

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'Etudes pour l'obtention du Diplôme
d'Ingénieur d'Etat en Automatique intitulé :

Modélisation et Commande du Robot Delta ISIR88. Simulation et Expérimentation

Présenté par

Walid HAMOUCHE
Hatem Ibrahim KADRI

Sous la direction de

Pr. Djamel BOUKHETALA
M. Hakim ACHOUR

Présenté et soutenu publiquement le 21/06/2017

Composition du jury :

Président	M. Rachid ILLOUL	Maître de Conférences A	ENP
Encadreur	M. Djamel BOUKHETALA	Professeur	ENP
Co-Encadreur	M. Hakim ACHOUR	Maître Assistant A	ENP
Examineur	M. Boualem HEMICI	Professeur	ENP

ENP 2017

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'Etudes pour l'obtention du Diplôme
d'Ingénieur d'Etat en Automatique intitulé :

Modélisation et Commande du Robot Delta ISIR88. Simulation et Expérimentation

Présenté par

Walid HAMOUCHE
Hatem Ibrahim KADRI

Sous la direction de

Pr. Djamel BOUKHETALA
M. Hakim ACHOUR

Présenté et soutenu publiquement le 21/06/2017

Composition du jury :

Président	M. Rachid ILLOUL	Maître de Conférences A	ENP
Encadreur	M. Djamel BOUKHETALA	Professeur	ENP
Co-Encadreur	M. Hakim ACHOUR	Maître Assistant A	ENP
Examineur	M. Boualem HEMICI	Professeur	ENP

ملخص

إن الهدف الذي ترمي إليه هذه المذكرة هو النمذجة و التحكم في الآلة المتوازية دلتا ازيير 88. فهي تعتبر آلة مشهورة اجتاحت في الآونة الأخيرة الساحة الصناعية نظرا لأدائها السريع و الدقيق. تبدأ الدراسة بنبذة نصف فيها بشكل عام الآلات والآلات المتوازية بشكل مستفيض. ثم تطرقنا بعدها إلى النمذجة الهندسية الحركية والديناميكية. قارنا بعد ذلك مختلف آليات التحكم (مع و بدون دمج النموذج الديناميكي المعكوس). أختبرنا مسارات مختلفة للإلتقاط والوضع و عرضنا نتائج المحاكاة والتجربة.

كلمات مفتاحية: الآليات المتوازية، الآلة دلتا، نمذجة و تحكم. ، التحكم في الآلات

Abstract

The aim of this work is the modeling and control of the parallel robot Delta ISIR88. This robot has gained a reputation of being fast and accurate and therefore it is highly recommended in pick and place operations in the industry. The study begins by an introduction about the different types of robots in general and then delves into the thorough details of parallel robots. Afterwards, a global modeling for the Delta robot ISIR88 is presented and validated. Three control structures are considered: open loop control, closed loop structure with and without the incorporation of the inverse dynamic model of the robot (IDM). Then, we proceeded by doing a comparison between the classic PD controller and the PD controller with the IDM. To ensure the utmost reliability of the work done, complex trajectories of pick and place have been tested. Simulation and experimental results are illustrated using Matlab-Simulink and FlexCom Development Kit.

Keywords: Parallel robots, Delta robot, modeling and control, Automatic control.

Résumé

Ce mémoire de Projet de Fin d'Etudes porte sur la modélisation et la commande du robot parallèle Delta ISIR88. Rapide et précis, il est utilisé dans plusieurs domaines d'industrie. L'étude débute par une introduction générale décrivant le travail effectué. Ensuite, les différents types de robots sont cités avec focalisation sur les robots parallèles. Par la suite, une modélisation globale du robot Delta ISIR88 a été faite et validée. Une fois le modèle conçu, une comparaison a été effectuée entre les différentes structures de commande et les régulateurs PD sans et avec l'incorporation du modèle dynamique inverse du robot ont été simulés. Notre étude a été poursuivie par l'implémentation de ces lois de commandes à l'aide du logiciel FlexCom pour différentes trajectoires de prise dépose. Une fois l'acquisition des résultats faite, le travail a été conclu par une étude comparative.

Mots-clés : Robots parallèles, Robot Delta, Modélisation et commande des robots, Asservissement.

Dédicaces

A mes parents pour leur soutien et encouragements.

A mes proches et à toute ma famille.

A mes amis et tous les gens qui m'aiment.

Au bonheur des plus chers, je dédie ce modeste travail.

Walid

Dédicaces

A mes chers parents.

A mes frères, Rami, Nadji et Ayoub.

A toute ma famille.

A tous mes amis qui m'ont soutenu.

Hatem Ibrahim

Remerciements

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

Nous tenons à remercier spécialement nos promoteurs M. Djamel BOUKHETALA et M. Hakim ACHOUR pour leur aide et leur disponibilité tout au long de la réalisation de ce travail.

Nous remercions également tous nos enseignants de l'Ecole Nationale Polytechnique qui nous ont tant donné tout au long de notre cursus. Nos profonds remerciements à M. Mohamed TADJINE et M. Chamseddine BOUEDJEDIR pour leurs précieux conseils.

Enfin, nos profonds remerciements s'adressent à nos familles, d'abord à nos chers parents qui n'ont jamais cessé de nous encourager et de nous soutenir et c'est grâce à leurs sacrifices et leurs patiences que ce travail a pu voir le jour.

Sommaire

Liste des tables	
Liste des figures.....	
Liste des acronymes.....	
Introduction Générale	15
Chapitre 1: Généralités sur la robotique	18
1.1. Introduction.....	18
1.2. Classification.....	19
1.2.1. Robots sériels :	21
1.2.2. Robots parallèles :	21
1.3. Modélisation et commande des robots	24
1.3.1. Structure de commande	24
1.3.2. Modélisation	25
1.3.3. Génération de trajectoire.....	28
1.3.4. Principales méthodes de commande.....	29
1.4. Conclusion.....	31
Chapitre 2: Le robot delta.....	33
2.1. Introduction.....	33
2.2. Historique.....	33
2.3. Applications.....	34
2.4. Le robot delta ISIR88.....	35
2.4.1. Description	35
2.4.2. Caractéristiques.....	38
2.5. Conclusion.....	40
Chapitre 3: Modélisation du robot delta ISIR88.....	42
3.1. Introduction.....	42

3.2. Modélisation géométrique.....	42
3.2.1. Indices et paramètres géométriques	42
3.2.2. Modèle géométrique direct « MGD ».....	44
3.2.3. Modèle géométrique inverse « MGI ».....	47
3.3. Espace de travail	50
3.4. Modélisation cinématique.....	52
3.4.1. Modèle Cinématique Direct « MCD »	52
3.4.2. Modèle Cinématique Inverse « MCI »	54
3.5. Singularités	55
3.6. Modèle des accélérations direct (MAD)	56
3.7. Modélisation dynamique.....	57
3.7.1. Modèle Dynamique Inverse « MDI »	64
3.7.1. Modèle Dynamique Direct « MDD »	66
3.8. Représentation d'état	66
3.9. Conclusion.....	68
Chapitre 4: Commande et Simulation.....	70
4.1. Introduction.....	70
4.2. Trajectoire de consigne.....	74
4.3. Commande avec régulateur PD.....	74
4.3.1. Commande avec régulateur PD classique	75
4.3.2. Commande avec régulateur PD et MDI à priori	76
4.3.3. Comparaison PD et PD-MDI.....	78
4.4. Tests de robustesse.....	78
4.5. Conclusion.....	80
Chapitre 5: Implémentation et Expérimentations.....	82
5.1. Introduction.....	82

5.2. Considérations pratiques	82
5.3. Environnement d'implémentation	84
5.4. Description des outils de programmation utilisés.....	86
5.5. Implémentation du régulateur PD et PD avec MDI.....	90
5.5.1. Méthodologie	90
5.5.2. Données enregistrées.....	90
5.5.3. Comparaison et discussion	95
5.6. Conclusion.....	95
Conclusion Générale.....	96
Bibliographie	98

Liste des tables

Tab.2.1 Caractéristiques Mécaniques du robot

Tab.2.2 Caractéristiques des actionneurs (Datasheet)

Tab.5.1 Comparaison Erreurs PD et PD MDI

Liste des figures

Fig.1.1 L'homme jouant l'instrument à vent

Fig.1.2 Le canard articulé

Fig.1.3 Le robot « Unimate » dans l'usine General Motors

Fig.1.4.a Robot B21

Fig.1.4.b Pioneer P3-DX

Fig.1.5 Robot manipulateur « SCARA »

