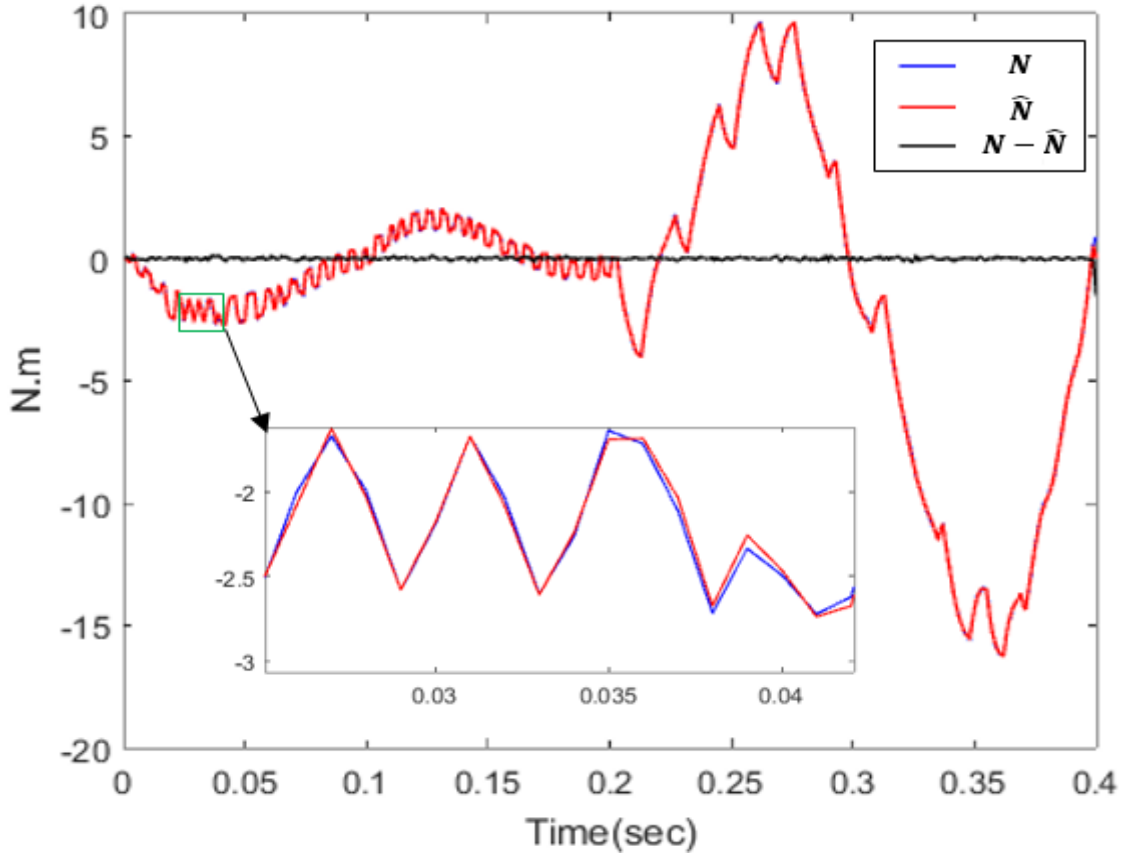


Fig. 3.6 L'erreur d'estimation de $N(q, \dot{q}, \ddot{q})$ par la technique TDE.

- **Méthodes de détermination de \bar{M} :**

Il existe plusieurs techniques servant à déterminer les éléments de la matrice \bar{M} :

- La plus utilisée est la méthode par ajustement, on commence par des petites valeurs et on augmente jusqu'à obtention des performances désirées.
- Une autre méthode est proposée dans [12], où \bar{M} dépend des éléments diagonaux de $M(q)$.
- Il y a aussi une version adaptative expliquée dans [10].

- **Démonstration que l'erreur d'estimation est bornée :**

La loi de commande est donnée par :

$$\tau = \bar{M}u + \hat{N}(q, \dot{q}, \ddot{q}) \quad (3.19)$$

La synthèse détaillée de u sera abordée dans la section suivante.

En considérant les équations (3.14) et (3.19) la dynamique en boucle fermée est donnée par :

$$u - \ddot{q} = \varepsilon \quad (3.20)$$

Avec $\varepsilon \in \mathbb{R}^3$ définie comme l'erreur de l'estimateur TDE en boucle fermée et donnée par :

$$\varepsilon := \bar{M}^{-1}(N - \hat{N}) \quad (3.21)$$

De (3.15) :

$$M\varepsilon = M(u - \ddot{q}) \quad (3.22)$$

Injectant (3.22) dans (3.10), on obtient :

$$M\varepsilon = Mu + C\dot{q} + G + \tau_d - \tau \quad (3.23)$$

On remplace τ par son expression donnée dans (3.19) :

$$M\varepsilon = Mu + C\dot{q} + G + \tau_d - \bar{M}u - N_{t-L} \quad (3.24)$$

De (3.12) :

$$N_{t-L} = (M_{t-L} - \bar{M})\ddot{q}_{t-L} + (C\dot{q})_{t-L} + G_{t-L} + F_{t-L} + (\tau_d)_{t-L} \quad (3.25)$$

On substitue (3.25) dans (3.24), on obtient :

$$M\varepsilon_t = (M - \bar{M})u_t - (M_{t-L} - \bar{M})\ddot{q}_{t-L} + \delta \quad (3.26)$$

Avec :

$$\delta = (C\dot{q})_t - (C\dot{q})_{t-L} + G_t - G_{t-L} + F_t - F_{t-L} + \tau_d - (\tau_d)_{t-L} \quad (3.27)$$

Pour un L suffisamment petit, il est clair que δ est bornée.

Substituant $\ddot{q}_{t-L} = \varepsilon_{t-L} - u_{t-L}$ de (3.20) dans (3.26) :

$$M_t\varepsilon_t = (M_t - \bar{M})\varepsilon_{t-L} + (M_t - \bar{M})(u_t - u_{t-L}) + (M_t - M_{t-L})\ddot{q}_{t-L} + \delta \quad (3.28)$$

Donc ε est donnée par :

$$\varepsilon_t = E\varepsilon_{t-L} + E\eta_1 + \eta_2 \quad (3.29)$$

Avec :

$$E = I - M_t^{-1}\bar{M} \quad (3.30)$$

$$\eta_1 = u_t - u_{t-L} \quad (3.31)$$

$$\eta_2 = M_t^{-1}[(M_t - M_{t-L})\ddot{q}_{t-L} + \delta] \quad (3.32)$$

Pour un L suffisamment petit, η_1 et η_2 sont bornées. Il est à noter que l'hypothèse $\|E\| < 1$ peut facilement être satisfaite par un choix convenable de \bar{M} [17].

En discret, (3.29) s'écrit comme suit :

$$\varepsilon(k) = E(k)\varepsilon(k-1) + E(k)\eta_1(k) + \eta_2(k) \quad (3.33)$$

La solution de l'équation aux différences précédente se donne ainsi :

$$\begin{aligned} \varepsilon(k) = & \prod_{m=1}^k E(m)\varepsilon(0) + \sum_{m=1}^k \prod_{p=m}^k E(p)\eta_1(m) \\ & + \sum_{m=2}^k \prod_{p=m}^k E(p)\eta_1(m-1) + \eta_2(k) \end{aligned} \quad (3.34)$$

Avec $\varepsilon(0)$ la valeur initiale de $\varepsilon(k)$.

Soit λ le maximum de $\|E\|$, $\lambda < 1$.

Soient ρ_1 et ρ_2 des constantes positives telles que $\|\eta_1\| \leq \rho_1$ et $\|\eta_2\| \leq \rho_2$.

Soit le lemme suivant :

$x, y \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ matrice symétrique, et μ le maximum de $\|A\|$.

Si $x = Ay$, alors $\|x\| \leq \mu\|y\|$.

De (3.34) et en vertu du lemme précédent :

$$\begin{aligned} \|\varepsilon(k)\| &\leq \lambda^k \|\varepsilon(0)\| + \sum_{m=1}^k \lambda^{k-m+1} \|\eta_1(m)\| \\ &\quad + \sum_{m=2}^k \lambda^{k-m+1} \|\eta_2(m-1)\| + \|\eta_2(k)\| \\ &\leq \lambda^k \|\varepsilon(0)\| + \rho_1 \sum_{m=1}^k \lambda^m + \rho_2 \sum_{m=0}^{k-1} \lambda^m \leq \frac{\rho_1 \lambda + \rho_2}{1 - \lambda} := \rho \end{aligned} \quad (3.35)$$

Remarque :

ρ est une constante positive. Donc, ε l'erreur TDE est bornée.

3.3.3 Synthèse de la loi de commande

Le couple d'entrée τ est donné dans (3.19), et nous avons vu comment déterminer une estimation de $N(q, \dot{q}, \ddot{q})$ en utilisant la technique TDE. Dans cette section nous allons nous intéresser à la synthèse de u regroupant le second et le troisième élément de TDC, u sera prise sous quatre formes :

- TDC simple en prenant u comme servo-contrôleur uniquement. Une fois comme PD d'ordre entier et une autre comme PD d'ordre fractionnaire.
- Dans les deux dernières nous allons ajouter l'élément de robustesse en utilisant des commandes à mode de glissement avec deux surfaces différentes.

L'objectif de commande est de déterminer l'entrée $\tau \in \mathbb{R}^3$ de façon à forcer $q \in \mathbb{R}^3$ à suivre asymptotiquement la trajectoire de référence $q_d \in \mathbb{R}^3$. Autrement dit, forcer l'erreur de poursuite $e \in \mathbb{R}^3$ à devenir asymptotiquement stable.

L'erreur de poursuite est donnée par :

$$e = q_d - q \quad (3.36)$$

3.3.3.1 TDC approche classique

- **TDC a base de l'estimateur TDE et servo-régulateur PD**

Ce régulateur, montré dans la figure 3.7, a été développé dans [10] et [11], en utilisant simplement l'estimation à retard temporel (TDE) pour déterminer le terme $N(q, \dot{q}, \ddot{q})$, puis imposer la dynamique de l'erreur de poursuite en boucle fermée sous forme proportionnelle dérivée d'ordre entier. Cette méthode est similaire à celle du couple calculé [18] avec la différence que cette dernière est basée sur le modèle pour annuler les non linéarités.

On prend :

$$u = \ddot{q}_d + K_d \dot{e} + K_p e \quad (3.37)$$

Substituant (3.37) et (3.17) dans (3.19), le couple de commande s'écrit :

$$\tau = \tau_{t-L} - \bar{M} \ddot{q}_{t-L} + \bar{M} (\ddot{q}_d + K_d \dot{e} + K_p e) \quad (3.38)$$

De (3.37) et (3.14) on obtient :

$$\bar{M} \ddot{q} + N(q, \dot{q}, \ddot{q}) = \hat{N}(q, \dot{q}, \ddot{q}) + \bar{M} (\ddot{q}_d + K_d \dot{e} + K_p e) \quad (3.39)$$

La dynamique du système en boucle fermée est donnée par l'équation :

$$\ddot{e} + K_d \dot{e} + K_p e = \varepsilon \quad (3.40)$$

Avec ε est l'erreur TDE donnée dans (3.21). Dans l'approche proposée par Hsia, l'erreur ε est proche de 0 si L est suffisamment petit et $N(q, \dot{q}, \ddot{q})$ est continue. Dans cette condition, la dynamique en boucle fermée est donnée dans [10] par :

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (3.41)$$

Par le choix de $K_d = \text{diag}(k_{d1}, k_{d2}, k_{d3})$; $K_p = \text{diag}(k_{p1}, k_{p2}, k_{p3})$ avec $k_{di}, k_{pi} > 0$, $i = 1,2,3$, la dynamique de l'erreur de poursuite en boucle fermée est complètement découplée. Finalement nous avons à déterminer les paramètres de réglage k_{di}, k_{pi} . Nous allons les prendre sous la forme proposée dans [11]:

$$k_{pi} = \omega_{ni}^2; \quad \text{et } k_{di} = 2\xi_i \omega_n$$

Dans la simulation les valeurs des paramètres sont $\omega_{ni} = 80 \text{ rad/s}$ et $\xi_i = 1$, $i = 1,2,3$

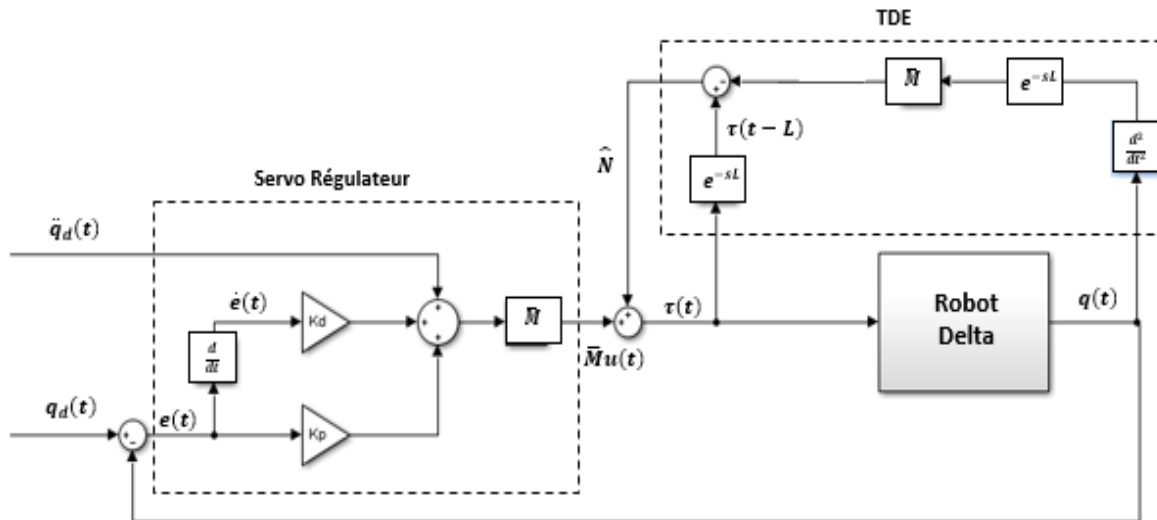


Fig. 3.7 Structure de commande TDC simple basée sur servo-contrôleur PD

Résultats de simulation avec commande TDC (PD)

Comme montré à la figure 3.8, les erreurs de poursuite articulaires ont encore diminué pour avoir une valeur maximale de 0.016 rad et une valeur RMS de 0.0066 rad ce qui s'est traduit par une assez bonne poursuite de la trajectoire désirée. De plus la charge de 0.4 kg porté dans la phase retour n'a pas dégradé les performances (superposition des trajectoires aller et retour) et ceci grâce à l'estimation TDE qui a pris en charge cette variation paramétrique d'où la robustesse de la commande TDC. Tout de même, la poursuite n'est pas assez satisfaisante dans les extrémités de la trajectoire ce qui peut être justifié par la dégradation de la TDE dans les variations brusques.

Quant à la commande, le premier constat à faire est l'absence de pics. Il en est de même pour les broutements. En outre la valeur RMS de la commande est de 0.1686 qui est inférieure à celle de PD classique et légèrement supérieure à celle de PD fractionnaire.

Les résultats étant bons mais toujours loin des performances convoitées, la quête du régulateur les assurant continue.