Fig.1.6 Robot manipulateur dans un atelier de soudage

Fig.1.7 Différentes architectures de robots

Fig.1.8 Exemple d'une structure parallèle

Fig.1.9 Robot delta de l'EPFL

Fig.1.10 Robot MinAngle

Fig.1.11 Synoptique de commande

Fig.1.12 Système de commande en robotique

Fig.1.13 Etapes exécutées par le système de commande

Fig.1.14 Modèle géométrique

Fig.1.15 Modèle cinématique

Fig.1.16 Modèle Dynamique

Fig.1.17 Commande PID décentralisée

Fig.1.18 Schéma de commande en boucle ouverte avec MDI

Fig.1.19 Commande avec MDI sur Matlab

Fig.1.20 Commande avec MDI en boucle fermée

Fig.2.1.a Architecture du Delta

Fig.2.1.b Robot delta d'ABB

Fig.2.2.a Robot Delta en chirurgie

Fig.2.2.b Robot Delta en industrie alimentaire et pharmaceutique

Fig.2.2.c Robot Delta micro positionnement et impression 3D

Fig.2.3 Robot Delta du LCP – ENP ISIR88

Fig.2.4 Constituants du robot delta

Fig.2.5 Vue de haut de la nacelle

Fig.2.6 Vue de haut des barres parallèles

Fig.2.7 Espace de travail du Delta

Fig.2.8 L'arbre télescopique assurant le quatrième degré de liberté

Fig.2.9 Actionneur du Delta

Fig.3.1 Schématisation et placement des repères

Fig.3.2 Vues de face et de profil d'une chaîne « i » du robot delta

Fig.3.3 Schéma d'une chaîne « i » après simplification

Fig.3.4 Modèle géométrique direct du Delta

Fig.3.5 La sphère contenant les points C_i du Delta

Fig.3.6 Erreur entre MGD et SimMechanics

Fig.3.7 Modèle géométrique inverse du Delta

Fig.3.8 Erreur MGI

Fig.3.9 Espace de travail de chaque chaîne cinématique

Fig.3.10 Intersection des trois espaces de travail de chaque chaîne

Fig.3.11 Espace de travail du Delta

Fig.3.12 Espace de travail du Delta et ses paramètres

Fig.3.13 Erreur MCD

Fig.3.14 Singularités du Delta

Fig.3.15 Erreur en accélération entre le MAD et SimMechanics

Fig.3.16 Force de gravité et résultante dynamique de la nacelle

Fig.3.17 Force de gravité et résultantes dynamiques des barres parallèles

Fig.3.18 Simplification de la force de gravité des barres parallèles

Fig.3.19 Simplification de la résultante dynamique de translation des barres

Fig.3.20 Forces et résultantes dynamiques simplifiées des barres parallèles

Fig.3.21 Forces extérieures et résultantes dynamiques générales

Fig.3.22 Vecteurs unitaires portés sur les barres parallèles

Fig.3.23 décomposition des forces et des résultantes dynamiques

Fig.3.24 forces extérieures et résultantes dynamiques extérieures

Fig.3.25 Schéma de structure de validation du MDI

Fig.3.26 Couples donnés par les deux le MDI et SimMechanics

Fig.3.27 Erreur entre notre MDI et celui de SimMechanics

Fig.3.28 Réducteur mécanique

Fig.4.1 Schéma de commande en boucle ouverte

Fig.4.2 Suivi de trajectoire en BO

Fig.4.3 Schéma de commande avec régulateur

Fig.4.4 Suivi de trajectoire articulaire pour la commande avec régulateur
Fig.4.5 Erreur de Suivi de trajectoire pour la commande avec régulateur
Fig.4.6 Régulateur avec MDI à priori
Fig.4.7 Suivi de trajectoire pour la commande avec régulateur et MDI à priori
Fig.4.8 Erreurs de position pour les trois techniques de commande (en m)
Fig.4.9 Trajectoire proposée de prise-dépose
Fig.4.10.a Schéma de commande PD
Fig.4.10.b Schéma de commande PD avec MDI
Fig.4.11 Suivi de trajectoire PD
Fig.4.12 Erreur de suivi de trajectoire PD
Fig.4.13 Suivi de trajectoire PD+MDI
Fig.4.14 Erreur suivi de trajectoire PD+MDI
Fig.4.15 Les commandes des trois bras PD+MDI
Fig.4.16 Comparaison Erreurs PD et PD+MDI
Fig.4.17 Couples de perturbations
Fig.4.18 Suivi de trajectoire sous l'effet de perturbations
Fig.4.19 Erreur PD & PD MDI sous l'effet de perturbations
Fig.5.1 Schéma pratique général de commande
Fig.5.2 Schéma moteur à courant continu
Fig.5.3 Rapport Moteur Bras
Fig.5.4 Le contrôleur muni de FlexCom encerclé
Fig.5.5 Liaison PC – Moteurs
Fig.5.6 Communication avec les actionneurs : Acquisition et Commande
Fig.5.7 Tableau de bord de l'ordinateur de commande
Fig.5.8 Processus d'initialisation
Fig.5.9 Bouton « Check Code »
Fig.5.10 Exécution du programme
Fig.5.11 Procédure pour passer au mode simulation
Fig.5.12 Trajectoire opérationnelle complexe
Fig.5.13.a Comparaison erreur 1 PD et PD MDI pour trajectoire complexe
Fig.5.13.b Comparaison erreur 2 PD et PD MDI pour trajectoire complexe
Fig.5.13.c Comparaison erreur 3 PD et PD MDI pour trajectoire complexe
Fig.5.14 Trajectoire opérationnelle de prise dépose

Fig.5.15 Comparaison erreur PD et PD MDI pour faible dynamique

Fig.5.16 Comparaison erreur PD et PD MDI pour forte dynamique

Fig.5.17 Comparaison Commandes injectées pour trajectoire forte dynamique

Liste des acronymes

α : Position angulaire du bras

BF : Boucle fermée

BO : Boucle ouverte

C_i : Couple au bras i

$d^2\alpha$: accélération angulaire du bras

d^2P : Accélération opérationnelle de l'organe terminal

$d\alpha$: vitesse angulaire du bras

dP : Vitesse opérationnelle de l'organe terminal

ENP : Ecole Nationale Polytechnique

EPFL : Ecole Polytechnique Fédérale de Lausanne

LCP : Laboratoire de Commande des Processus

LSRO : Laboratoire des Systèmes Robotiques

MAD : Modèle des Accélérations Direct

MCD : Modèle Cinématique Direct

MCI : Modèle Cinématique Inverse

MDD : Modèle Dynamique Direct

MDI : Modèle Dynamique Inverse

MGD : Modèle Géométrique Direct

MGI : Modèle Géométrique Inverse

P : Position opérationnelle de l'organe terminal

PD : Proportionnel Dérivée

PID : Proportionnel Intégrateur Dérivée

RE : Retour d'état

RP : Robot parallèle

Introduction Générale

Durant les dernières décennies, la robotique a connu un grand essor, notamment dans les applications industrielles. De plus en plus de tâches sont dévolues aux robots qui sont de plus en plus complexes. Du soudage à la chirurgie en passant par le micro positionnement, chaque tâche requiert un robot avec des caractéristiques particulières.

Dans ce mémoire de projet de fin d'études, nous nous sommes intéressés à une architecture assez particulière : les robots parallèles. Et plus précisément au robot Delta conçu et développé par le professeur Reymond Clavel [8] [10] à l'Ecole Polytechnique Fédérale de Lausanne. Ces structures, d'une complexité grandissante mais surmontable, fascinent des milliers de chercheurs qui sont intéressés par leur potentiel en termes de rapidité et de précision [1] [3] [4] [6] [7]. Depuis leur création dans les années 80, ils ont envahi le secteur industriel et en particulier celui de l'agro-alimentaire

Depuis sa création, plusieurs modèles du robot Delta ont été présentés. Lors de sa conception, R. Clavel a également élaboré son modèle géométrique (directe et inverse) ainsi que son modèle dynamique. Après, des modifications ont été apportées par A. Codourey [9] et d'autres méthodes ont été exposées dans [17] [18] [19] et [25]. Une approche vectorielle a aussi été proposée par N. I. Mokhtari et H. Nadour dans [6]. Cette dernière approche est certes plus simple à manipuler analytiquement mais celle de R. Clavel peut être implémentée plus facilement ce qui justifie notre choix de l'adopter.

C'est dans ce contexte que s'inscrivent les travaux de recherche de ce mémoire visant bien à l'étude des lois de commande ainsi que l'application expérimentale de ces derniers pour la commande du robot Delta ISIR88 du laboratoire LCP.

Ce mémoire, composé de cinq chapitres, est organisé de la façon suivante :

Le premier chapitre s'intéresse à donner des généralités sur les robots et leur classification. Chaque famille de robot présentant ses propres avantages et inconvénients.

S'en suit dans le deuxième titre une présentation générale du robot Delta et ses caractéristiques.

Le troisième chapitre est consacré à l'étude de la modélisation géométrique, cinématique et dynamique du robot Delta ISIR88 disponible à l'Ecole Nationale Polytechnique, en spécifiant son volume de travail et ses singularités. Le modèle dynamique proposé tient compte de la roue de réduction présente dans le bras du robot ISIR88, ce qui le rend plus proche de la réalité que celui présenté par le Professeur R. Clavel [8]. Pour valider les modèles obtenus, nous les comparons avec le bloc SimMechanics du robot Delta ISIR88 proposé par N. I. Mokhtari et H. Nadour dans [6].

Une fois le modèle obtenu, nous nous sommes intéressés à la commande du robot. Pour cela, le quatrième chapitre expose les trois schémas de commande qui ont été testés et comparés en simulation à l'aide du logiciel « Matlab-Simulink » pour en tirer le meilleur.

Le dernier chapitre est consacré à la phase expérimentale, où, à l'aide de l'environnement de commande « FlexCom Development Kit », nous avons pu implémenter le régulateur PD sans et avec l'incorporation de notre modèle dynamique inverse. Pour cela, plusieurs trajectoires de prise-dépose ont été testées et une étude comparative a été élaborée. Cette partie expérimentale constitue notre contribution principale dans le présent travail.

Enfin, une conclusion générale clôture le manuscrit.

CHAPITRE 1

Généralités sur la robotique

Chapitre 1: Généralités sur la robotique

1.1. Introduction

Le terme « Robot » est dérivé du mot tchèque « robota » qui signifie travail forcé ou asservissement [1]. De nos jours, On préfère le définir comme une machine automatique dotée d'une mémoire et d'un programme, capable de se substituer à l'homme pour effectuer certaines tâches, ces tâches peuvent être dangereuses, répétitives, difficiles ou se situant dans des milieux inaccessibles ou hostiles. Les ancêtres des robots sur les automates, on cite l'Homme mécanique jouant d'un instrument à vent et le canard articulé en cuivre mangeant et refoulant sa nourriture après ingestion (simulation du processus de digestion + mouvement des ailes) de Jacques Vaucanson (1737) [2].