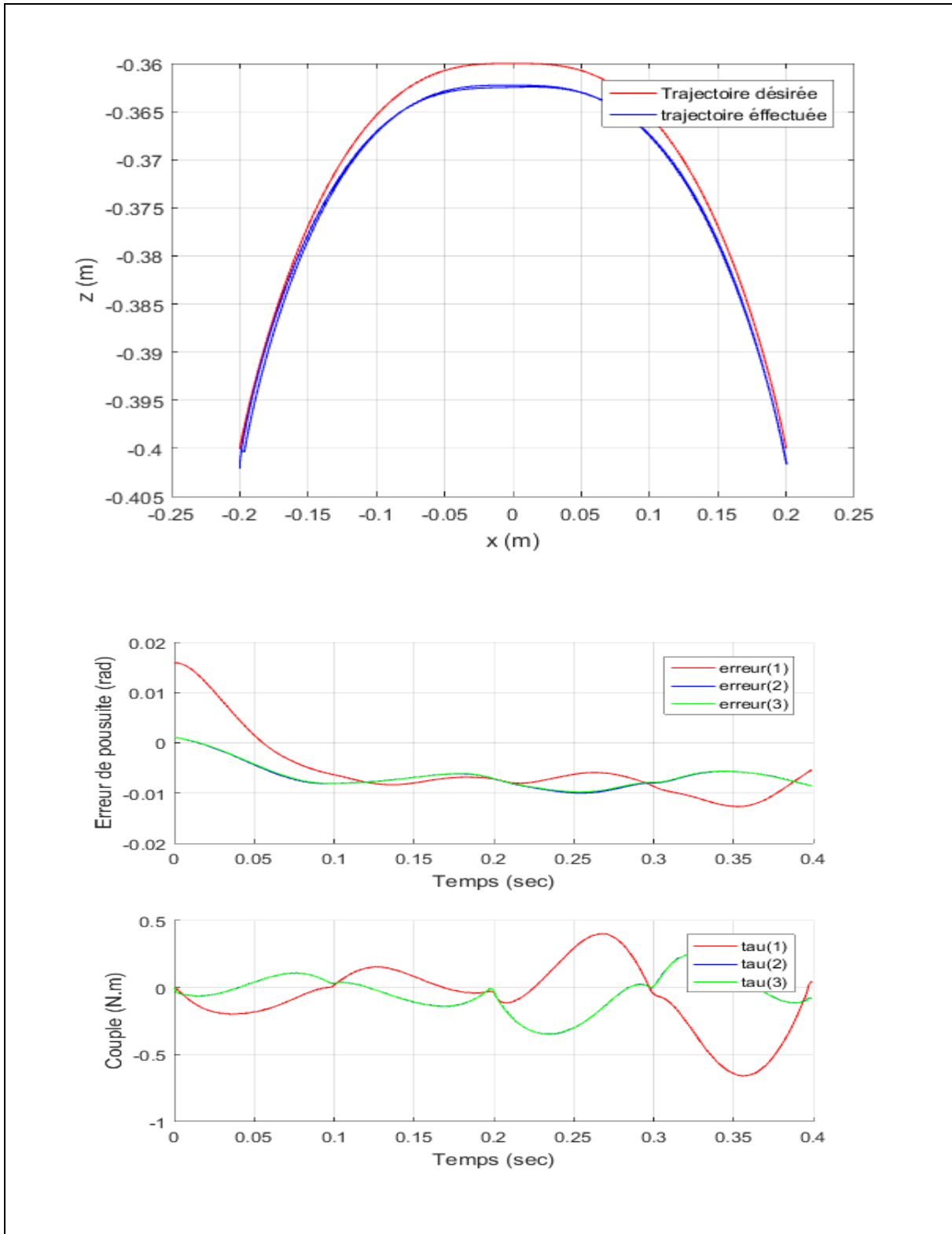


Fig. 3.8 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-PD.

• **TDC à base d'estimateur TDE et servo-régulateur PD d'ordre fractionnaire**

Cette fois nous allons appliquer un autre servo-régulateur, au lieu d'imposer l'erreur de poursuite en boucle fermée sous forme proportionnelle dérivée, nous allons prendre la version fractionnaire. La structure de commande sera la même (figure 3.7) avec la différence que l'ordre de différenciation est non entier.

La loi du servo-régulateur est donnée par :

$$u = \ddot{q}_d + K_d \frac{d^r e}{dt^r} + K_p e \quad (3.42)$$

avec $r \in \mathbb{R}^+$

De la même façon que le PD d'ordre entier et sous les mêmes conditions (L suffisamment petit, N est une fonction continue) l'erreur de poursuite en boucle fermée est :

$$\ddot{e} + K_d \frac{d^r e}{dt^r} + K_p e = 0 \quad (3.43)$$

Où $K_d = \text{diag}(k_{d1}, k_{d2}, k_{d3})$; $K_p = \text{diag}(k_{p1}, k_{p2}, k_{p3})$;

A noter que la dynamique de l'erreur de poursuite en boucle fermée est complètement découplée. Elle est donnée par trois équations différentielles indépendantes.

La convergence de l'erreur peut être démontrée en suivant les étapes suivantes :

- Premièrement on détermine les constantes du PD d'ordre fractionnaire.
- En utilisant le filtre d'Oustaloup nous pouvons approximer la dérivée d'ordre non entier sous forme de fonction de transfert d'ordre entier.

$$s^r \cong G(s) = \frac{Z(s)}{P(s)} \quad (3.44)$$

avec Z et P des polynômes d'ordre $2n + 1$ où n est l'ordre du filtre d'Oustaloup.

En appliquant la transformation de Laplace à l'équation (3.43) on obtient :

$$s^2 + K_d s^r + K_p = 0 \quad (3.45)$$

De (3.44) et (3.45) :

$$(s^2 + K_p)P(s) + K_d Z(s) = 0 \quad (3.46)$$

- Finalement, on utilise le critère algébrique de Routh pour vérifier la stabilité de (3.46).

Dans la simulation on garde des mêmes valeurs des gains k_{di}, k_{pi} comme la simulation précédente et l'ordre de dérivation $r = 0.96$.

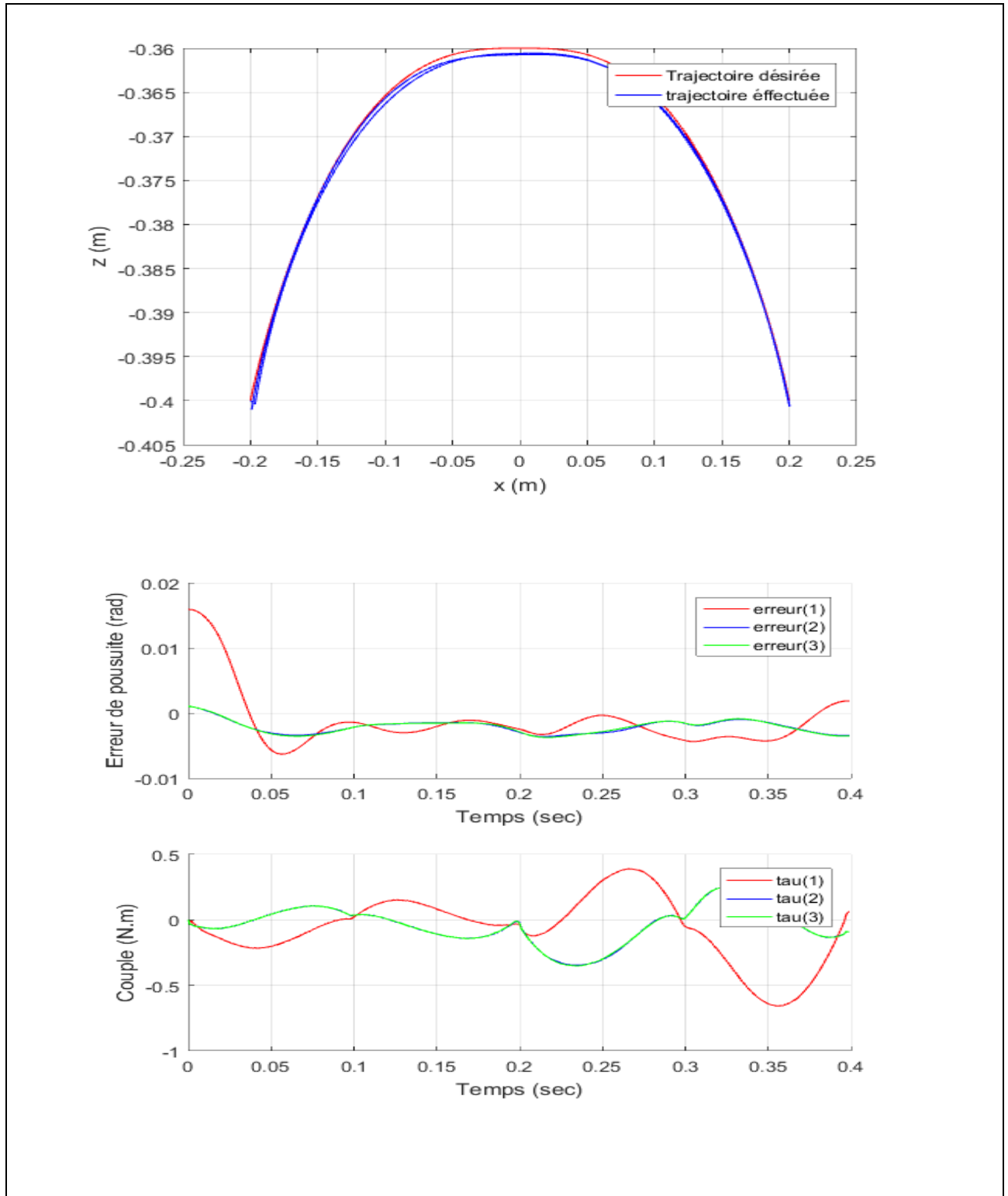


Fig. 3.9 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-PD d'ordre fractionnaire.

Résultats de simulation avec commande TDC(PDF)

La combinaison de TDC avec PD d'ordre fractionnaire peut être qualifiée de doublement robuste du fait que les deux approches sont connues pour leur robustesse. La poursuite s'est améliorée nettement et clairement. En effet la valeur RMS de l'erreur de poursuite a considérablement diminué pour avoir comme valeur 0.0028 rad. Il en est de même pour la valeur maximale qui est de 0.016 rad. Concernant la commande, elle est très acceptable avec valeur maximale de 0.7 N.m et une valeur RMS de 0.1674 avec absence de pics et de broutement.

Une amélioration de plus est souhaitable pour avoir de bons résultats en pratique aussi.

3.3.4.2 TDC avec un élément de robustesse

L'hypothèse que $N(q, \dot{q}, \ddot{q})$ est une fonction continue dans l'approche de Hsia peut mener à l'augmentation de l'erreur de poursuite si elle n'est pas satisfaite [13]. En effet, en présence de discontinuités dures, ce qui est le cas dans la majorité des systèmes réels, une dégradation des performances TDE est engendrée, autrement dit l'erreur TDE ne peut plus être négligée et elle sera traduite par une dégradation de la poursuite.

Afin de remédier à ce problème, le recours à la commande par modes de glissement (SMC) combinée à la TDE avait été proposé [19]. Cet ajout est l'élément de robustesse. La structure de commande est montrée à la figure 3.5.

La commande par modes de glissement est l'une des découvertes les plus importantes dans la théorie de commande moderne [20][16]. De plus, il a été montré qu'elle peut résoudre efficacement le problème de l'erreur résiduelle de TDE.

Dans la suite, nous allons utiliser deux surfaces de glissement différentes, la première étant la surface linéaire de Slotine et la deuxième sera une surface non linéaire (surface integral) de Jin.

- **TDC avec surface de glissement linéaire**

L'approche des modes glissants 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.

La surface de glissement linéaire de Slotine [15] est donnée par :

$$S = \dot{e} + \lambda e \quad (3.47)$$

Telle que $S \in \mathbb{R}^3$ et $\lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3); \lambda_i > 0; i = 1, 2, 3$

Si on arrive à la surface de glissement $S = 0$ l'erreur va converger vers l'origine.

Pour déterminer la loi de commande, on se base sur la fonction de Lyapunov suivante :

$$V = \frac{1}{2} S^T S \quad (3.48)$$

V est définie positive, la dérivée de V par rapport aux temps se donne ainsi :

$$\begin{aligned} \dot{V} &= S^T \dot{S} \\ &= S^T (\ddot{e} + \lambda \dot{e}) \\ &= S^T (\ddot{q}_d - \ddot{q} + \lambda \dot{e}) \end{aligned} \quad (3.49)$$

On remplace \ddot{q} à partir de l'équation dynamique (3.14) dans (3.49) on trouve :

$$\dot{V} = S^T (\ddot{q}_d - \bar{M}^{-1}(\tau - N) + \lambda \dot{e}) \quad (3.50)$$

On continue le développement en remplaçant τ par son expression dans (3.19), on obtient :

$$\begin{aligned} \dot{V} &= S^T (\ddot{q}_d - \bar{M}^{-1}(\bar{M}u + \hat{N} - N) + \lambda \dot{e}) \\ &= S^T (\ddot{q}_d + \bar{M}^{-1}(N - \hat{N}) + \lambda \dot{e} - u) \end{aligned} \quad (3.51)$$

Rappelons que l'erreur de TDE est donnée dans (3.21) par $\varepsilon = \bar{M}^{-1}(N - \hat{N})$ et posons :

$$\begin{aligned} u &= \ddot{q}_d + \lambda \dot{e} + K \text{sign}(S) \\ K &= \text{diag}(k_1, k_2, k_3); \end{aligned} \quad (3.52)$$

D'après (3.21) et (3.52) l'équation (3.51) devient :

$$\dot{V} = S^T (\varepsilon - K \text{sign}(S)) \quad (3.53)$$

On note par S_i les éléments de S

D'après (3.35) l'erreur de TDE avait une borne supérieure positive si on respecte la règle ($\|I - M^{-1}\bar{M}\| < 1$) et L suffisamment petite. $\|\varepsilon\| \leq \rho$.

Si on choisit $K_{min} = \min(k_1, k_2, k_3) > \rho$ l'inégalité suivante est toujours vraie :

$$\dot{V} \leq \sum_{i=1}^3 |S_i| (\rho - K_{min}) \leq -\gamma \sum_{i=1}^3 |S_i| \quad (3.54)$$

Avec $\gamma > 0$.

Par conséquent \dot{V} est définie négative, d'après la deuxième loi de Lyapunov S converge asymptotiquement vers l'origine ce qui implique la convergence de l'erreur e vers l'origine.

De (3.19) et (3.52) la loi de commande est donnée par :

$$\tau = \tau_{t-L} - \bar{M} \ddot{q}_{t-L} + \bar{M} (\ddot{q}_d + \lambda \dot{e} + K \text{sign}(S)) \quad (3.55)$$

Résultats de simulation avec commande TDC (SM Slotine)

Les paramètres de la simulation :

$$k_i = 80 ; \quad i = 1,2,3.$$

$$\lambda_i = 200; \quad i = 1,2,3.$$

Ce qui est remarquable dans ces résultats (Fig.3.10) est les performances étonnantes obtenues. Elles sont de loin les meilleures jusqu'à présent avec une erreur de poursuite maximale de seulement 0.016 rad et une valeur RMS de 0.0016 rad. Toutefois, la commande présente des oscillations remarquables (broutements) dues à la fonction signe qui est à commutation à très haute fréquence (discontinue). Cette commande ne sera donc pas implémentée pratiquement pour éviter de porter atteinte aux moteurs. Nous allons tout de même remédier à ce problème en utilisant la fonction sigmoïde qui est continue et ne présente pas de variations brusques. Les résultats obtenus dans ce cas sont montrés dans la figure 3.11.

Nous remarquons qu'effectivement la commande ne présente plus de broutement tout en gardant presque les mêmes performances.