Fig.1.1 L'homme jouant l'instrument à vent [2]

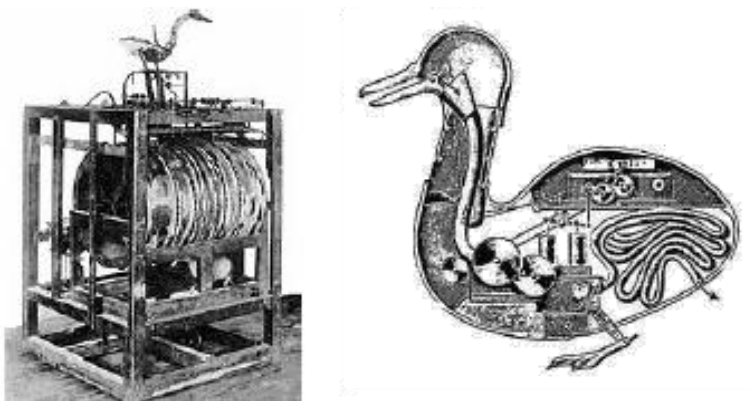


Fig.1.2 Le canard articulé [2]

En 1954, le premier robot industriel a vu le jour, il s'agit de « Unimate », un bras manipulateur utilisé, en 1961, dans les lignes d'assemblage de « General Motors ».



Fig.1.3 Le robot « Unimate » dans l'usine General Motors

Depuis, les robots se sont incrustés dans notre vie et on trouve aujourd'hui les robots industriels, domestiques, médicaux, militaires, ...etc

1.2. Classification

Il existe plusieurs critères pour classer les robots selon leurs architectures, fonctionnalités, méthodologies de modélisation ou de commande.

Selon la fonctionnalité, on trouve deux grandes familles de robots :

- Les robots mobiles : C'est des robots qui se déplacent, dotés de fonctions de perception, de décision et d'action. Par exemple les robots d'assistance domestiques, les voitures autonomes, ...



Fig.1.4.a Robot B21



Fig.1.4.b Pioneer P3-DX

- Les robots manipulateurs : Ce sont des robots avec des fonctions similaires à un bras humain. Les liens de ce type de robots sont reliés par des axes permettant, soit du mouvement de rotation ou de translation [2]. Ils peuvent être autonomes ou contrôlés manuellement et peuvent être utilisés pour effectuer une variété de tâches avec une grande précision.

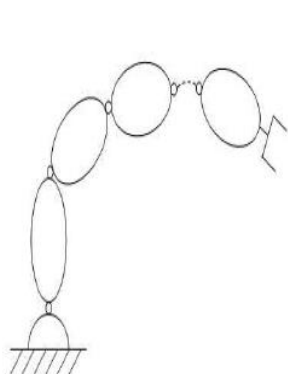


Fig.1.5 Robot manipulateur « SCARA » [2]

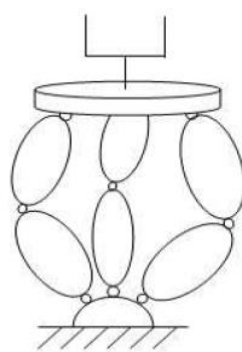


Fig.1.6 Robot manipulateur dans un atelier de soudage [2]

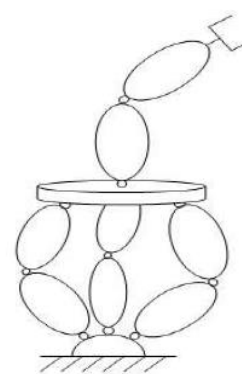
Ces robots manipulateurs sont classés, selon leur architecture, en 2 grandes familles : Les robots sériels (ou série) et les robots parallèles. À noter aussi l'existence d'une architecture appelée « hybride » qui est le mélange des deux.



(a) Architecture sérielle



(b) Architecture parallèle



(c) Architecture hybride

Fig.1.7 Différentes architectures de robots [3]

1.2.1. Robots sériels :

Un robot sériel est structuré sous forme d'une chaîne ouverte de succession de corps reliés entre eux par une articulation d'un seul degré de liberté (Rotatoire ou prismatique). Chaque articulation est mue par un actionneur.

Avantages :

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

Inconvénients :

- Une inertie élevée due aux masses réparties sur toute la chaîne cinématique (actionneurs et organes de transmission) qui peut engendrer des erreurs de positionnement. Chaque articulation a une erreur de positionnement, et en démarrant de la base jusqu'à l'organe terminal, ces erreurs s'additionnent.
- Un manque de rigidité. En effet, le fait que les éléments soient disposés en série implique que chaque corps doit supporter sa charge ainsi que celle du corps suivant.

Les inconvénients cités ci-dessus ont poussé les chercheurs à introduire un nouveau type de structure appelée la structure parallèle. Elle offre des avantages pratiques, mais elle a tout de même ses propres inconvénients [4].

1.2.2. Robots parallèles :

Un robot parallèle est constitué d'une plateforme fixe (base) et d'une autre mobile (nacelle ou effecteur) liées entre elles par plusieurs chaînes cinématiques en parallèle.

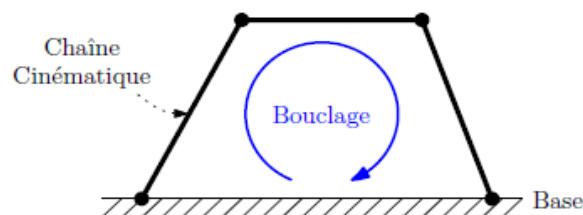


Fig.1.8 Exemple d'une structure parallèle [3]

Avantages :

- Capacité de charge plus importante que celle d'un robot série avec un poids de robot équivalent.
- L'accélération de l'organe terminale peut être très élevée (Peut atteindre 50g pour le robot delta par exemple)
- Une bonne rigidité mécanique confère aux structures parallèles une bonne précision.
- Le fait qu'on peut mettre les actionneurs et les capteurs sur la base fixe offre une grande flexibilité technologique. Par exemple, si besoin, on peut utiliser un moteur puissant même si sa masse est importante. Même chose pour les capteurs et les accessoires.
- La symétrie des robots parallèles facilite le processus de fabrication (Le robot delta par exemple est constitué de trois chaînes cinématiques identiques en plus de la base solide et de la nacelle) [4].

“Parallel-kinematic technology can be applied in any domain where you need multiple axes and outstanding performances.”

LSRO-EPFL

Inconvénients :

- L'étude de telles structures peut s'avérer très compliquée (Modélisation et commande).
- La fermeture de la chaîne cinématique limite l'espace de travail (Les points de l'espace atteignables par la nacelle). Pour y remédier, il faut faire un bon dimensionnement du robot en fonction des caractéristiques souhaitées.
- La présence de singularités peut conduire à une perte de commandabilité dans certains points de l'espace.

Quelques exemples des robots parallèles [5]

Le delta

Le robot delta est l'un des robots parallèles les plus vendus au monde. Créé par le professeur R. Clavel en 1989 au sein du Laboratoire de Système Robotiques, il existe actuellement plus de 4000 copies à travers le monde. En industrie, il est beaucoup utilisé dans les processus Prise-Dépose. Une étude détaillée de ce type de robot est présentée dans la suite de ce mémoire.



Fig.1.9 Robot delta de l'EPFL [5]

Le MinAngle

Le MinAngle est un robot parallèle de 3 degré de liberté, constitué de trios chaînes cinématique identiques flexibles. L'effecteur peut se déplacer selon les axes X, Y et Z.

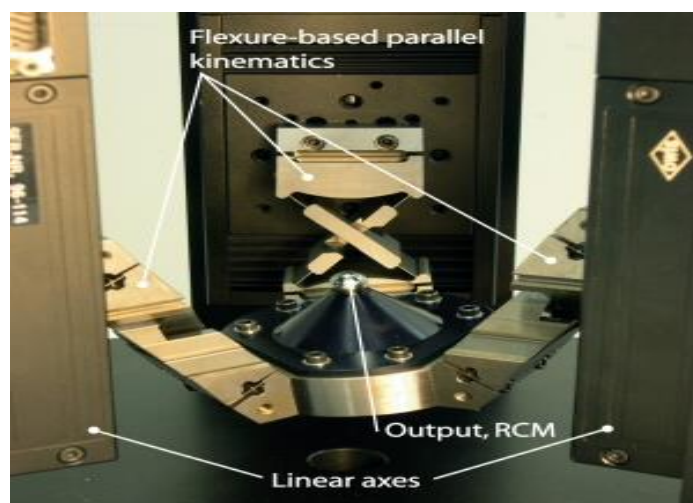


Fig.1.10 Robot MinAngle [5]

1.3. Modélisation et commande des robots

1.3.1. Structure de commande

L'étude de ce type de système passe par deux phases majeures : La modélisation et la commande. Une fois que le modèle du robot est obtenu (La méthodologie est présentée dans le paragraphe qui suit), On passe à l'étape de la commande, c'est-à-dire élaborer un programme dans le but de s'assurer que l'organe terminal du robot atteigne les positions désirées à tout instant, on parle ici de l'asservissement en trajectoire.

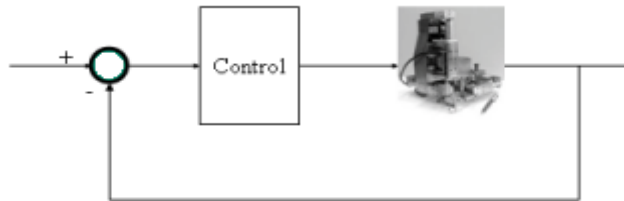


Fig.1.11 Synoptique de commande [5]

Pour une référence donnée, à un instant « t », le robot reçoit en entrée les tensions d'alimentation de ses actionneurs, ces tensions sont appelées : Signaux de commande. Selon la valeur de ces signaux ainsi que la géométrie et la cinématique du robot, la nacelle se positionne à une position (X, Y, Z) en ayant une vitesse (X', Y', Z') et une accélération (X'', Y'', Z'') . Ces valeurs, obtenues à l'aide de capteurs, sont comparées aux valeurs désirées (appelées références). On calcule ensuite l'erreur et selon la valeur de cette dernière, le contrôleur fournit les tensions à appliquer à l'instant « $t+1$ » et ainsi de suite. Ce principe de fonctionnement est appelée « Commande en boucle fermée ».