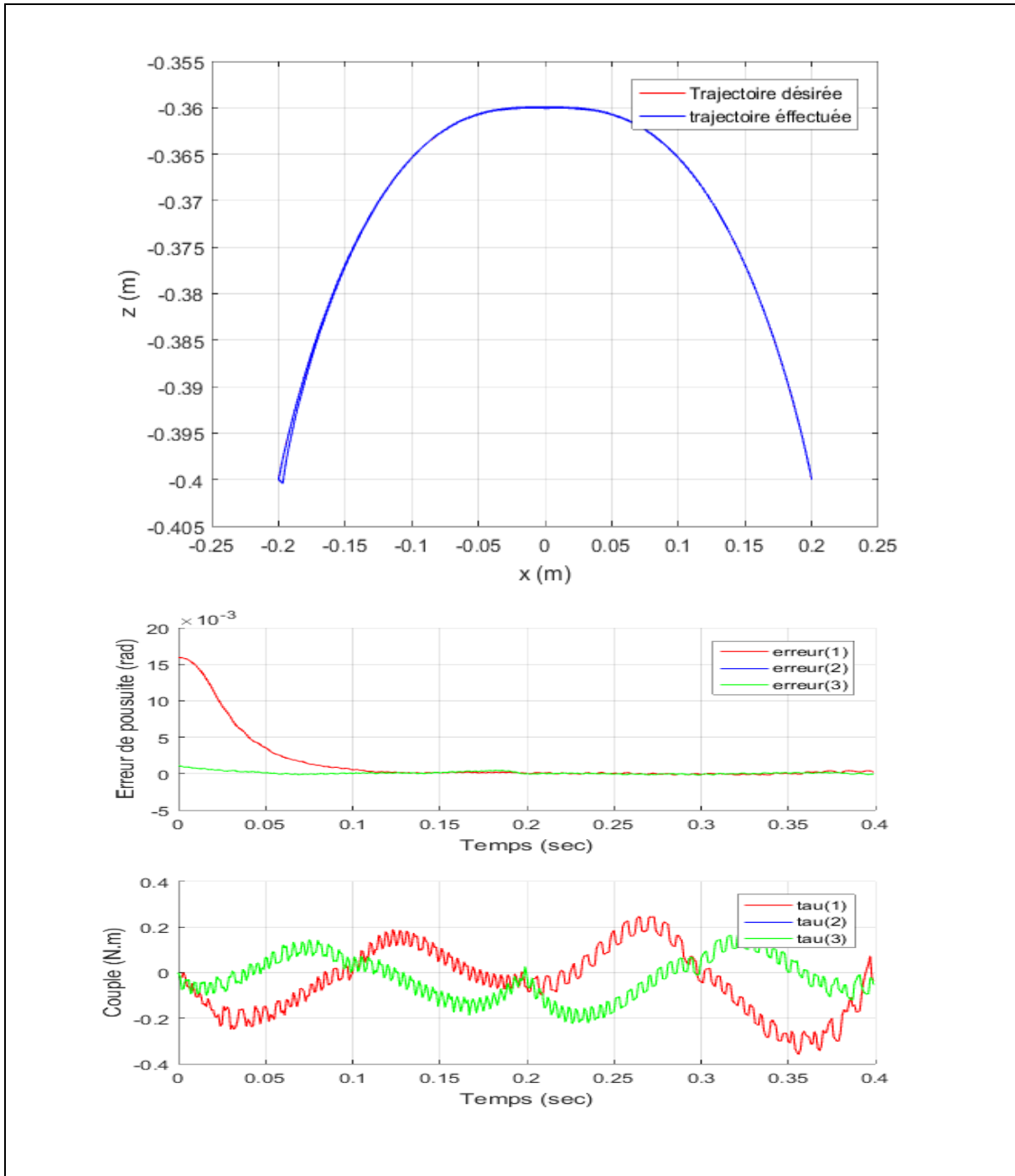


Fig. 3.10 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-SM avec surface de Slotine et fonction signe.

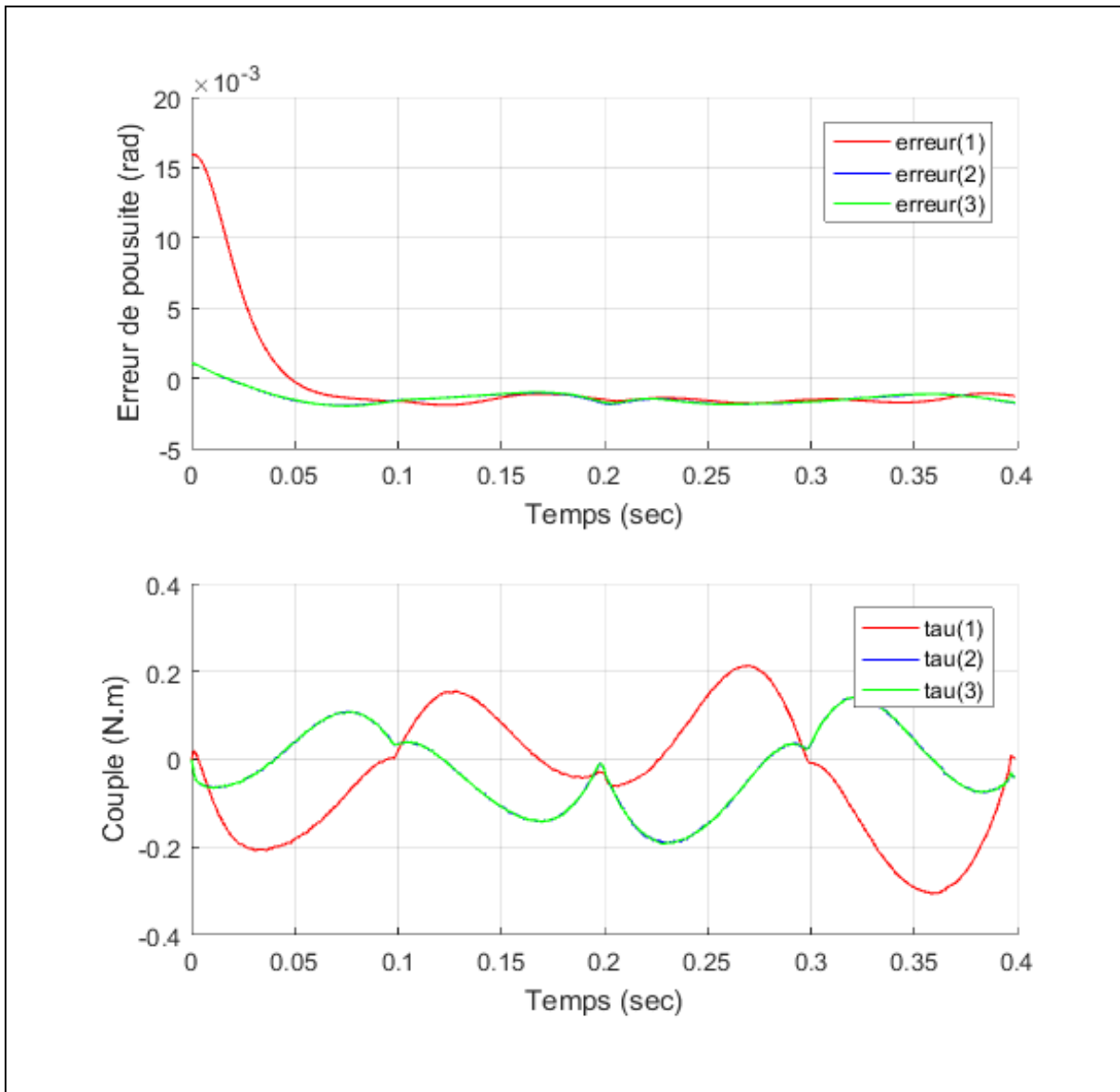


Fig. 3.11 Erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-SM avec surface de Slotine et fonction sigmoïde.

- **TDC avec une surface non linéaire**

Le vecteur d'état représentant la dynamique de l'erreur de poursuite est asymptotiquement stable lorsqu'il appartient à la surface de glissement. Mais nous ne pouvons pas garantir qu'au début ce vecteur soit dans la surface dû à la phase d'attraction ('reaching phase' en anglais). Cette dernière phase conduit à un temps de convergence qui peut être considérable.

Afin de surmonter cette barrière et arriver à un temps de convergence plus rapide la commande par mode de glissement terminal (TSMC) est proposée dans [15]. L'idée clé de cette technique est d'éliminer la phase d'attraction en choisissant une surface de façon que le vecteur d'état

représentant la dynamique de l'erreur y appartienne dès le début. Dans ce qui suit nous allons adopter la surface non linéaire proposée par Jin [16] et satisfaisant la condition précédente.

Cette surface est donnée par :

$$S := \int_0^t (\ddot{e} + K_d \dot{e} + K_p e) dt \quad (3.56)$$

Où $K_d = \text{diag}(k_{d1}, k_{d2}, k_{d3})$; $K_p = \text{diag}(k_{p1}, k_{p2}, k_{p3})$;

Il est clair que pour $t = 0$ s, $S = 0$ quelle que soit l'erreur de poursuite, ce qui veut dire l'élimination de la phase d'attraction précédemment évoquée.

Par analogie avec la section précédente nous allons déterminer la loi de commande en se basant sur la même fonction de Lyapunov donnée par la formule (3.48) .

En prenant en considération (3.44), la dérivée de la fonction de Lyapunov devient :

$$\begin{aligned} \dot{V} &= S^T (\ddot{e} + K_d \dot{e} + K_p e) = S^T (\ddot{q}_d - \ddot{q} + K_d \dot{e} + K_p e) \\ &= S^T (\ddot{q}_d - \ddot{q} + K_d \dot{e} + K_p e) \end{aligned} \quad (4.57)$$

De (3.7) et (3.12), (3.49) devient :

$$\begin{aligned} \dot{V} &= S^T (\ddot{q}_d + \bar{M}^{-1}(N - \hat{N}) + K_d \dot{e} + K_p e - u) \\ &= S^T (\ddot{q}_d + \varepsilon + K_d \dot{e} + K_p e - u) \end{aligned} \quad (4.58)$$

Prenons u comme proposée par Jin [16] :

$$u = \ddot{q}_d + K_d \dot{e} + K_p e + \beta \text{sig}(S)^\gamma \quad (4.59)$$

Avec :

$$\beta = \text{diag}(\beta_1, \beta_2, \beta_3); \beta_i > 0$$

Et $:\text{sig}(S)^\gamma = [|S_1|^{\gamma_1} \text{sign}(S_1), |S_2|^{\gamma_2} \text{sign}(S_2), |S_3|^{\gamma_3} \text{sign}(S_3)]^T$.

$$0 < \gamma_i < 1 \quad (i = 1, 2, 3)$$

Au lieu de $\text{sign}(S)$, l'utilisation de $\text{sig}(S)^\gamma$ diminue considérablement le broutement ('chattering' en anglais).

De (3.58) et (3.59) nous obtenons :

$$\begin{aligned} \dot{V} &= S^T (\varepsilon - \beta \text{sig}(S)^\gamma) = - \sum_{i=1}^3 \beta_i |S_i|^{1+\gamma_i} + \sum_{i=1}^3 S_i \varepsilon_i \\ &\leq -\beta_{\min} \sum_{i=1}^3 |S_i|^{1+\gamma_i} + \sum_{i=1}^3 |S_i| |\varepsilon_i| \end{aligned} \quad (4.60)$$

Avec $\beta_{min} = \min(\beta_1, \beta_2, \beta_3) > 0$

Donc :

$$\dot{V} \leq -\beta_{min} \sum_{i=1}^3 (|S_i|^{\gamma_i} - \beta_{min}^{-1} |\varepsilon_i|) \quad (4.61)$$

\dot{V} : est définie négative si la condition suivante est vérifiée :

$$|S_i| < (\beta_{min}^{-1} \rho)^{\frac{1}{\gamma_i}} \quad (4.62)$$

Donc par un propre choix de β_{min} , \dot{V} sera définie négative ce qui implique d'après la deuxième loi de Lyapunov la convergence de S vers l'origine ce qui donne la convergence de l'erreur de poursuite e vers l'origine.

Les paramètres du régulateur dans la simulation sont :

$$k_{pi} = 1600$$

$$k_{di} = 80$$

$$\gamma_i = 0.6$$

$$\beta_i = 80$$

Avec $i=1,2,3$.

Résultats de simulation avec commande TDC (SM Jin)

Cette dernière commande est remarquablement la meilleure sur tous les plans. En effet, la trajectoire effectuée est superposée à la trajectoire désirée, que ce soit sans ou avec charge sur la nacelle. De plus, les erreurs de poursuite articulaires sont minimales et ont pour RMS 0.0016 rad, une valeur qui est la plus petite atteinte. Quant à la commande, elle est aussi bonne que les performances obtenues avec une valeur RMS de 0.0989 N.m. En outre, elle est très adaptée à l'implémentation pratique du fait qu'elle prend en considération les non linéarités de tous types sans présenter de broutements.

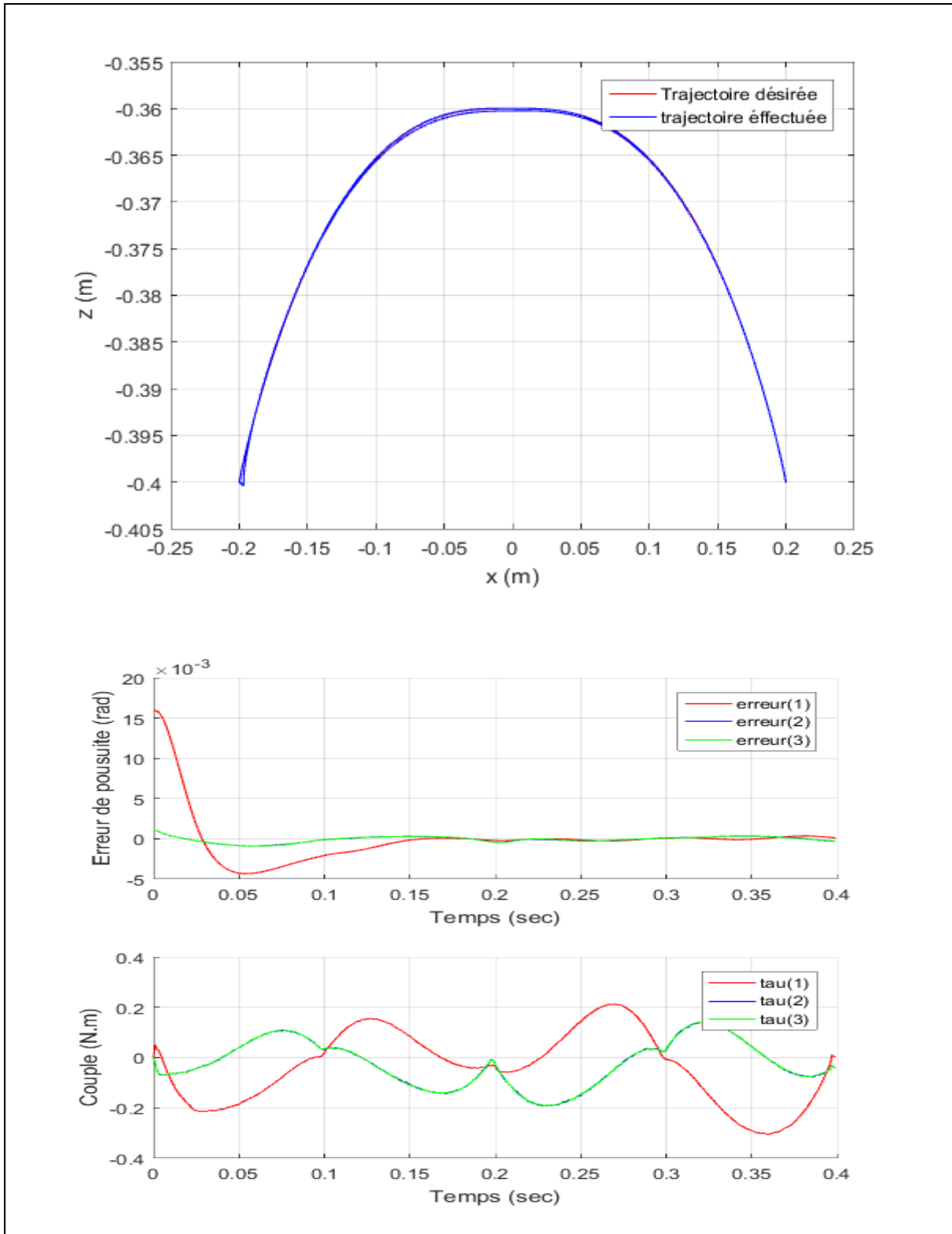


Fig. 3.12 Trajectoire effectuée, erreurs de poursuite articulaires et commandes en couple dans le cas d'un régulateur TDC-TSMC avec surface de Jin.