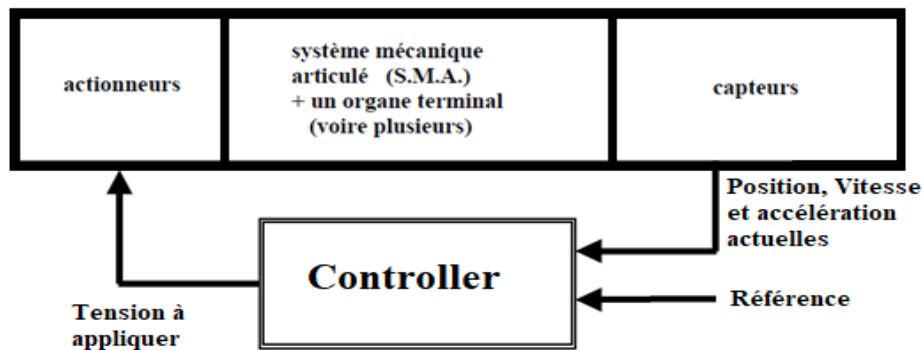


Fig.1.12 Système de commande en robotique [5]

Dans la pratique, La génération des signaux de commande ainsi que l'acquisition des données des capteurs sont assurée par un ordinateur. A noter aussi l'interaction avec l'environnement extérieur qui est aussi prise en considération. Cette interaction peut être la présence d'obstacles, perturbations, ...etc.

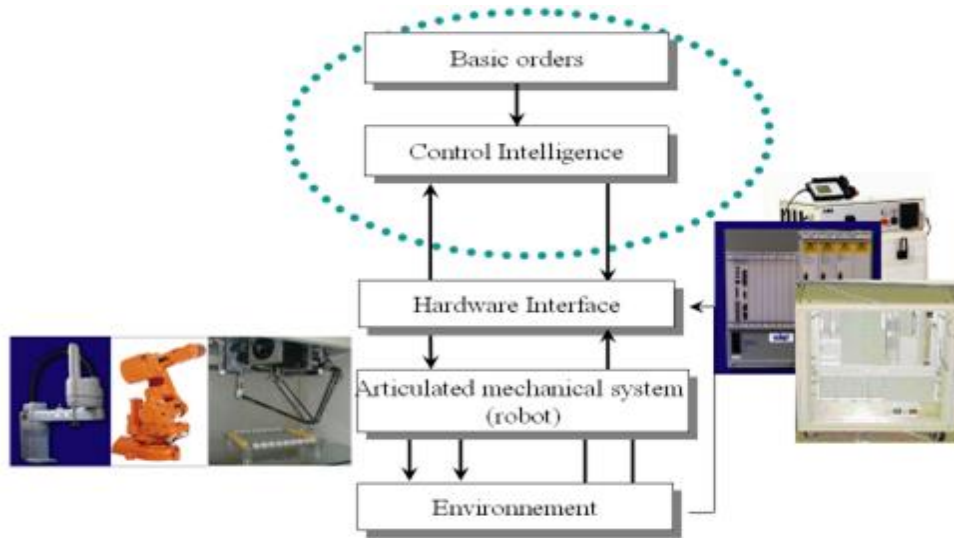


Fig.1.13 Etapes exécutées par le système de commande [5]

1.3.2. Modélisation

Modélisation géométrique

Le modèle géométrique du robot permet d'exprimer les coordonnées opérationnelles (x ; y ; z) de sa nacelle en fonction des angles articulaires (θ_1 ; θ_2 ; θ_3) ou l'inverse.

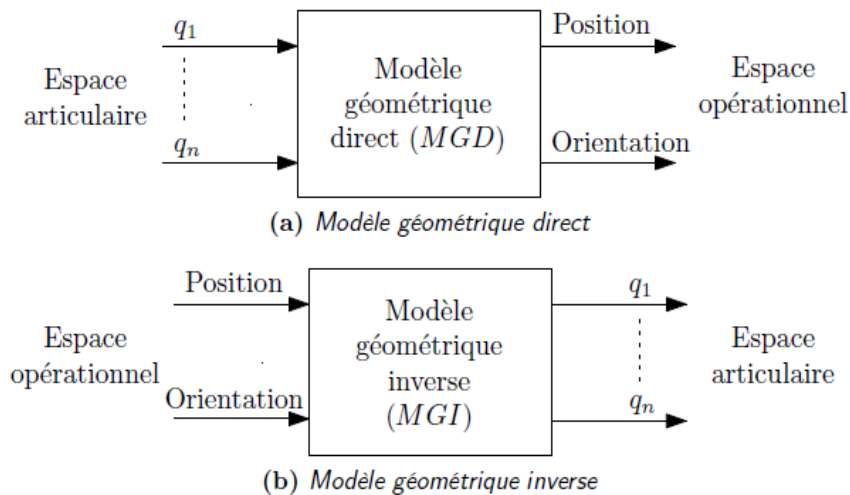


Fig.1.14 Modèle géométrique [3]

Pour le calcul du modèle géométrique d'une structure série, on utilise généralement des conventions simplificatrices de Khalil & Al. Quant aux robots parallèles, l'étude n'est pas systématique et il faut ressortir des relations géométriques selon la structure particulière du robot étudié.

Modélisation cinématique

La modélisation cinématique d'un robot manipulateur en général et du robot Delta en particulier, permet d'établir les relations entre les vitesses articulaires (généralisées) et les vitesses opérationnelles de l'organe terminal.

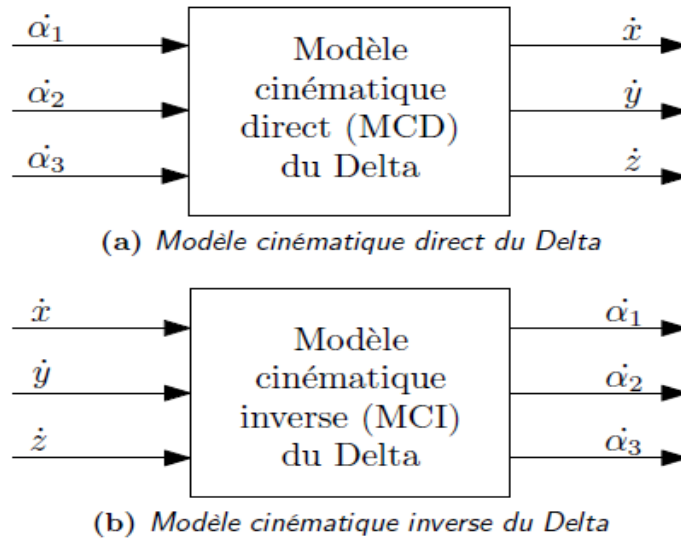


Fig.1.15 Modèle cinématique [3]

Ce modèle fait apparaître la matrice Jacobienne « J » qui est primordiale dans l'étude des robots (Par exemple pour le calcul des singularités)

Modélisation dynamique

L'utilisation du modèle dynamique est primordiale dans les étapes de conception et de commande du robot. Il s'agit d'exprimer la relation entre l'effort des actionneurs (Forces, Couples), les accélérations et les trajectoires du mouvement résultant de l'organe terminal.

Le modèle dynamique d'un robot exprime cette relation dans un sens ou dans l'autre.

C'est pour cela que nous avons deux cas : Modèle dynamique direct et Modèle dynamique inverse [3].

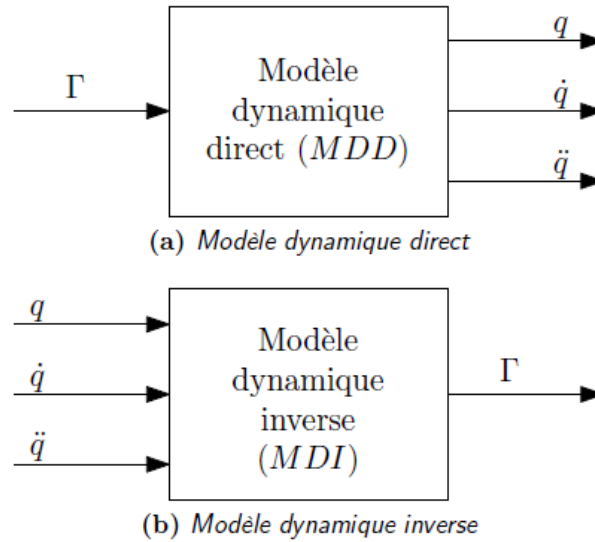


Fig.1.16 Modèle Dynamique [3]

La formule générale est du type:

$$\Gamma = B(q) \cdot \ddot{q} + G(q) + C(q, \dot{q}) + F(q, \dot{q}) + K(q)$$

Avec :

Γ Le vecteur de couple généralisé

$B(q)$ La matrice d'inertie

$G(q)$ Le vecteur gravité

$C(q, \dot{q})$ Les termes centrifuge et coriolos

$F(q, \dot{q})$ Les termes de frottements

$K(q)$ Les termes de rigidité

Les travaux sur la modélisation dynamique des robots parallèles ont commencé avec l'analyse dynamique de la plate forme de Stewart en traitant principalement soit les problèmes d'oscillation ou les problèmes de la dynamique inverse dans des formalismes très simples. Plus tard, d'autres travaux ont présenté des analyses plus élaborées pour résoudre le problème de la modélisation dynamique des RP, en utilisant des formalismes mécaniques différents : Le formalisme Lagrange-Euler, le principe des travaux virtuels et les équations de Newton-Euler. [10]. À noter que ce modèle est très important car on en ressort la représentation d'états du système. Une représentation sur laquelle se basent plusieurs types de commande.

1.3.3. Génération de trajectoire

La tâche de déplacement d'un robot est spécifiée en définissant un chemin que le robot doit suivre.

Un chemin est une séquence de points définis soit dans l'espace des tâches (opérationnel) (afin de situer l'organe terminal), soit dans l'espace des configurations (articulaire) du robot (afin d'indiquer les valeurs des paramètres de liaison).