3.4 Tests de Robustesse

Pour tester la robustesse des lois de commande que nous avons déjà montré dans ce chapitre, par rapport aux variations paramétriques. On varie la masse de charge de 0 kg à 2 kg. Les résultats sont montrés dans les figures 3.13 pour RMS de l'erreur de poursuite et 3.14 pour RMS de couple.

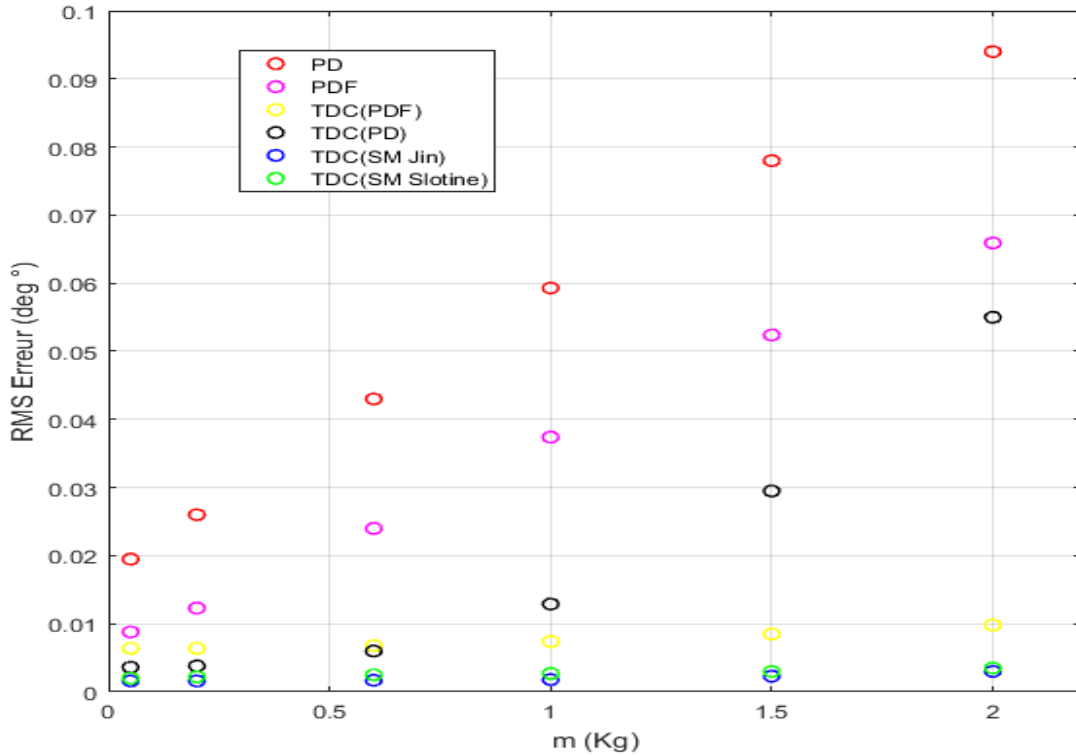


Fig. 3.13 La variations des RMS de l'erreur en fonction de masse de charge.

On remarque que les régulateurs TDC (SM Jin) et TDC (SM Slotine) sont clairement les plus robustes du fait que la valeur RMS de l'erreur de poursuite reste constante en dépit de la variation de la masse. L'origine de cette robustesse est expliquée dans la partie théorique. En seconde place vient la commande TDC (PDF) avec une variation minime de la valeur RMS de l'erreur de poursuite. Ensuite, la robustesse diminue de la commande TDC (PD) ayant un profile parabolique à la commande PDF puis PD ayant des profiles linéaires avec des variations nettement remarquables.

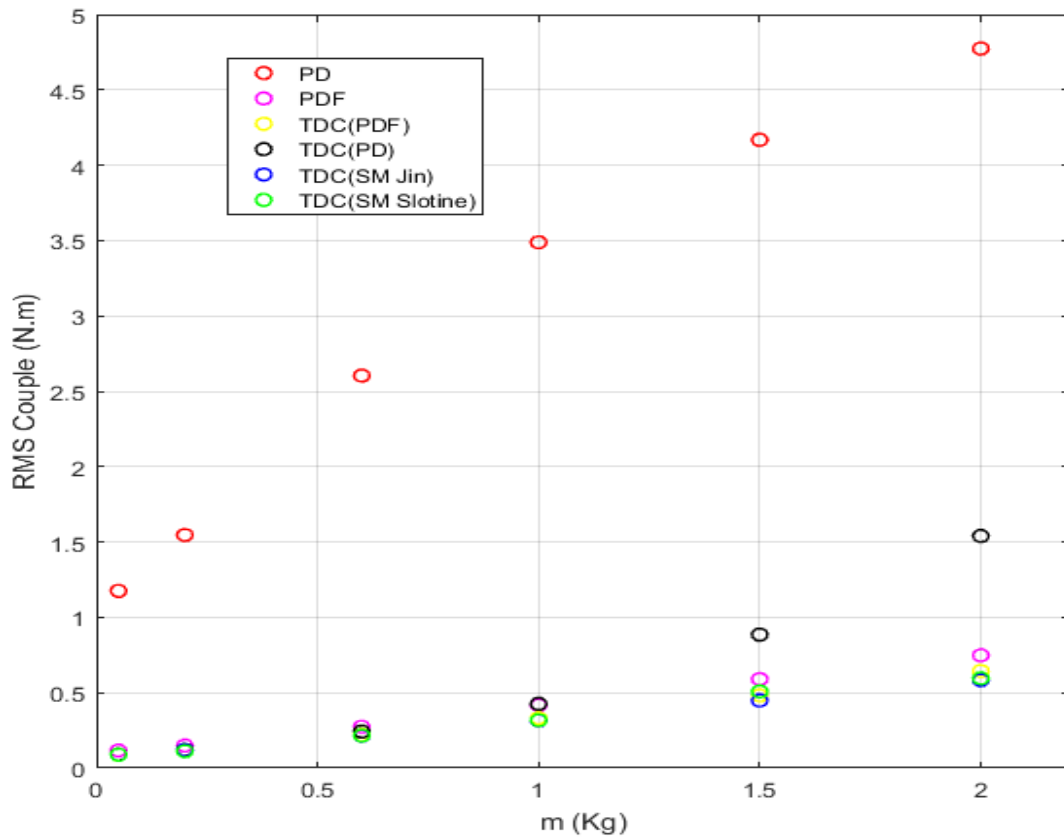


Fig. 3.14 La variations des RMS du couple (commande) en fonction de masse de charge

Les variations de la valeur RMS du couple dans le cas des régulateurs TDC (SM Jin) puis TDC (SM Slotine) puis TDC (PDF) et enfin PDF sont très petites pour des variations de masse assez importantes. Dans le cas de la commande TDC (PD), les variations du couple sont petites pour des masses inférieures à 1 Kg mais deviennent importantes pour les masses excédant 1 Kg. La commande la plus déficiente ici est le PD avec une augmentation considérable du couple suivant un profil linéaire de grande pente en fonction de la masse de la charge.

Remarque : La matrice \bar{M} est prise égale à $diag(35 \cdot 10^{-5}; 35 \cdot 10^{-5}; 35 \cdot 10^{-5})$.

3.5 Conclusion

Dans ce chapitre nous nous sommes intéressés au problème de commande du robot Delta. Pour ce faire, nous avons fait recours à plusieurs commandes à savoir PD, PD d'ordre fractionnaire et TDC sous plusieurs formes. L'étude a comporté le développement théorique, la simulation et la comparaison des résultats obtenus. La commande qui a donné les meilleures performances et robustesse est la commande TDC avec mode de glissement de surface de Jin. L'explication des résultats obtenus est développée dans le chapitre présent.

Chapitre 4

Expérimentation

4.1 Introduction

Dans cette dernière partie, nous allons nous intéresser au problème d'implémentation des commandes proposées dans la partie précédente. Pour ce faire, une introduction à l'environnement d'implémentation et aux considérations pratiques est faite, suivie de l'illustration de la trajectoire utilisée et programmée préalablement. Ensuite les figures illustrant les résultats d'erreurs de poursuite articulaires et commandes (couples) sont données et analysées. Enfin, des tests de robustesse sont faits.

4.2 Considérations pratiques

Pour passer de la simulation à la pratique nous devons mettre à l'esprit un ensemble de considérations pour la préservation du matériel (le robot) du laboratoire de commande des processus (LCP), et afin d'atteindre l'objectif de commande. Parmi ces considérations :

- Se documenter sur l'environnement avant de manipuler.
- La simulation est faite sur un modèle qui ne représente pas parfaitement le système réel.
- Le langage de programmation lors de la simulation diffère du langage de programmation utilisé en implémentation.
- S'assurer que les tensions de commande ne sont pas supérieures aux tensions maximales des moteurs et ne présentent pas de broutement ('chattering').
- Utiliser le bouton d'arrêt d'urgence en cas d'anomalies.
- S'assurer que les unités de mesure utilisées dans l'environnement d'implémentation correspondent aux unités de simulation ou dans le cas contraire effectuer les conversions nécessaires comme pour les cas suivants :

- Du couple à la tension [5] :

$$\tau(N.m) = 3.08 \times u(v)$$

- Du radian aux degrés :

$$\theta(rad) = \frac{\pi}{180} \times \theta(deg)$$

- Le temps d'échantillonnage choisi doit prendre en considération les temps d'acquisition des capteurs (en prend le cas le plus défavorable). Dans notre cas $T_s = 0.001s$.

4.3 Environnement d'implémentation

Le robot DELTA ISIR 88 du laboratoire LCP de l'école polytechnique 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.

La commande est implantée sur ce PC (Core Duo 2,8 Ghz) avec Windows XP et une extension en temps réel (RTX de IntervalZeroInc). Les actionneurs du robot sont des moteurs DC brushless (minertiamotor mini series UGTMEM-03LB2), avec un rapport de transmission égal à 12. L'algorithme de commande est programmé en langage C++ et exécuté avec une fréquence d'échantillonnage de 1kHz.

Sur le plan Software, **FlexCom** qui est une plateforme de programmation des lois de commande à implémenter et des trajectoires existe et jouit d'une simplicité de modification des programme remarquable ("ouverte") [25][26].

La description des outils de programmation ainsi que les étapes de mise en marche sont données dans [5].

Le langage de programmation utilisé dans la plateforme FlexCom est le C++. Lors de la simulation nous avons utilisé des fonctions MATLAB, donc nous avons dû les convertir vers le C++ tantôt à l'aide de MATLAB C CODER, tantôt en la reprogrammant de nouveau en C++.

4.4 Trajectoire de référence implémentée

Cette trajectoire est complexe. Elle est une combinaison de plusieurs trajectoires l'une d'entre elles est de type prise dépose mentionnée au chapitre 1. Une représentation graphique de cette trajectoire dans l'espace opérationnel est illustrée dans la figure 4.1 :

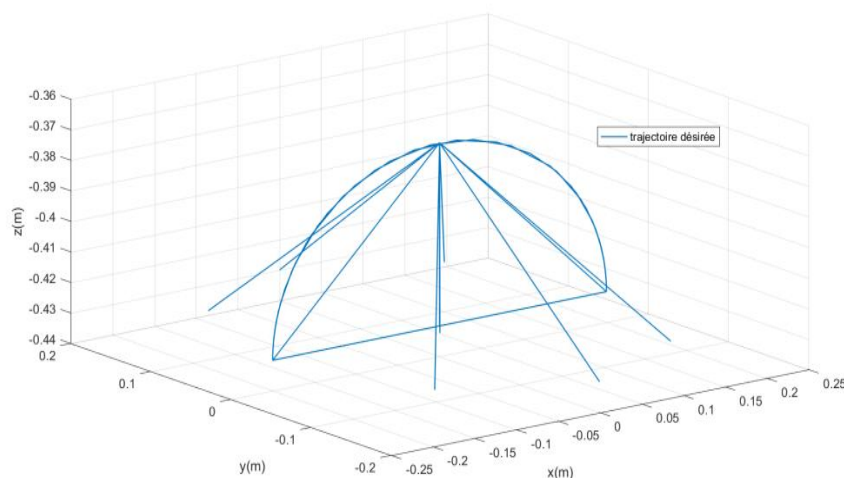


Fig. 4.1 Trajectoire de référence

4.5 L'implémentation des lois de commande :

Afin de valider les lois de commande développées dans le chapitre 03, et visualiser leurs performances, nous allons implémenter ces lois de commande sur le Robot Delta ISIR 88 et chaque loi sera testée par trois tests différents. Le premier, librement sans masse et dans les deux autres nous allons introduire une variation paramétrique par deux masses différentes.

Les paramètres des régulateurs que nous allons implémenter sur le Robot Delta ISIR 88 sont donnés dans le tableau 4.1.