Le problème de la génération de mouvement est de calculer les séquences souhaitées (consigne) de variables articulaires ou de variables liées à l'organe terminal qui assurent le passage du robot par le chemin désiré.

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

- les mouvements entre 2 points avec des mouvements libres entre les points,
- les mouvements entre 2 points via une séquence de points intermédiaires désirés, spécifiés notamment pour éviter les obstacles ; la trajectoire est libre entre les points intermédiaires,
- les mouvements entre 2 points, la trajectoire étant contrainte entre les points (trajectoire rectiligne par exemple),
- les mouvements entre 2 points via des points intermédiaires, la trajectoire étant contrainte entre les points intermédiaires.

Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace des configurations : elle se traduit par une séquence de positions articulaires constituant les consignes des asservissements.

Dans les deux derniers cas, la trajectoire étant fixée à tout instant dans l'espace opérationnel, il est préférable de raisonner dans cet espace.

La loi de commande engendrée doit ensuite être transformée en consignes articulaires par le changeur de coordonnées [4].

1.3.4. Principales méthodes de commande

En robotique, on a besoin de commander la position de chaque articulation selon une trajectoire désirée appelée référence. Le but étant de s'assurer qu'à chaque instant, la position mesurée (donnée par les capteurs) est identique à la position indiquée par la trajectoire de référence.

Comme expliqué précédemment, le travail d'un automaticien consiste à synthétiser la loi de commande, c'est-à-dire les tensions à injecter à chaque instant pour suivre la trajectoire et ainsi atteindre l'objectif de commande. Nous citons ici les principales méthodes de commande utilisées en robotique (Et plus précisément pour les robots parallèles).

Vu la similitude de la structure des équations cinématiques et dynamiques entre les robots sériels et parallèles, la plupart des lois de commandes classiques développées pour les robots à chaînes sériels s'appliquent également pour les mécanismes parallèles. Toutefois, les lois commandes existants peuvent être groupées en « Commande sans modèle dynamique » et « Commande avec modèle dynamique ».

Dans le domaine industriel, la commande la plus utilisée pour les robots est la commande classique type PID (ou PD) décentralisée c'est-à-dire le modèle du robot est considéré comme linéaire et découplé (Voir figure suivante). L'avantage principal de cette méthode est l'implémentation facile et le faible coût en calcul. Les paramètres du contrôleur sont réglés expérimentalement en minimisant au maximum l'erreur de suivi de trajectoire. L'inconvénient majeur de cette méthode est la présence de pics dans la réponse temporelle du robot et la dégradation des performances dans les mouvements rapides [10].

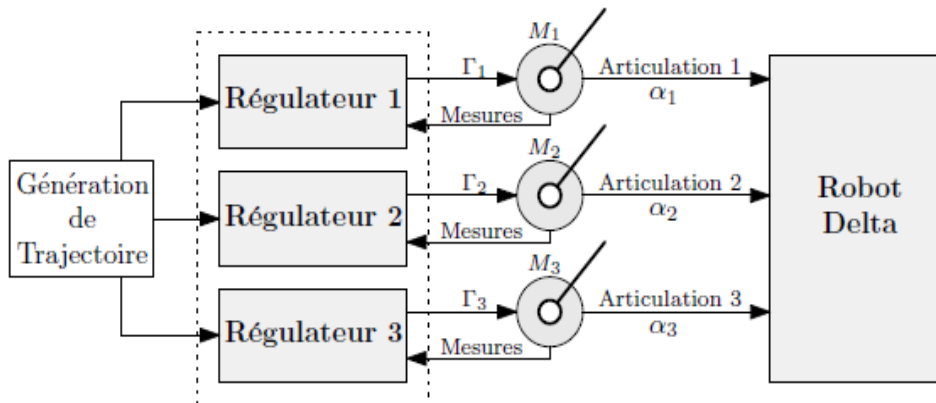


Fig.1.17 Commande PID décentralisée [3]

Afin d'éviter ce problème, l'incorporation du modèle dynamique inverse dans les schémas de commande du robot est devenue nécessaire.

Ce dernier a pour rôle de calculer le couple à priori nécessaire pour chaque articulation comme le montre la figure ci-dessous.

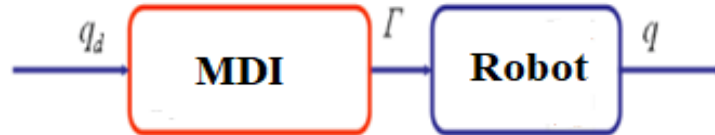


Fig.1.18 Schéma de commande en boucle ouverte avec MDI

Afin d'assurer le suivi parfait de la trajectoire désirée tout en respectant les conditions de robustesse, on ferme la boucle de commande en insérant un bloc de régulation. Le schéma se présente ainsi sous la forme :

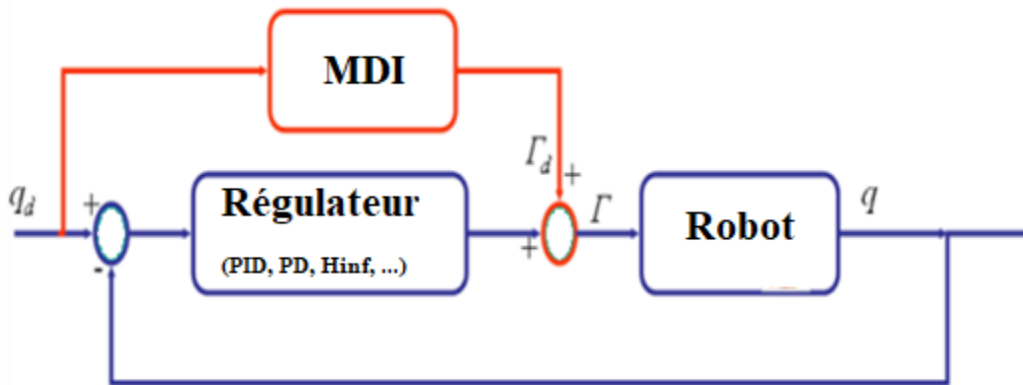


Fig.1.20 Commande avec MDI en boucle fermée

Plusieurs méthodes sont utilisées pour la synthèse du régulateur, on cite :

La commande classique PID

La commande par bouclage linéarisant

La commande adaptative

La commande par modes glissants (Sliding Mode)

La commande par logique floue

La commande robuste H_∞

1.4. Conclusion

Dans ce chapitre, nous avons présenté quelques généralités sur les robots : définitions, historique et classification.

Ceci nous a permis de tirer les avantages et les inconvénients des structures sérielles afin de les comparer aux structures parallèles. Bien évidemment, toutes les caractéristiques générales des robots parallèles s'appliquent au robot Delta ISIR88. Une étude plus détaillée des caractéristiques de ce dernier sera faite dans le chapitre suivant.

Enfin, nous avons présenté les différentes méthodes de modélisation et commande utilisées en robotique.

CHAPITRE 2

Le Robot Delta

Chapitre 2: Le robot delta

2.1. Introduction

L'objet de ce mémoire étant la modélisation et la commande du « Delta », nous entamons notre travail avec une présentation générale de ce robot créé en 1985 par le Professeur R. Clavel au sein du Laboratoire LSRO de l'Ecole Polytechnique Fédérale de Lausanne.

Le delta est un robot parallèle à trois degrés de liberté de translation (X, Y, Z), constitué de trois chaînes cinématiques identiques, une base et une nacelle. Il est caractérisé par une rapidité exceptionnelle (plus de 200 opérations par minute) tout en ayant une grande précision.

Il est mu par trois actionneurs qui peuvent être rotatoires (Cas du robot étudié dans cette thèse) ou rotationnels.

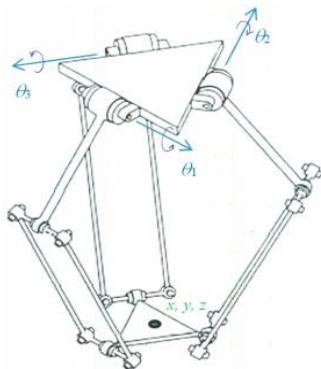


Fig.2.1.a Architecture du Delta



Fig.2.1.b Robot delta d'ABB

2.2. Historique

1985 Reymond Clavel, ingénieur doctorant à l'École polytechnique fédérale de Lausanne (EPFL), Laboratoire de système robotiques (LSRO), l'imagine et dépose un brevet.

1987 il entre en application avec une licence vendue aux frères Demaurex qui entreprennent sa commercialisation pour l'industrie de mise en boîte (conditionnement des boîtes de chocolat).

1988 Demaurex commence le développement du robot Delta en partenariat avec l'EPFL. Ils parviennent en **1992** de vendre la première application industrielle du robot delta à un fabricant de produits de boulangerie.

Pour plusieurs années, le robot sera commercialisé par Bosch Packaging Technology SA (Ex. Demarex SA) sous les noms de Presto, Paloma, Delfi et Astor, par ABB sous le nom de Flexpicker, ainsi que SIG-Bosch sous les noms de SIGMonoPacker et XR31 et reste sans contredit le robot parallèle le plus populaire.

En **2007**, le brevet monte dans le domaine public [4].