Régulateur	Paramètres
Proportionnel Dérivée (PD) implémenté par défaut dans le Robot Delta.	$K_p = 2.2$ $K_d = 0.0145$
Proportionnelle Dérivé d'ordre Fractionnaire (PDF).	$K_p = 2.2$ $K_d = 0.0145$ $r = 0.91$
Time Delay Control avec servo-régulateur PD TDC(PD) .	$K_p = 1600$ $K_d = 80$
TDC avec mode de glissement utilisant la surface de Slotine TDC (SM Slotine) .	$\lambda = 200$ $K = 80$
TDC avec mode de glissement utilisant la surface de Jin TDC (SM Jin) .	$K_p = 1600$ $K_d = 80$ $\gamma = 0.6$ $\beta = 80$

Tableau 4.1 Les paramètres des régulateurs

Pour les commandes implémentées avec la technique TDE, les paramètres suivants sont les mêmes pour toutes les versions.

$$L = 0.001(sec) ;$$

$$\bar{M} = diag(36.10^{-6}; 36.10^{-6}; 36.10^{-6}).$$

4.6 résultats expérimentaux

4.6.1 Commande PD

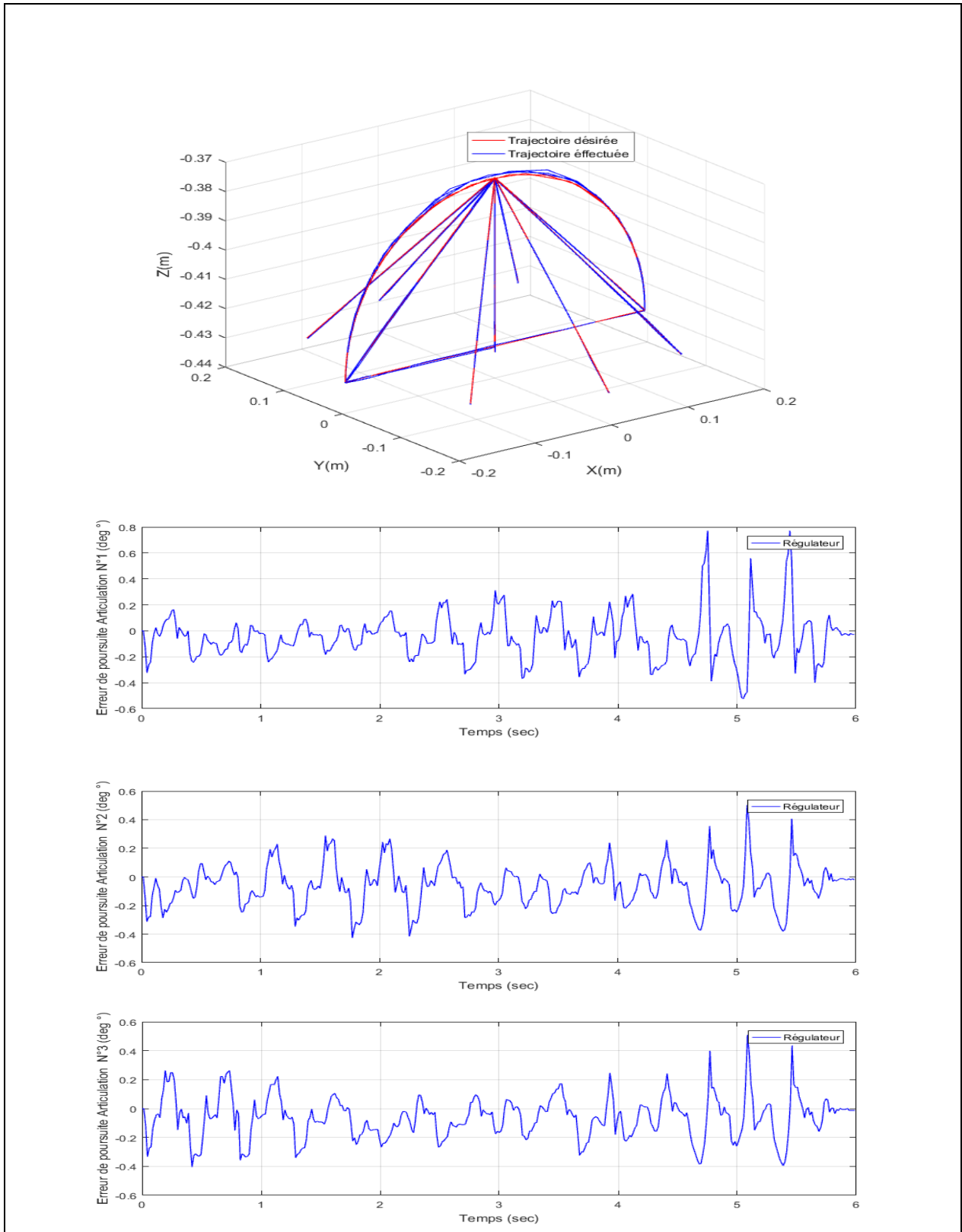


Fig.4.2 Résultats expérimentaux montrant la trajectoire effectuée et les erreurs de poursuite articulaires dans le cas du régulateur PD.

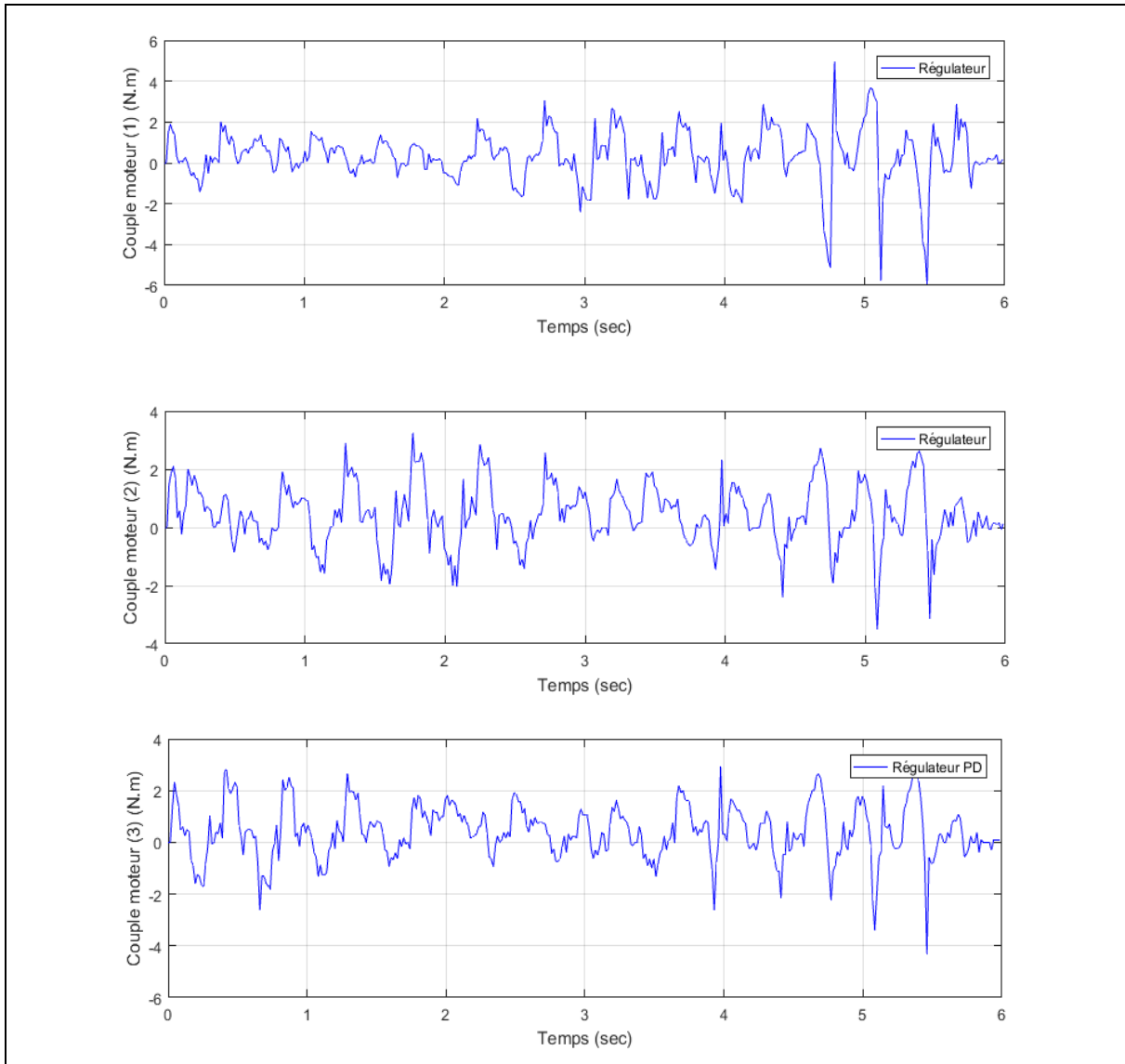


Fig.4.3 Résultats expérimentaux montrant les commandes en couple du PD

Le robot Delta commandé avec le régulateur PD suit la trajectoire désirée (Fig.4.2) d'une manière assez acceptable (la valeur RMS de l'erreur est 0.17°) malgré sa structure très simple, mais il existe des phases où l'erreur de poursuite (Fig.4.2) atteint des grandes valeurs (un max de 0.772° pour l'articulation 1), ses phases correspondent à des changements brusques d'accélération. Ce qui peut poser des problèmes dans les applications nécessitant la précision. La commande ne présente pas des grands pics ou des broutements (Fig.4.3). Elle est loin de la valeur maximale acceptée qui est de 15.4 N.m.

4.6.2 Commande PD d'ordre fractionnaire

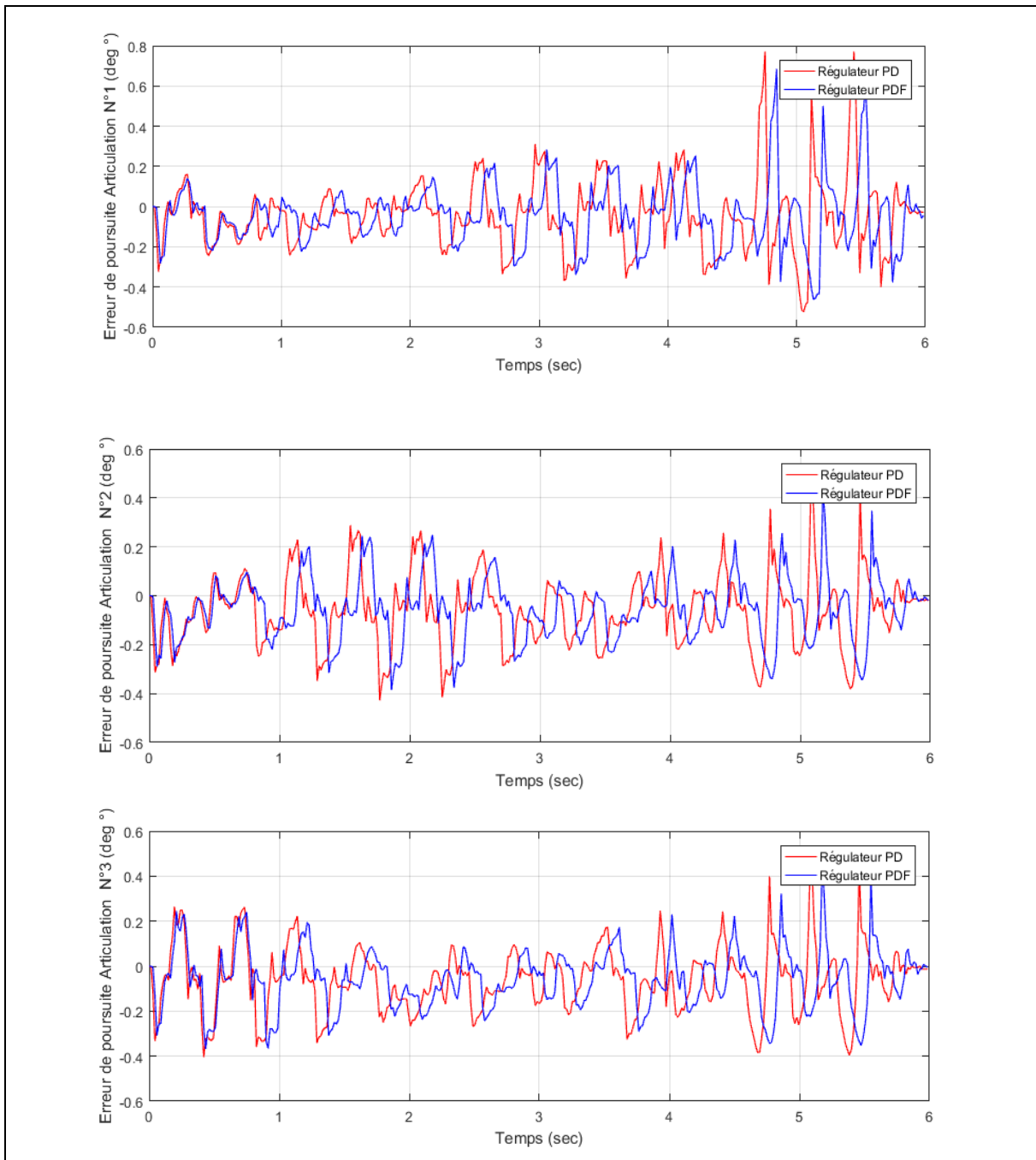


Fig.4.4 Comparaison des erreurs de poursuite articulaires de PD et PDF.

En utilisant le régulateur PD d'ordre fractionnaire, il n'y a pas eu de grande amélioration globalement (Fig.4.4). Toutefois, Les erreurs de poursuite articulaires ont légèrement diminué. En effet, les valeurs maximales pour le PD d'ordre fractionnaire sont 0.761° , 0.433° et 0.454° , inférieures à celles du PD classique 0.772° , 0.502° et 0.510° pour les articulations 1, 2 et 3 respectivement.

Les commandes (Fig.4.5) de ces deux régulateurs ont presque le même profil mais nous pouvons considérer la commande de la version fractionnaire meilleure du fait qu'elle a une valeur RMS 1.193 N.m inférieure à celle de la version classique 1.201 N.m.

Dans la simulation (section 3.2.2), nous avons obtenu des résultats nettement meilleurs que ceux de l'expérimentation. Ce qui peut être justifié par l'introduction d'erreurs lors de l'approximation de la dérivée fractionnaire en implémentation (difficulté de spécification de la plage de fréquences, ordre réduit de l'approximation).

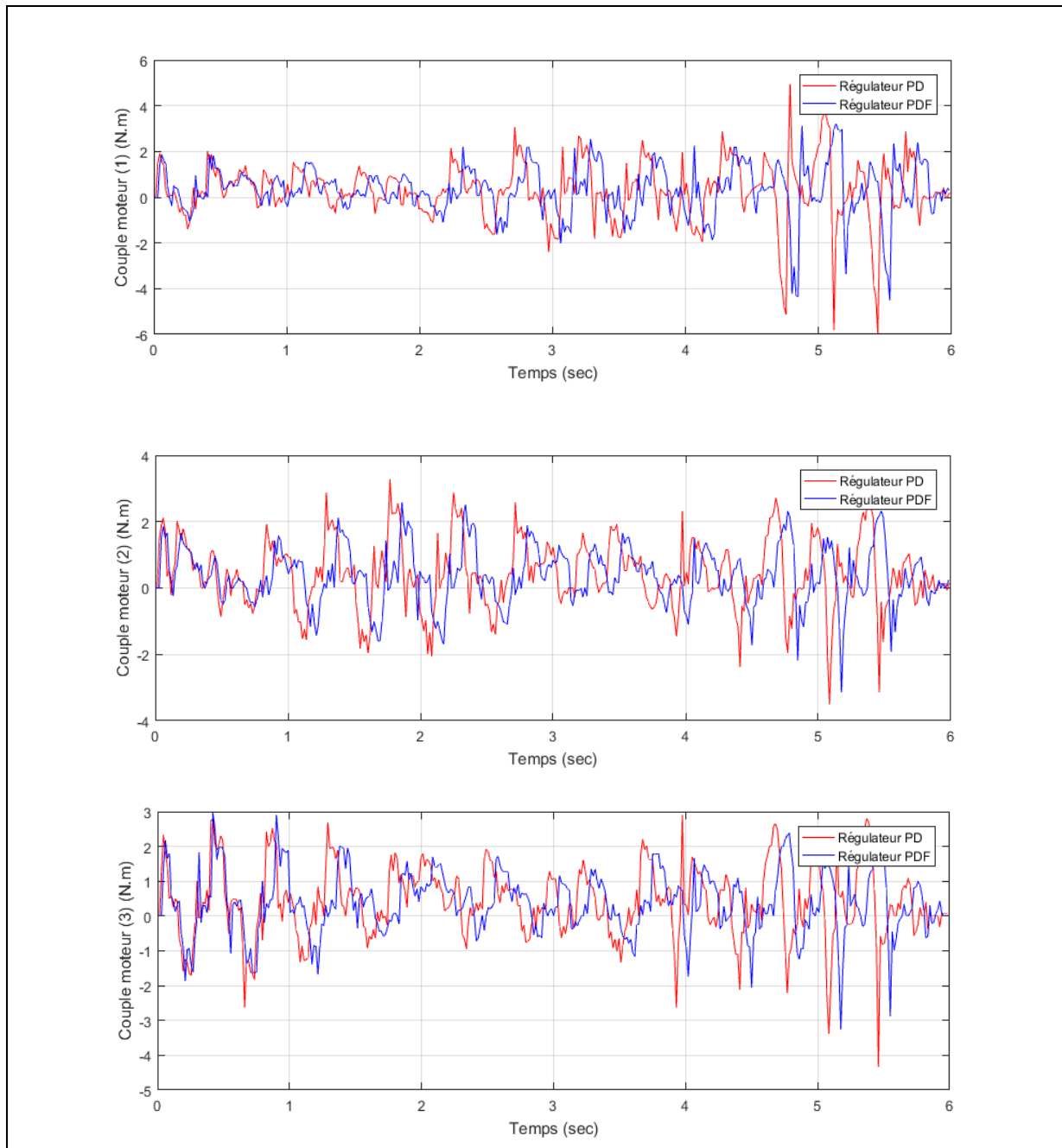


Fig.4.5 Comparaison entre les commandes en couple de PD et PDF.

4.6.3 Commande TDC avec PD

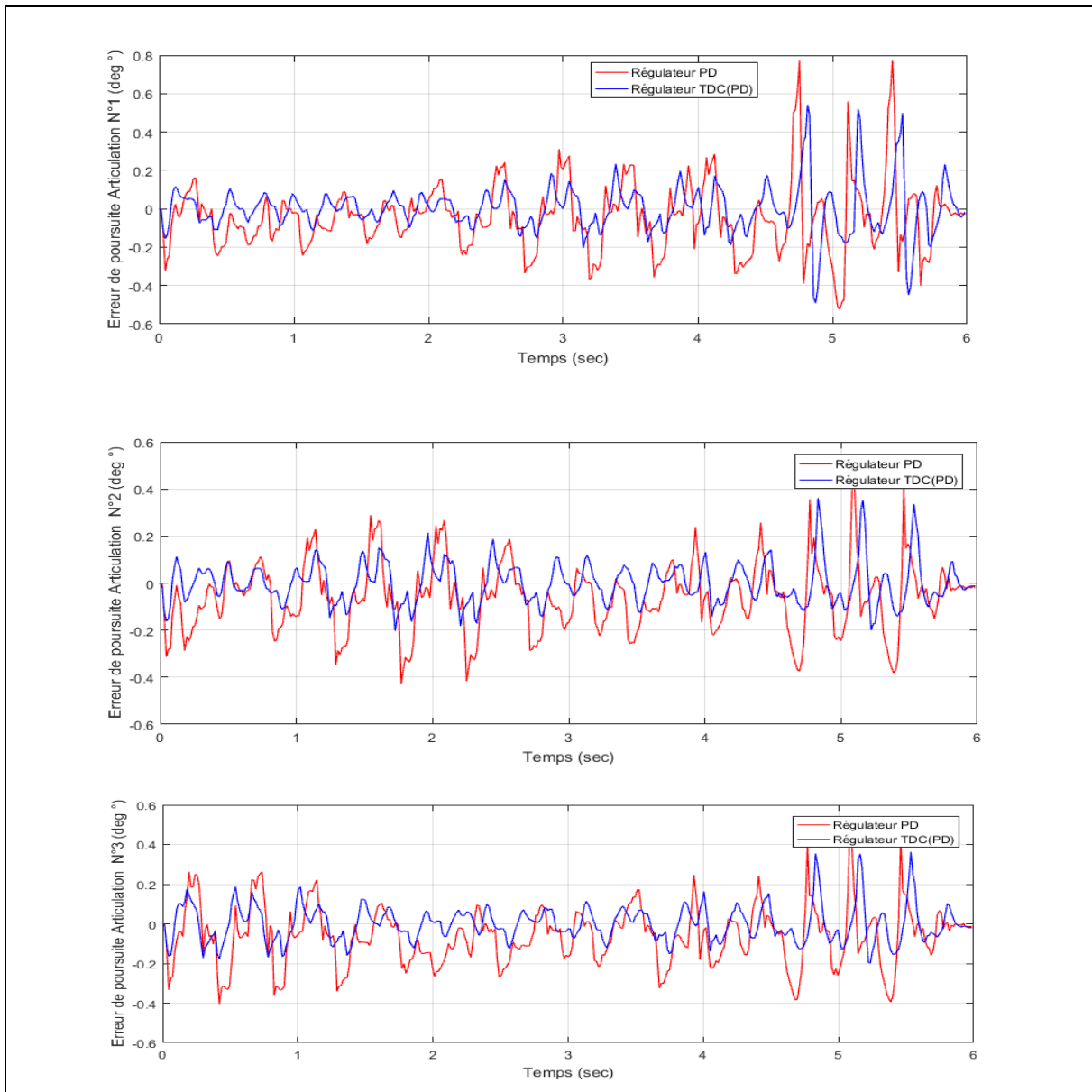


Fig.4.6 Comparaison entre PD et TDC (PD) : les erreurs de poursuite articulaires

L'effet de l'estimateur TDE chargé d'évaluer le modèle en temps réel et de compenser les non linéarités est clair dans les résultats obtenus. En effet, les erreurs de poursuite (Fig.4.6) ont considérablement diminué pour avoir pour valeurs maximales 0.579° , 0.432° et 0.501° pour les articulations 1,2 et 3 respectivement. En outre, la valeur RMS de l'erreur est de 0.114° contre 0.170° pour celle du PD. Quant à la commande, on remarque une légère augmentation dans le cas de TDC (avec servo-régulateur PD), sa valeur RMS égale 1.231 N.m , est supérieure à celle du régulateur PD classique (1.201 N.m).

La commande TDC (Fig.4.7) ici utilisée a donné de meilleurs résultats en poursuite pour une petite augmentation de la commande tout en se passant de la connaissance du modèle.

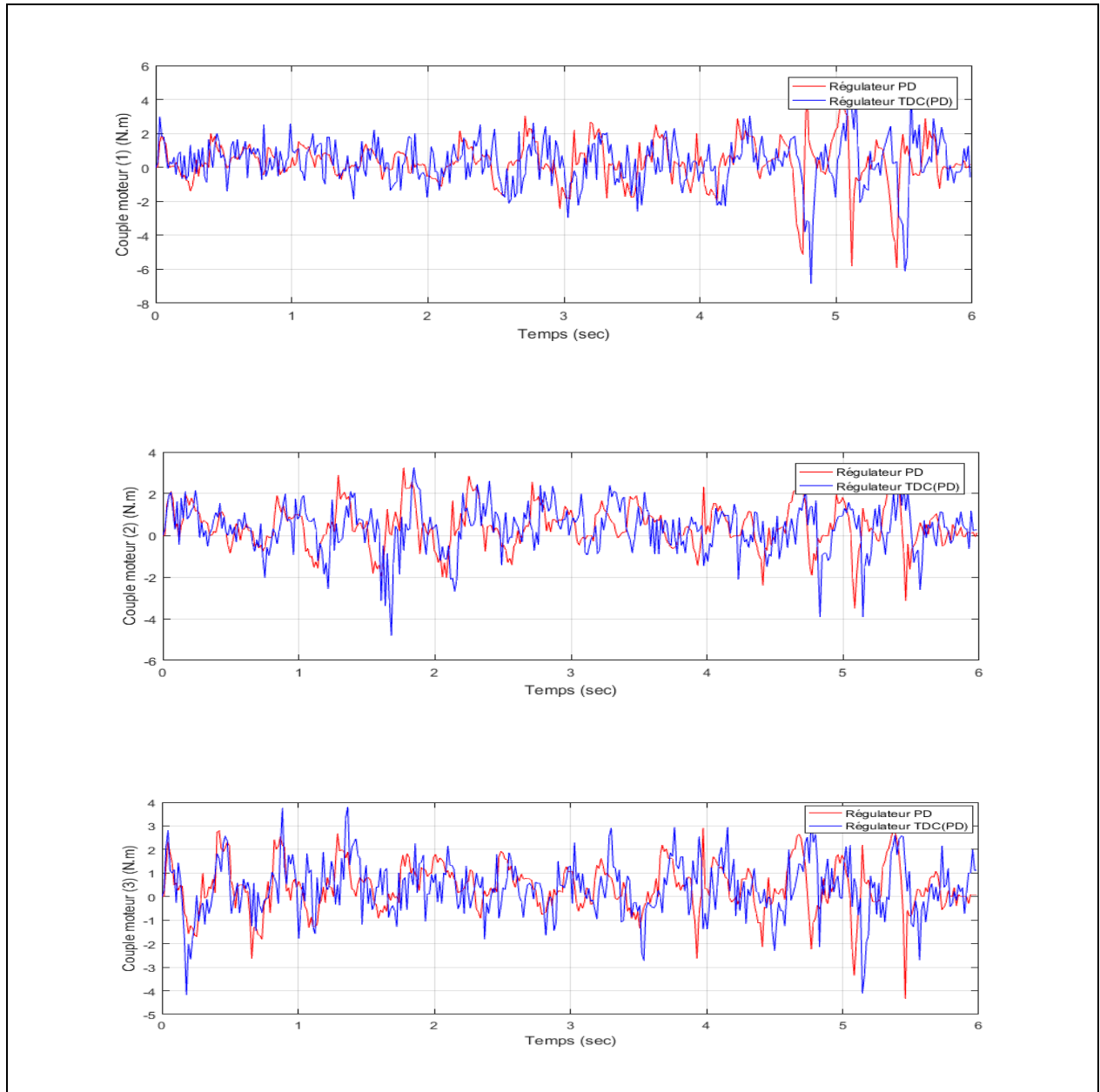


Fig.4.7 Comparaison entre les commandes en couple de PD et TDC (PD).

4.6.4 Commande TDC avec mode de glissement (surface de Slotine)

L'apport de cette approche (ajout d'un élément de robustesse) est la compensation des non-linéarités dures causées par les changements brusques et qui dégradent les performances de l'estimation TDE. Ceci est remarquable dans les résultats obtenus. En fait, les erreurs de poursuite articulaires ont incontestablement diminué, la différence est plus visible dans les pics de l'erreur de poursuite (Fig.4.8) correspondant à des pics d'accélération dans la trajectoire de

référence (changements brusques). Les valeurs maximales des erreurs sont 0.443° , 0.224° et 0.261° pour les articulations 1,2 et 3 respectivement. Elles sont les plus petites jusqu'à présent enregistrées avec une valeur RMS de 0.056° . Concernant la commande, les couples (Fig.4.9) ont aussi diminué, leur valeur RMS est de 1.208 N.m.

Il est à noter que l'implémentation n'a pas été faite avec la fonction signe pour les raisons évoquées dans la partie simulation.

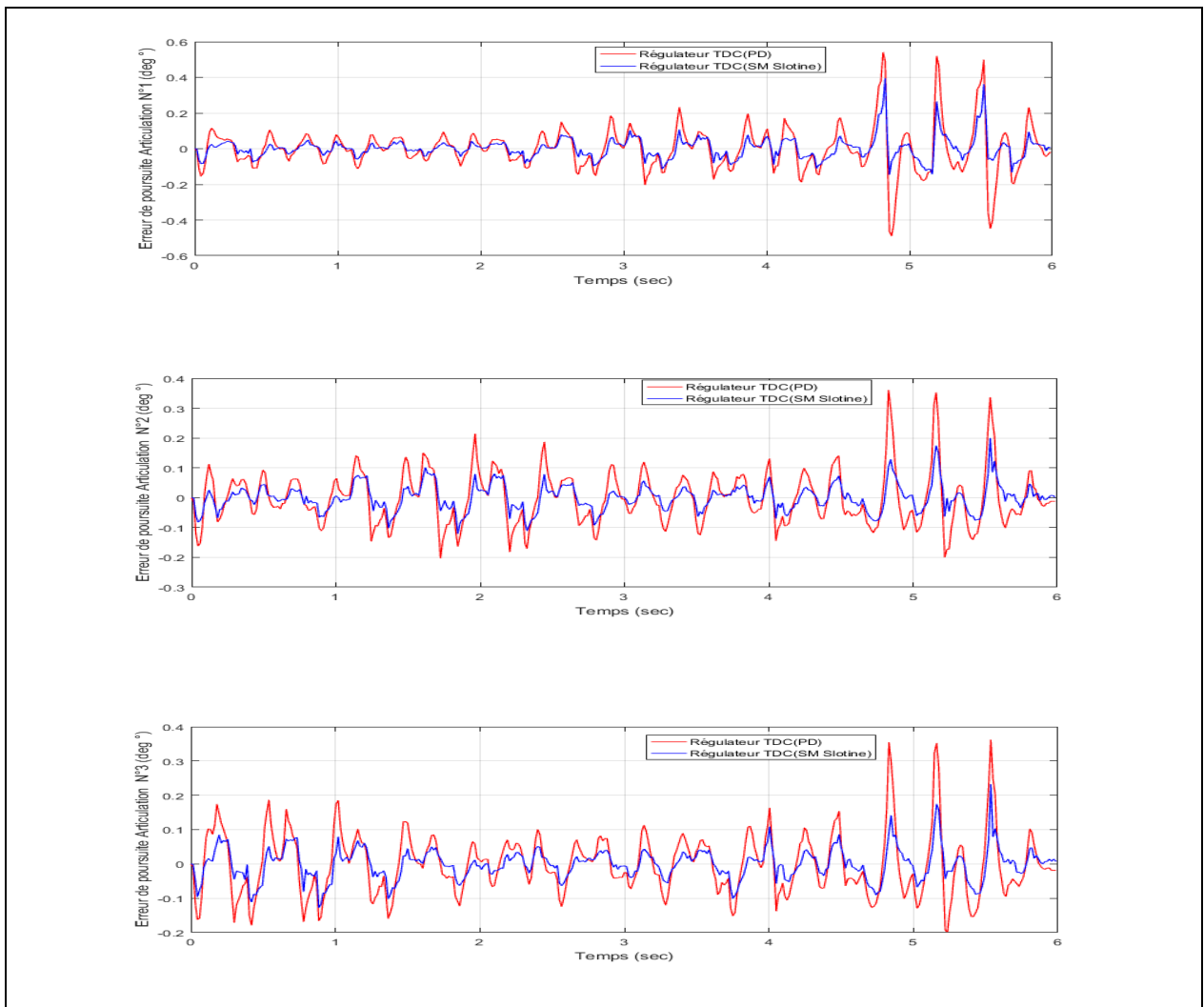


Fig.4.8 Comparaison des erreurs de poursuite articulaires entre TDC(PD) et TDC (SM, Slotine):