2.3. Applications

Initialement conçu pour manipuler des objets minuscules comme du chocolat, le robot Delta est, de nos jours, utilisés dans des applications diverses :

- L'industrie alimentaire et pharmaceutique (Fig.2.2.a)
- Application médicale et télé-chirurgie (Fig.2.2.b)
- Les systèmes de micro positionnement et Impression 3D (Fig.2.2.c)



Fig.2.2.a Robot Delta en chirurgie



Fig.2.2.b Robot Delta en industrie alimentaire (à gauche) et pharmaceutique (à droite)

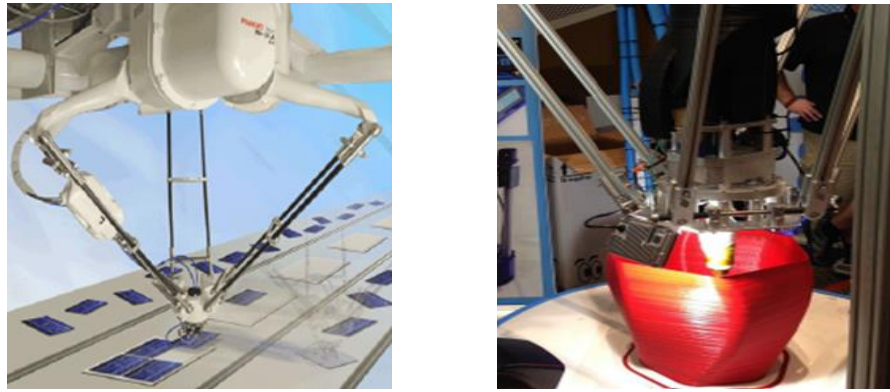


Fig.2.2.c Robot Delta micro positionnement et impression 3D

2.4. Le robot delta ISIR88

2.4.1. Description

Le robot disponible au Laboratoire de Commande des Processus (LCP) de l'École Nationale Polytechnique et faisant l'objet de cette étude est un robot delta à actionneurs rotatoires, de trois degrés de liberté. Le contrôle du robot est assuré par un ordinateur muni du système de commande FlexCom (voir chapitre suivant pour plus de détails) comme le montre la figure 2.3 :



Fig.2.3 Robot Delta du LCP – ENP ISIR88

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

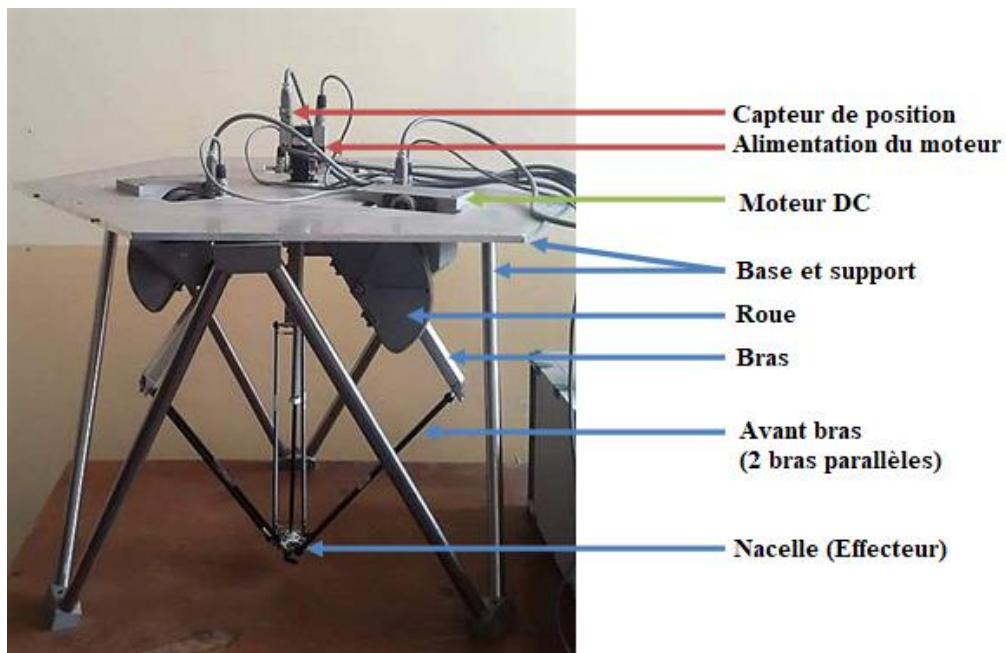


Fig.2.4 Constituants du robot delta du LCP

Les trois actionneurs ainsi que la structure de chaîne fermée permettent un déplacement linéaire de la nacelle qui est présentée sur la figure suivante.

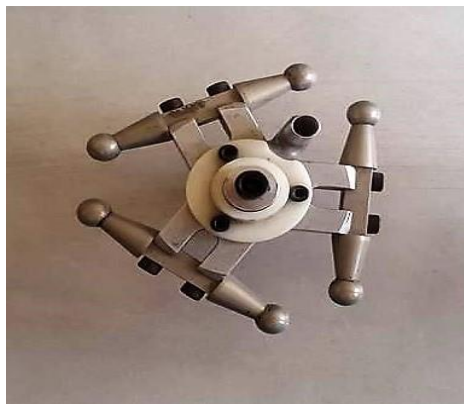


Fig.2.5 Vue de haut de la nacelle

Chaque chaîne cinématique est constituée d'un bras joint à deux barres parallèles (Voir Fig.4). Le mouvement de chaque bras est assuré par un moteur à courant continu placé sur la base fixe et dont les caractéristiques sont détaillées dans le paragraphe suivant.



Fig.2.6 Vue de haut des barres parallèles

Les articulations aux extrémités des barres parallèles sont de type rotule, chaque barre peut donc tourner autour de son axe longitudinal; cette rotation ne perturbe pas le comportement de cette structure articulée formant le parallélogramme de l'espace; La transmission entre un moteur est un bras n'est pas direct, en effet elle est assurée par un système de réduction constitué par une poulie, un pignon et une roue afin d'obtenir un rapport de réduction égal à 12:1. (Couple au niveau du bras = 12 x Couple au niveau du moteur). Donc si on note α la position angulaire du bras et C le couple rapporté au bas, et θ la position angulaire du moteur et Γ le couple fourni par le moteur, nous aurons les relations suivantes :

$$\begin{cases} C = \Gamma \times 12 \\ \alpha = \frac{\theta}{12} \end{cases}$$

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

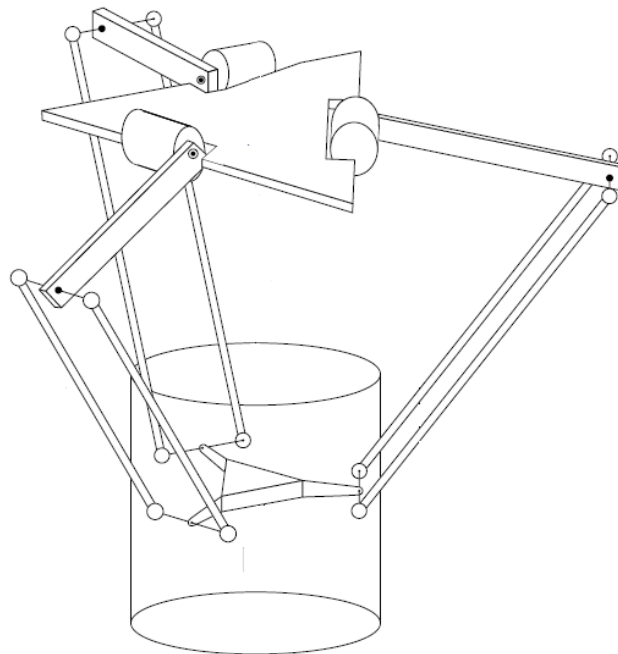


Fig.2.7 Espace de travail du Delta [3]

A noter qu'un quatrième degré de liberté peut être rajouté. Il s'agit d'un arbre télescopique permettant la rotation de l'effecteur autour de lui-même.



Fig.2.8 L'arbre télescopique assurant le quatrième degré de liberté

Vu que ce quatrième degré de liberté n'influe pas beaucoup sur la dynamique du robot étant assuré par un autre moteur à part, nous avons décidé, pour des raisons de simplification, de ne pas prendre la rotation de la nacelle en considération (i.e. enlever l'arbre télescopique) durant notre étude.

2.4.2. Caractéristiques

Caractéristiques de la structure mécanique du robot :

	Eléments	Symbole	Valeur
Longueurs	Longueur des bras	l_1	0.2 m
	Longueur des barres parallèles	l_2	0.3 m
	Rayon de la base	r_1	0.15 m
	Rayon de la nacelle	r_2	0.024 m
Masses	Masse du bras	M_1	0.08366 Kg
	Masse des barres parallèles	M_2	2 x 0.01438 Kg
	Masse de la roue	M_3	0.1863 Kg
	Masse de la nacelle	M_4	0.0416 Kg

Tab.2.1 Caractéristiques mécaniques du robot ISIR88

Caractéristiques des actionneurs (Moteurs) :

Les actionneurs sont des moteurs électriques à courant continu, ayant les caractéristiques suivantes :



Fig.2.9 Actionneur du Delta

Peak Rated Torque	N-m	1.01
Rated Torque	N-m	0.49
Torque Constant	N-m/A \pm 10%	0.119
Armature Winding Resistance (at 25°C)	$\Omega \pm 10\%$	1.35
Armature Inductance	mH	1.2
Peak Current	A	8.7
Voltage Constant	V/1000rpm \pm 10%	12.5
Viscous Damping Coefficient	N-m/1000rpm	0.00882
Friction Torque	N-m	0.333
Inertia	kg·m ² $\times 10^{-4}$	0.372
Mechanical Time Constant	ms	3.5
Electrical Time Constant	ms	0.89
Power Rate	kW/s	6.44
Torque Inertia Ratio	rad/s ²	13200
Thermal Resistance	deg C/watt	2.45
Thermal Time Constant	minutes	13
Max Allowable Armature Temperature	°C	155
Rated Speed	r/min	1000
Max Safe Operating Speed	r/min	2000
Max No Load Speed	r/min	6000

Tab.2.2 Caractéristiques des actionneurs du ISIR88 (Datasheet)

2.5. Conclusion

Ce chapitre débute par une présentation générale du robot Delta en exposant son historique et ses domaines d'application. Il représente la dernière génération des robots parallèles et est caractérisé par sa légèreté, sa précision et ses accélérations très élevées.

Ensuite, une description détaillée du robot Delta ISIR88 à été présentée en mettant en avant ses constituants et ses caractéristiques mécaniques et électriques. En fait, il s'agit d'un modèle particulier du robot Delta à entraînement rotatif contenant une roue de réduction entre l'actionneur et le bras.

Dans le chapitre suivant, nous allons nous intéresser à la modélisation géométriques, cinématique et dynamique du robot Delta ISIR88, étape indispensable avant de passer à sa commande.

CHAPITRE 3

Modélisation du robot Delta ISIR88

Chapitre 3: Modélisation du robot delta ISIR88

3.1. Introduction

Dans ce chapitre, nous allons faire une présentation et une description du robot Delta, à 3 ddl et à entraînement direct, utilisé dans ce travail de thèse pour implémenter les différentes lois de commande proposées. Ce dernier est validé par le modèle non linéaire du robot Delta représenté sous un bloc SimMechanics [6].

3.2. Modélisation géométrique

Pour le cas du robot Delta à trois degrés de liberté à actionnement rotatif, le modèle géométrique permet de lier les coordonnées cartésiennes de la nacelle X, Y et Z avec les angles des trois articulations rotoïdes du robot α_i .

La structure assez particulière du robot delta nous permet de travailler sur une chaîne du robot au lieu de trois.

Afin de simplifier l'étude, nous supposons que la nacelle reste tout le temps parallèle à la base fixe et qu'elle n'effectue que des mouvements de translations. Nous considérons aussi que celle-ci est remplacée par un point matériel « P ».

Enfin, les barres parallèles peuvent être considérées comme un seul bras situé au milieu de ces dernières.

3.2.1. Indices et paramètres géométriques

Comme indiquée sur la figure ci-dessous, le repère de base « R » est choisi au milieu du cercle passant par les trois articulations. L'axe Z est orienté vers le haut et X vers l'articulation i. L'indice i ($i=1...3$) est utilisé pour identifier les trois chaînes (resp. bras, moteur).

Chaque chaîne est décalée à celle qui la suit de 120° , cependant, on choisit les repères relatifs à chaque chaîne en tournant le repère « R » autour de son axe OZ de $\varphi_1=0^\circ$, $\varphi_2=120^\circ$ et $\varphi_3=240^\circ$ [11] [12] [15].

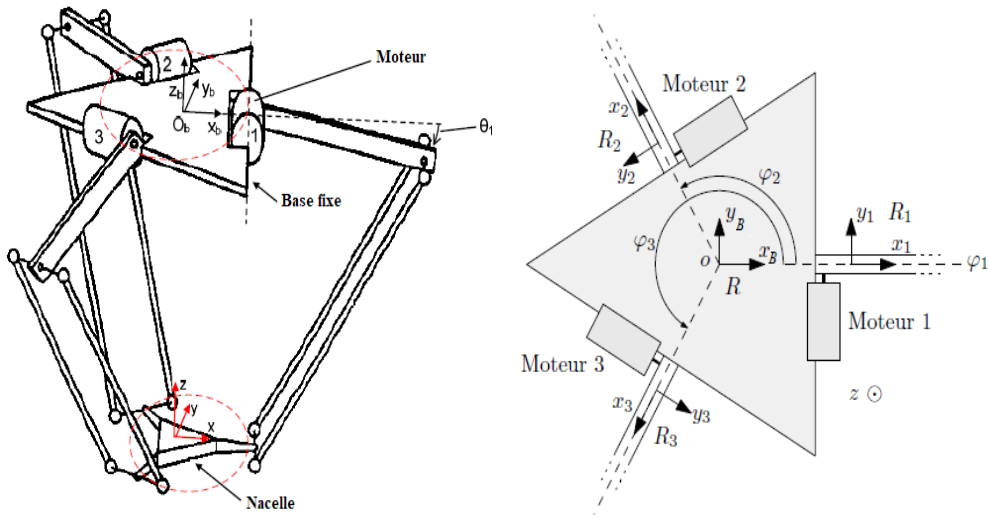


Fig.3.1 Schématisation et placement des repères [1] [3]

Les matrices de rotations R_iT sont données par :

$${}^R_iT = \begin{pmatrix} \cos\varphi_i & -\sin\varphi_i & 0 \\ \sin\varphi_i & \cos\varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

On note par β_i l'angle entre l'avant bras (barres parallèles) et le bras adjacent. Ci représente l'extrémité commune entre le bras et l'avant bras. Pour finir, on introduit un dernier paramètre γ_i qui représente l'angle entre le plan vertical de la chaîne i noté Π_i et les barres parallèles comme le montre la figure suivante :

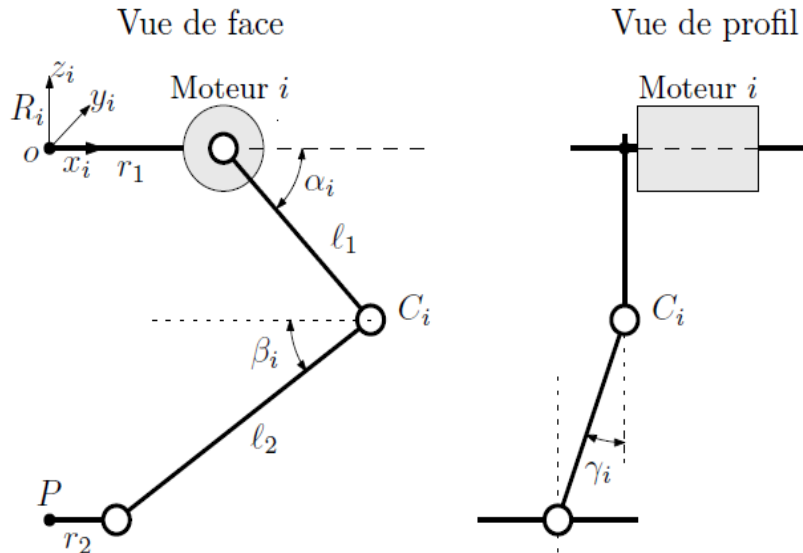


Fig.3.2 Vues de face et de profil d'une chaîne « i » du robot delta [3].

Une dernière simplification est de faire une translation horizontale de toute la chaîne de « r_2 » dans la direction du point O. Ceci donne :

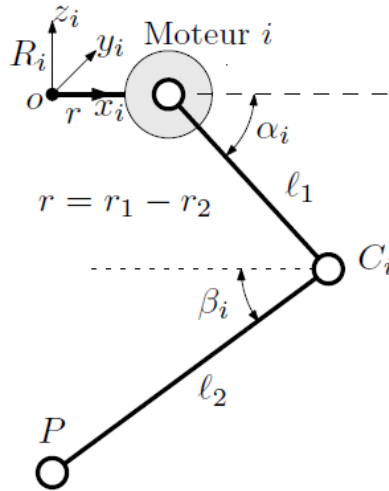


Fig.3.3 Schéma d'une chaîne « i » après simplification [3].

A noter l'apparition d'un nouveau paramètre « r » égal à la différence entre le rayon de la nacelle et celui de la base fixe (voir figure).

3.2.2. Modèle géométrique direct « MGD »

Depuis la création du robot Delta par le professeur R. Clavel, plusieurs méthodes pour le calcul du modèle géométrique ont été développées, nous citons le modèle original donné par R. Clavel [12] lui-même et le modèle basé sur un développement vectoriel expliqué dans [8].



Fig.3.4 Modèle géométrique direct du Delta [7]

Comme le montre la figure 5.3, les coordonnées du point Ci dans le repère Ri est donné par :

$$\overrightarrow{OC_i/R_i} = \begin{pmatrix} r + l_1 \cos(\alpha_i) \\ 0 \\ -l_1 \sin(\alpha_i) \end{pmatrix} \quad (3.2)$$

Pour passer au repère de base, nous faisons une rotation de φ_i autour de l'axe Z qui se traduit mathématiquement par une multiplication matricielle :

$$\overrightarrow{OC_i} = {}^R T_i \times \overrightarrow{OC_{i/R_i}} \quad (3.3)$$

$$\overrightarrow{OC_i} = \begin{pmatrix} \cos\varphi_i & -\sin\varphi_i & 0 \\ \sin\varphi_i & \cos\varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} r + l1 \cos(\alpha_i) \\ 0 \\ -l1 \sin(\alpha_i) \end{pmatrix} \quad (3.4)$$

Nous obtenons enfin, les coordonnées du point Ci dans le repère de base « R » :

$$\overrightarrow{OC_i} = \begin{pmatrix} (r + l1 \cos(\alpha_i)) \cos\varphi_i \\ (r + l1 \cos(\alpha_i)) \sin\varphi_i \\ -l1 \sin(\alpha_i) \end{pmatrix} \quad (3.5)$$

La structure particulière du robot delta fait qu'à chaque instant, et quel que soit la configuration articulaire, les trois points C1, C2 et C3 appartiennent à une même sphère centrée en P et ayant comme rayon l2. Elle a pour équation :

$$(xi - X)^2 + (yi - Y)^2 + (zi - Z)^2 = l2^2 \quad (3.6)$$

Avec (X Y Z) les coordonnées du point P (La nacelle) et (xi yi zi) les coordonnées des points Ci.