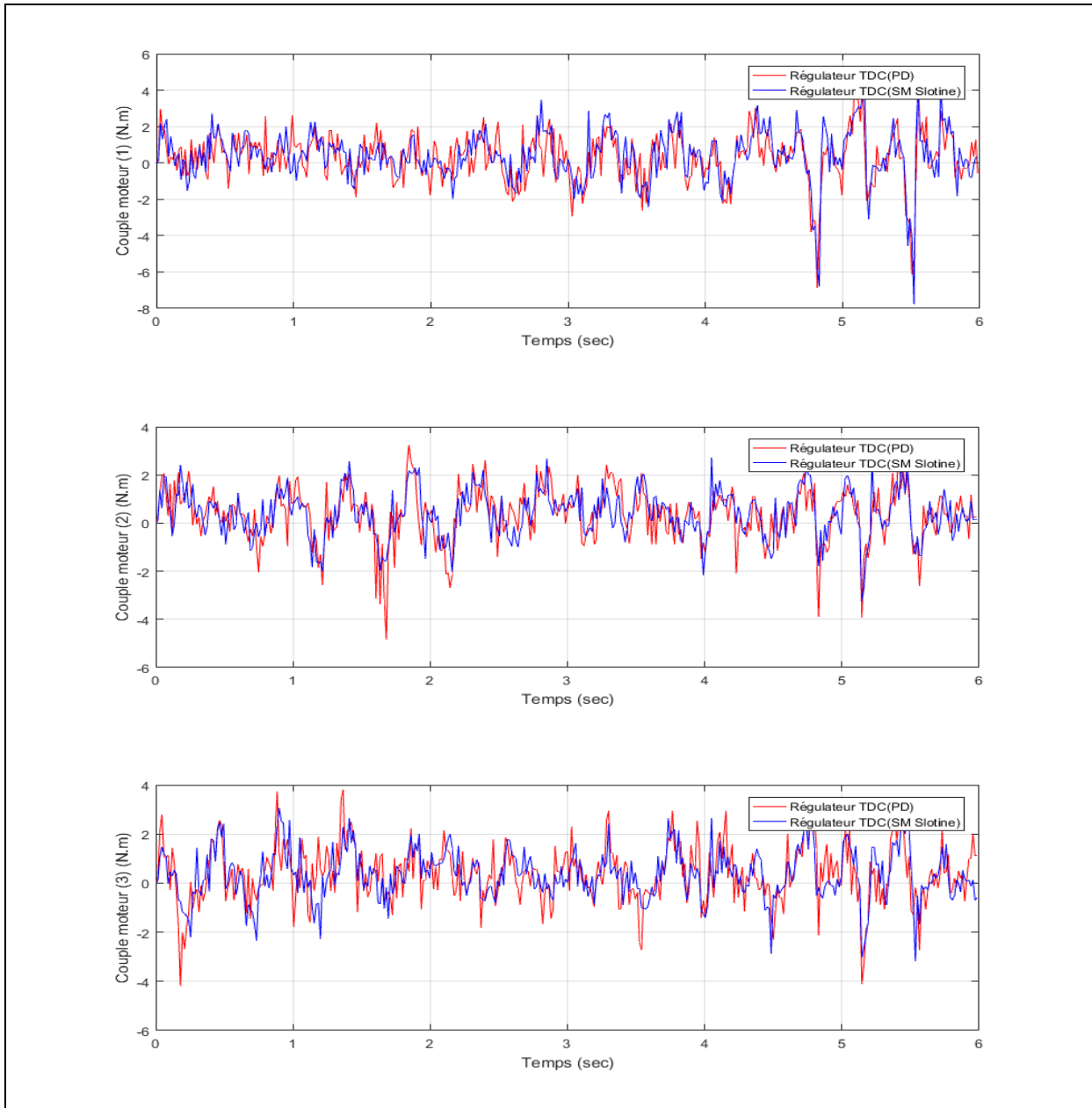


Fig.4.9 Comparaison des commandes en couple entre TDC(PD) et TDC (SM, Slotine):.

4.6.5 Commande TDC avec mode de glissement (surface de Jin)

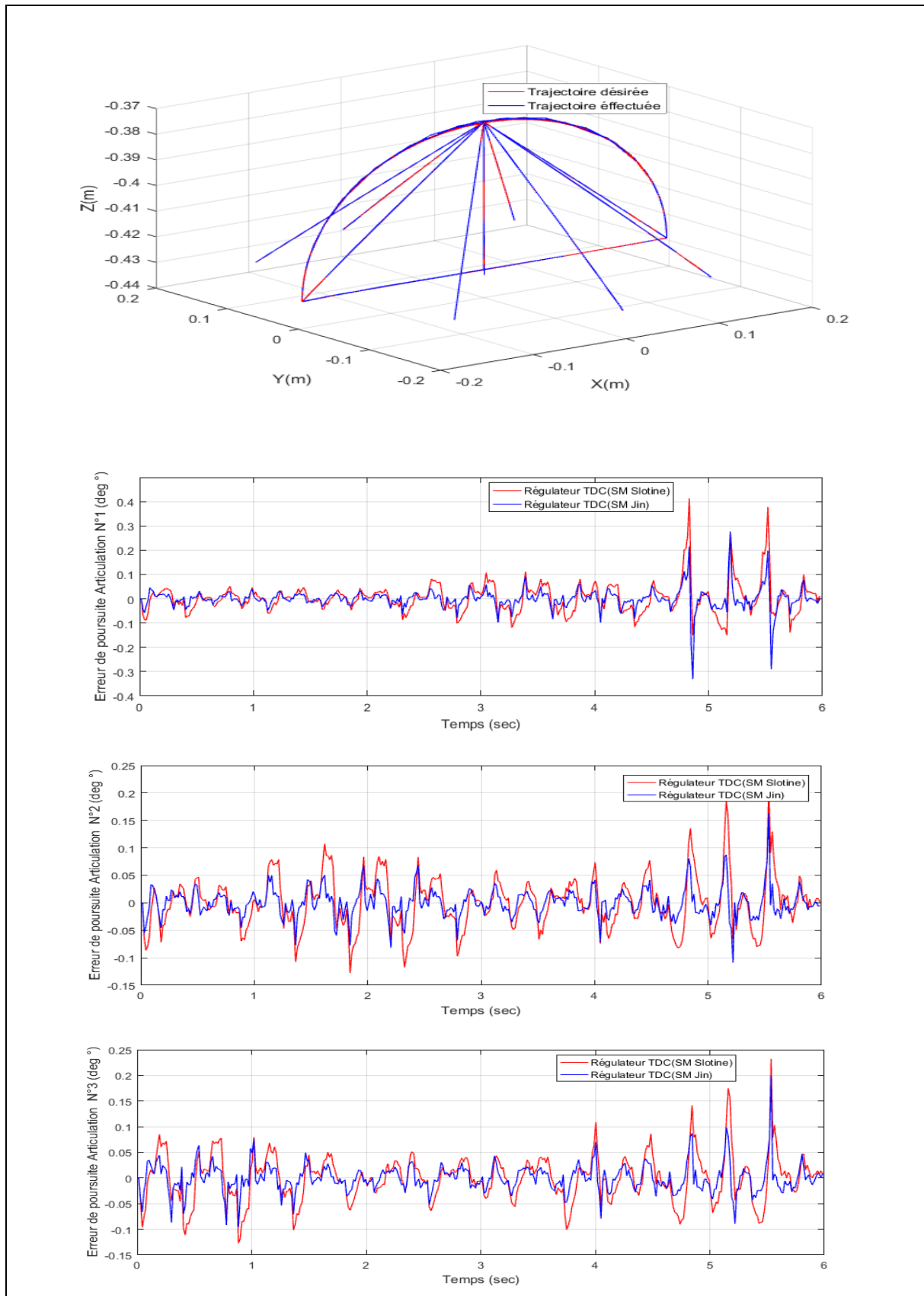


Fig.4.10 Comparaison entre les erreurs de poursuite articulaires de TDC (SM, Slotine) et TDC (SM, Jin) et trajectoire effectuée TDC (SM, Jin)

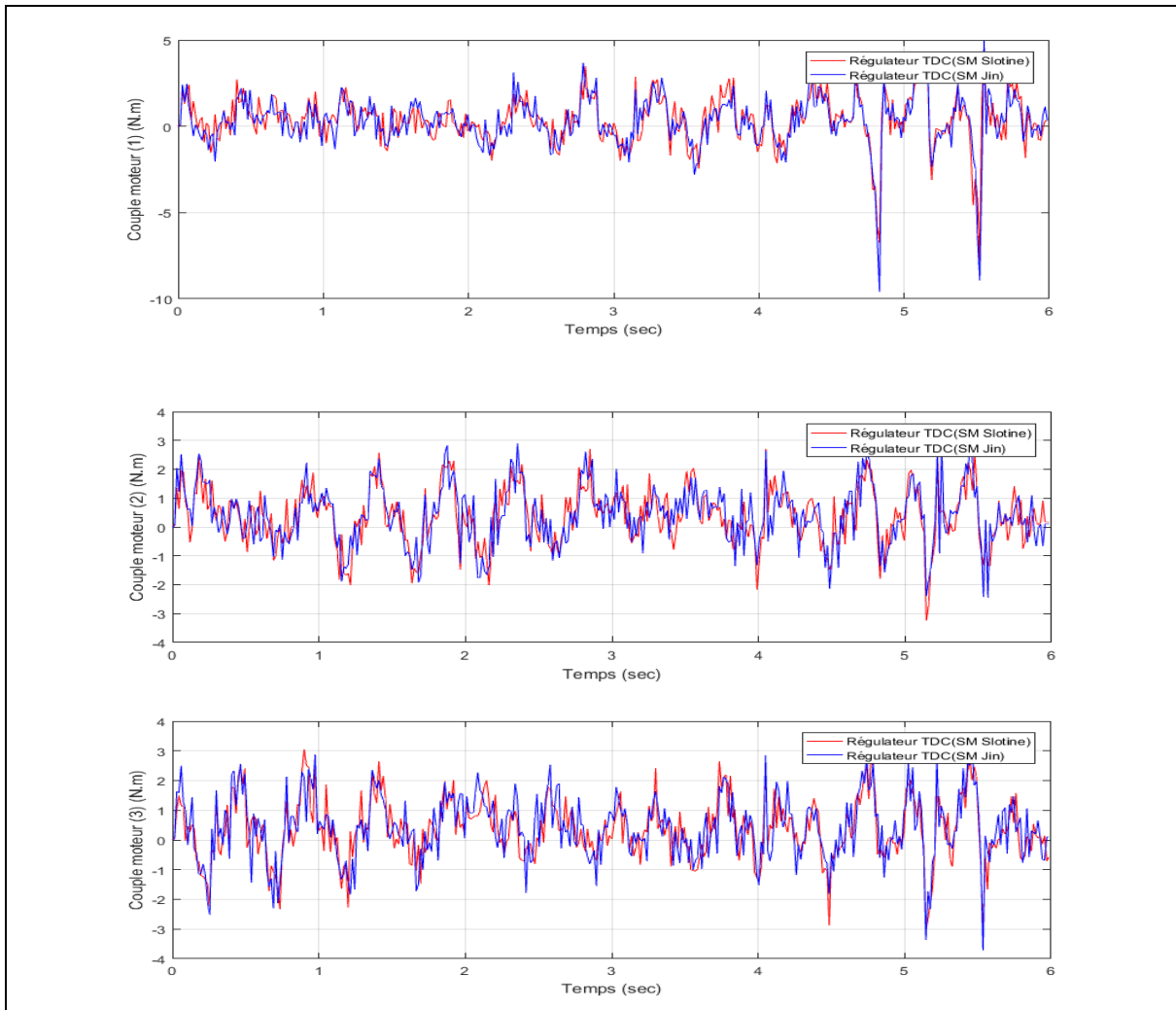


Fig.4.11 Comparaison des commandes en couple entre TDC(SM, Slotine) et TDC (SM, Jin)

La particularité de cette commande est le temps de convergence de l'erreur de poursuite vers l'origine qui est supposé être très minimale vue l'astuce utilisée (section 3.3). En effet, l'expérimentation vient confirmer ceci. Les erreurs de poursuite articulaires (Fig.4.10) ont encore diminué par rapport à la commande précédente (avec surface de Slotine), donc on vient d'enregistrer de nouveau les meilleures performances. Les valeurs maximales des erreurs sont 0.296° , 0.173° et 0.21° pour les articulations 1,2 et 3 respectivement. Ses bonnes performances dans l'espace articulaire se sont traduites par une bonne poursuite de la trajectoire de référence dans l'espace opérationnel comme montrée dans la figure 4.10 Cette commande est donc très adaptée aux applications nécessitant la précision. Concernant les couples, ils sont presque identiques à ceux développés par les moteurs dans le cas de la commande (Fig.4.11) précédente avec surface de Slotine. La valeur RMS de la commande est en fait égale à 1.242 N.m.

4.6.6 Tests de robustesse :

Dans le souci de tester la robustesse des commandes synthétisées, nous allons faire subir au robot une variation paramétrique (le paramètre en question est la masse de la nacelle) en lui ajoutant des charges et nous allons examiner son effet tout en gardant la même commande. Le tableau 4.2 montre les résultats des essais effectués.

Régulateur	Critère	N° d'articulation	PD	PDF	TDC (PD)	TDC (SM Slotine)	TDC (SM Jin)
<i>m=0 g</i>	Erreur max (deg°) Articulation	(1)	0.772	0.761	0.579	0.443	0.296
		(2)	0.502	0.433	0.432	0.224	0.173
		(3)	0.510	0.454	0.501	0.261	0.210
	RMS Erreur (deg°)	(1,2,3)	0.170	0.175	0.114	0.056	0.035
	RMS couple (N.m)	(1,2,3)	1.201	1.193	1.231	1.208	1.242
<i>m=46 g</i>	Erreur max (deg°) Articulation	(1)	0.856	0.803	0.653	0.528	0.332
		(2)	0.548	0.494	0.518	0.246	0.168
		(3)	0.541	0.497	0.481	0.268	0.214
	RMS Erreur (deg°)	(1,2,3)	0.190	0.188	0.126	0.061	0.039
	RMS couple (N.m)	(1,2,3)	1.285	1.283	1.271	1.302	1.314
<i>m=114 g</i>	Erreur max (deg°) Articulation	(1)	0.936	0.881	0.705	0.681	0.548
		(2)	0.568	0.531	0.512	0.291	0.179
		(3)	0.589	0.571	0.549	0.305	0.223
	RMS Erreur (deg°)	(1,2,3)	0.210	0.200	0.153	0.073	0.048
	RMS couple (N.m)	(1,2,3)	1.447	1.436	1.453	1.480	1.474

Tableau 4.2 Résultats des tests de robustesse aux variations paramétriques

Le premier résultat qu'on peut tirer est le fait que toutes ses commandes peuvent être considérées robustes aux variations de masse ici testées. Il n'y a pas de grandes variations des erreurs, la plus grande enregistrée est de 0.085° pour la première articulation dans le cas de la variation de la masse de charge de 0g à 46g et 0.21° de 46g à 114 g.

Pour les deux masses 46 g et 114 g, les valeurs RMS de l'erreur diminuent en avançant dans le tableau de gauche à droite alors que les valeurs RMS de la commande ne varient pas grandement donc les performances enregistrées en essais libres sont les mêmes que ceux en charge.

La commande TDC utilisant le mode glissant avec surface de Jin étant celle qui a donné les meilleures performances, nous allons dans ce qui suit illustrer ses résultats de robustesse.

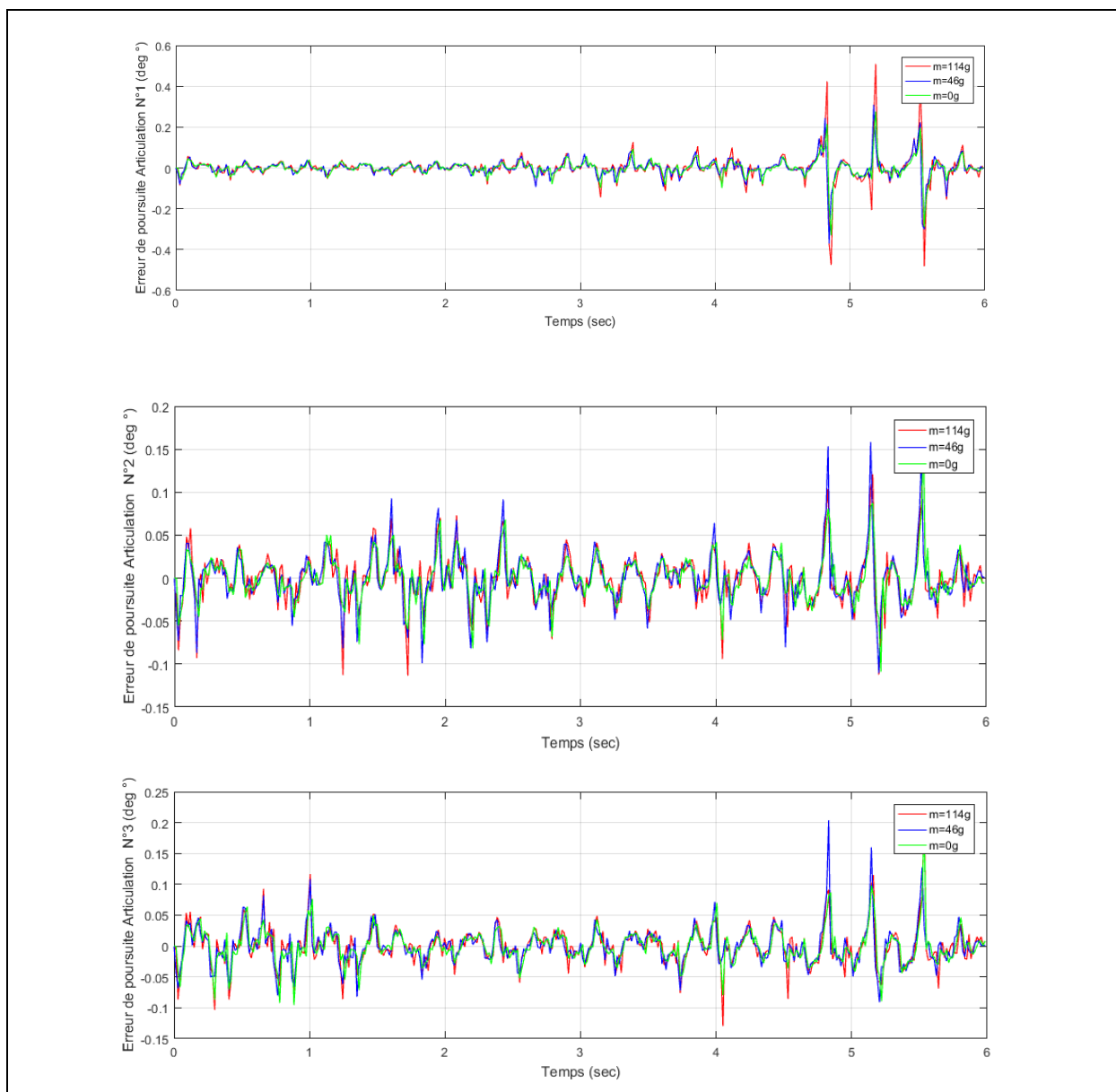


Fig.4.12 Erreurs de poursuite articulaires pour les différents essais en masses (TDC-JIN)

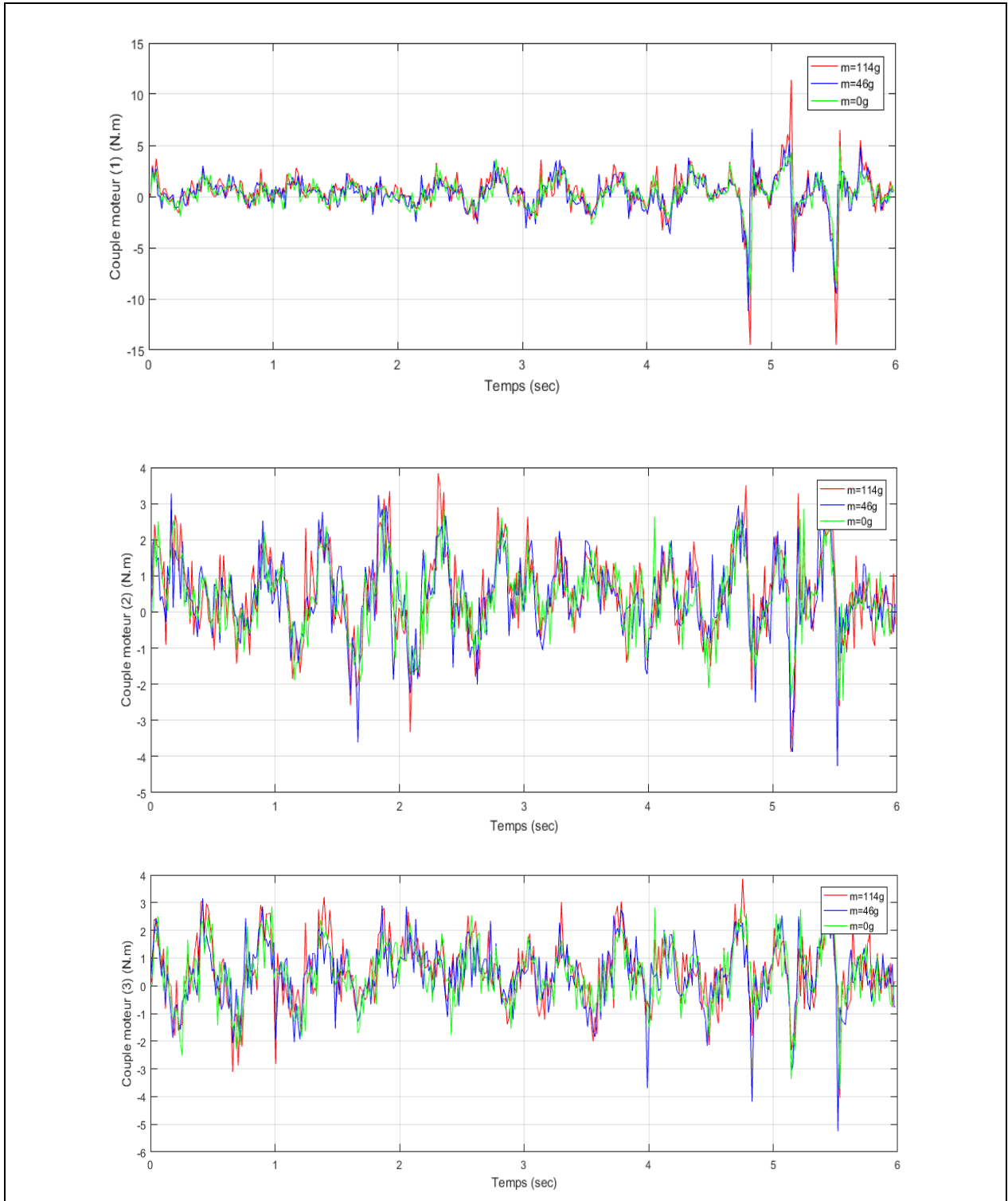


Fig. 4.13 Couples pour les différents essais en masses (TDC-JIN)

Nous remarquons que pour les erreurs de poursuite articulaires et les couples, il n'y a pas de grandes différences dans les essais libres ou avec charge à part dans les pics (variations brusques). On peut confirmer que les performances de la commande TDC avec mode de

glissement de surface de Jin ne se dégradent pas dans le cas de variation paramétrique (commande robuste).

4.7 Conclusion

Dans ce chapitre, nous avons abordé le problème de l'expérimentation des commandes synthétisées au chapitre précédent. Le passage de la simulation à l'expérimentation nécessite la prise en considération des différences existantes entre les deux environnements, ce qui a été traité en premier lieu. Les commandes expérimentées sont le PD, le PD d'ordre fractionnaire et la TDC. Ensuite, les résultats obtenus sont analysés, ils sont assez proches de ceux de la simulation. La commande ayant donné les meilleures performances est la commande TDC avec mode de glissement de surface non linéaire de Jin. En effet, la combinaison de l'estimation TDE avec le servo-régulateur par mode de glissement donne une commande qui est non seulement sans modèle (modèle estimé par TDE) mais aussi énergétiquement économe (la multiplication par \bar{M} qui est petite dans la formule du couple). De plus, elle permet d'atteindre une convergence rapide donc une poursuite plus précise. A la fin du chapitre, des tests de robustesses ont été faits et ont montré que les commandes synthétisées sont robustes par rapport aux variations paramétriques testées du moment que la poursuite de la trajectoire reste assurée.

Conclusion Générale

Notre travail, comme l'intitulé du PFE l'indique, porte sur la modélisation et la commande du robot Delta ISIR 88 disponible au labo LCP de l'ENP. Ce travail s'est fait en deux étapes principales à savoir la simulation sous MATLAB /SIMULINK puis l'expérimentation sur le robot réel.

La modélisation du robot Delta occupe une place importante que ce soit dans la simulation de son fonctionnement ou dans sa commande. Comme pour tous les robots manipulateurs, elle comporte la modélisation géométrique, cinématique et dynamique. Le robot Delta est de structure parallèle complexe, d'où l'absence de méthodes systématique permettant sa modélisation comme pour le cas des robots sériels. Toutefois, des travaux portant sur la modélisation du robot Delta ont déjà été faits. Les modèles que nous avons développés se basent essentiellement sur les travaux de Codourey en prenant en considération les différences existantes (à l'exemple des réducteurs qui engendrent des modifications dans le modèle dynamique). La particularité de cette modélisation est sa simplicité en dépit de la complexité du Delta. Elle tire des hypothèses simplificatrices auxquelles elle fait appel. Cette simplicité se paie au prix de la nécessité d'application de commande robuste pour compenser les incertitudes introduites.

Quant à la commande du Delta, nous avons commencé par simuler et expérimenter le régulateur PD implémenté par défaut dans le robot. Ensuite, nous avons utilisé d'autres stratégies de commande à l'instar de la commande PD d'ordre fractionnaire et la commande TDC toutes deux connues pour être robustes. Les développements théoriques ont été faits où les notions de stabilité et de poursuite ont été abordées.

Nous avons obtenu des résultats très intéressants, surtout pour le cas de la commande TDC. En effet, les meilleures performances ont été enregistrées dans le cas de la commande combinant la technique d'estimation TDE avec la stratégie des modes glissants utilisant la surface non linéaire de Jin. Ses résultats sont justifiés par l'efficacité de la TDE dans l'estimation des non-linéarités, de dynamiques inconnues et des perturbations puis dans leur compensation permettant ainsi de réaliser une commande sans connaissance même du modèle. De plus, le mode glissant est intrinsèquement robuste et à travers un choix approprié de la surface de glissement, il a été possible de considérablement diminuer le temps de convergence des erreurs de poursuite donc de la précision du robot dans le suivi de trajectoire.

La contrainte principale dans l'utilisation de la TDE est le choix d'un retard temporel assez petit pour lequel les signaux à estimer ne varient pas grandement. Le robot Delta est connu pour sa rapidité, ses dynamiques à variations brusques risquaient de remettre en cause l'utilisation de la TDE. Tout de même, le numérique permet de travailler à haute fréquence et de ce fait choisir un retard temporel (identique au pas d'échantillonnage) suffisamment petit. Donc nous pouvons confirmer que la commande TDC peut être utilisée pour une large classe de systèmes.

En guise de perspectives, le travail que nous avons fait peut être la base pour concevoir ensuite un robot Delta et le commander avec la stratégie TDC ou pour appliquer la commande TDC sur d'autres systèmes.

Références Bibliographiques

- [1] Khalil, Wisamaet Dombre, Etienne : Modélisation, identification et commande des robots. Hermes Sciences Publications, Paris, 2^eédition, 1999.
- [2] V. Shekher , P.Rai et O.Prakash “Tuning and Analysis of Fractional Order PID Controller” International Journal of Electronic and Electrical Engineering. ISSN 0974-2174 Vol 5, No 1 (2012), pp. 11-21.
- [3] Abdelelah Kidher Mahmood ,“ Design Optimal Fractional Order PID Controller Utilizing Particle Swarm Optimization Algorithm and Discretization Method “(IJESE), ISSN: 2319–6378, Vol 1, Issue-10, August 2013.
- [4] Aleksei Tepljakov, “ Fractional-order Modeling and Control of Dynamic Systems “ , Springer Theses, DOI 10.1007/978-3-319-52950-9_8.
- [5] W.Hamouche et H.I.Kadri, Modélisation et commande du robot Delta ISIR88. Simulation et Experimentation, Projet de Fin d’étude, ENP, Alger 2017 .
- [6] N.I.Mokhtari et H.Nadour, Contribution à la modélisation et à la commande du robot delta ISIR88, Projet de Fin d’étude , ENP, Alger 2016.
- [7] R.Meryem, Modélisation et commande robuste de robot Delta, ENP Alger 2016.
- [8] K.Oualidet et A.Larbi, Modélisation et commande d’un robot parallèle de type delta, Projet de Fin d’étude, ENP, Alger2010.
- [9] C. E. Boudjedir, D. Boukhetala, and M. Bouri. "Iterative learning control of multivariable uncertain nonlinear systems with non repetitive trajectory". *Nonlinear Dynamics*, vol. 95, no.3, pp. 2197-2208, 2019.
- [10] T. C. Hsia, “A new technique for robust control of servo systems,” *IEEE Trans. Ind. Electron.*, vol. 36, no. 1, pp. 1–7, Feb. 1989.

- [11] K. Youcef-Toumi and O. Ito, "A time delay controller design for systems with unknown dynamics," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 112, no. 1, pp. 133–142, Mar. 1990.
- [12] M. Jin, S. H. Kang, and P. H. Chang, "Robust compliant motion control of robot with nonlinear friction using time-delay estimation," *IEEE Trans. Ind. Electron.*, vol. 55, pp. 258–269, Jan. 2008.
- [13] M. Jin, J. Lee, P. H. Chang and Ch. Choi, "Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control" *IEEE Trans. Ind. Electron.*, Vol. 56, no. 9, September 2009.
- [14] Wang Y, Yan F, Jiang S, Chen B. "Time delay control of cable-driven manipulators with adaptive fractional-order nonsingular terminal sliding mode". *Adv Eng Softw* Jul. 2018;121:13–25.
- [15] J.J. Slotine, W. Li : *Applied nonlinear control*. USA: Prentice Hall, 1991. 459p ISBN 0-13-040890-5.
- [16] M. Jin, Yi. Jin, P. H. Chang and Ch. Choi, "High -Accuracy tracking control of robot manipulators using time delay estimation and terminal sliding mode" *Int J Adv Robotic Sy*, 2011, Vol. 8, No. 4, 65-78.
- [17] M. Spong and M. Vidyasagar, "Robust linear compensator design for nonlinear robotic control," *IEEE J. Robot. Autom.*, vol. 3, pp. 345-351, 1988.
- [18] F. L. Lewis, *et al.*, *Control of Robot Manipulators*. New York: Macmillan Publishing Company, 1993.
- [19] S.-U. Lee and P. H. Chang, "Control of a heavy-duty robotic excavator using time delay control with integral sliding surface," *Control Eng. Pract.*, vol. 10, pp. 697-711, 2002.
- [20] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*, 2nd ed.: CRC Press, 2003.

- [21] CLAVEL,Reymond. Conception d'un robot parallèle rapide à 4 degrés de liberté.146 p. Thèse de Doctorat : microtechnique : Lausanne, EPFL : 1991.
- [22] Codourey, A., Dynamic Modelling and Mass Matrix Evaluation of the DELTA Parallel Robot for Axes Decoupling Control.
- [23] Sternheim, F., 1987. Computation of the direct and inverse geometric models of the DELTA4 parallel robot, Robotersysteme, Vol. 3, pp. 199-203.
- [24] Codourey, A., 1991. Contribution B la commande des robots rapides et précis. Application au robot DELTA B entraînement direct Thèse de Doctorat : EPFL, No. 922.
- [25] M.Bouri, FlexCom a Windows PC based robot controller an open architecture, Moveit Automation.
- [26] M.Bouri, Getting started with FlexCom, 2014.
- [27] N.Bouarroudj, Commande par mode de glissement d'ordre fractionnaire optimisées par métaheuristique. Application aux systèmes non linéaire et interconnectés, Thèse de Doctorat, ENP,Alger 2017.