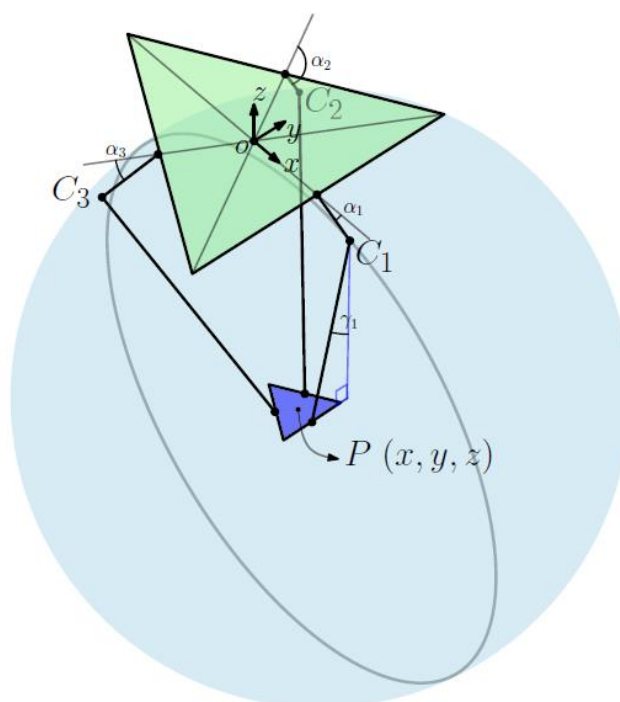


Fig.3.5 La sphère contenant les points Ci du Delta [3]

Nous remplaçons les coordonnées des points C_i par leurs formules (3.5) dans l'équation de la sphère (3.6), nous obtenons le système d'équations suivant (3.7) :

$$\begin{cases} \left((r + l1 \cos(\alpha_1)) \cos\varphi_1 - X \right)^2 + \left((r + l1 \cos(\alpha_1)) \sin\varphi_1 - Y \right)^2 + (-l1 \sin(\alpha_1) - Z)^2 = l2^2 \\ \left((r + l1 \cos(\alpha_2)) \cos\varphi_2 - X \right)^2 + \left((r + l1 \cos(\alpha_2)) \sin\varphi_2 - Y \right)^2 + (-l1 \sin(\alpha_2) - Z)^2 = l2^2 \\ \left((r + l1 \cos(\alpha_3)) \cos\varphi_3 - X \right)^2 + \left((r + l1 \cos(\alpha_3)) \sin\varphi_3 - Y \right)^2 + (-l1 \sin(\alpha_3) - Z)^2 = l2^2 \end{cases}$$

C'est un système de 3 équations non linéaires à trois inconnus X , Y et Z . Sa résolution peut être numérique en utilisant, par exemple, la fonction « *fsolve* » de Matlab, ou analytiquement comme expliqué dans [12] et donné ci-dessous. En fait, en posant :

$$\begin{aligned} D_i &= -l2^2 + l1^2 + r^2 + 2 \times r \times l1 \times \cos(\alpha_i) \\ E_i &= 2 \times (r + l1 \times \cos(\alpha_i)) \times \cos(\varphi_i) \\ F_i &= 2 \times (r + l1 \times \cos(\alpha_i)) \times \sin(\varphi_i) \\ G_i &= -2 \times l1 \times \sin(\alpha_i) \end{aligned}$$

Et en notant :

$$\begin{aligned} H1 &= E1G2 - E1G3 - E2G1 + E2G3 + E3G1 - E3G2 \\ H2 &= -E1F2 + E1F3 + E2F1 - E2F3 - E3F1 + E3F2 \\ H3 &= -E1D2 + E1D3 + E2D1 - E2D3 - E3D1 + E3D2 \\ H4 &= F1D2 - F1D3 - F2D1 + F2D3 + F3D1 - F3D2 \\ H5 &= -F1G2 + F1G3 + F2G1 - F2G3 - F3G1 + F3G2 \end{aligned}$$

Alors, on obtient les expressions des coordonnées (X Y Z) qu'on notera (3.8) :

$$\begin{aligned} X &= Z \frac{H5}{H2} + \frac{H4}{H2} \\ Y &= Z \frac{H1}{H2} + \frac{H3}{H2} \\ Z &= \frac{-M \pm \sqrt{M^2 - 4LN}}{2L} \end{aligned}$$

Avec :

$$\begin{aligned} L &= \frac{H5^2 + H1^2}{H2^2} + 1 \\ M &= 2 \frac{H5H4 + H1H3}{H2^2} - \frac{H5E_1 + H1F_1}{H2} - G_1 \\ N &= \frac{H4^2 + H3^2}{H2^2} - \frac{H4E_1 + H3F_1}{H2} + D_1 \end{aligned}$$

Remarque :

L'expression de Z donne toujours deux résultats. Dans notre cas, et selon le placement du repère de base « R », nous prenons à chaque fois la valeur de Z négative.

Afin que nous puissions valider ce modèle géométrique, nous le comparons au bloc SimMechanics du robot delta obtenu par la conception 3D sous SolidWorks. Ce dernier a été conçu et développé par Nour Islam MOKHTARI et Housseine NADOUR dans [6]. Ce bloc SimMechanics représente l'architecture réelle du robot delta, prenant compte de tous les paramètres physiques et les contraintes structurelles. Nous injectons aux deux blocs (SimMechanics et MGD) les mêmes entrées. Ces entrées sont des angles α_i assurant une trajectoire circulaire. L'erreur entre le MGD et le bloc SimMechanics est représentée sur Fig.3.6.

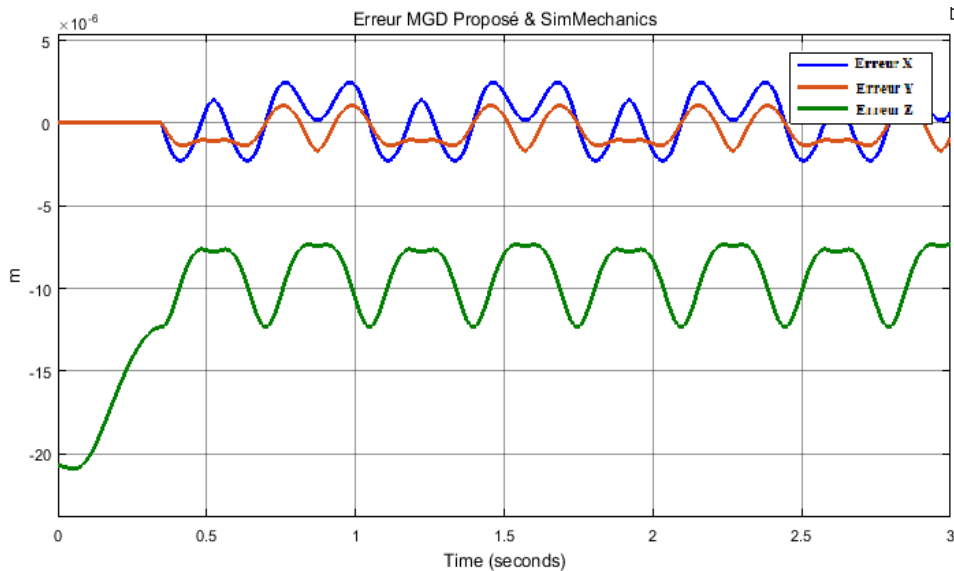


Fig.3.6 Erreur entre MGD et SimMechanics

Toutes les erreurs trouvées sont très petites, ce qui signifie que notre Modèle Géométrique Direct se comporte comme le bloc SimMechanics d'où la validation du MGD.

3.2.3. Modèle géométrique inverse « MGI »

Ce modèle donne la relation inverse du MGD, c'est-à-dire le passage de l'espace de position de la nacelle (x, y, z) vers l'espace articulaire ($\alpha_1, \alpha_2, \alpha_3$) et donc à ($\theta_1, \theta_2, \theta_3$) en prenant compte le rapport de réduction 1:12. Ce modèle est très utile car il permet d'avoir, à chaque instant « t », la configuration articulaire nécessaire pour obtenir la trajectoire désirée.



Fig.3.7 Modèle géométrique indirect du Delta [7]

Il s'agit tout simplement de résoudre le système d'équations (3.7) mais en prenant cette fois les paramètres X , Y et Z comme données et les α_i comme inconnus. Nous pouvons résoudre le système non linéaire numériquement ou analytiquement. L'une des méthodes analytiques de résolution est celle proposée dans [8] [9] [10]. En développant l'équation de la sphère (3.6) et en remplaçant x_i , y_i et z_i par leurs valeurs, on obtient :

$$\left((r + l_1 \cos \alpha_i) \cos \varphi_i - X \right)^2 + \left((r + l_1 \cos \alpha_i) \sin \varphi_i - Y \right)^2 + \left(-l_1 \sin \alpha_i - Z \right)^2 = l_2^2 \quad (3.9)$$

Qui, en la développant devient :

$$\begin{aligned} & (r + l_1 \cos(\alpha_i))^2 \cos^2 \varphi_i + X^2 - 2 X (r + l_1 \cos(\alpha_i)) \cos \varphi_i + \\ & (r + l_1 \cos(\alpha_i))^2 \sin^2 \varphi_i + X^2 - 2 X (r + l_1 \cos(\alpha_i)) \sin \varphi_i + Z^2 + \\ & (l_1 \sin(\alpha_i))^2 + 2 Z l_1 \sin(\alpha_i) = 2^2 \end{aligned} \quad (3.10)$$

Sachant que $\sin^2(x) + \cos^2(x) = 1$, l'équation (3.9) devient :

$$\begin{aligned} & (r + l_1 \cos(\alpha_i))^2 + X^2 - 2 X (r + l_1 \cos(\alpha_i)) \cos \varphi_i + Y^2 - 2 Y (r + \\ & l_1 \cos(\alpha_i)) \sin \varphi_i + Z^2 + (l_1 \sin(\alpha_i))^2 + 2 Z l_1 \sin(\alpha_i) = l_2^2 \end{aligned} \quad (3.11)$$

On sépare les inconnus d'un côté et les données d'un autre pour obtenir :

$$\cos(\alpha_i) [2 r l_1 - 2 X l_1 \cos \varphi_i - 2 Y l_1 \sin \varphi_i] + \sin(\alpha_i) [2 Z l_1] = l_2^2 - l_1^2 - r^2 - X^2 - Y^2 - Z^2 + 2 X r \cos \varphi_i + 2 Y r \sin \varphi_i \quad (3.12)$$

Et en posant :

$$A_i = 2 r l_1 - 2 X l_1 \cos \varphi_i - 2 Y l_1 \sin \varphi_i$$

$$B_i = 2 Z l_1$$

$$C_i = l_2^2 - l_1^2 - r^2 - X^2 - Y^2 - Z^2 + 2 X r \cos \varphi_i + 2 Y r \sin \varphi_i$$

L'équation (3.11) devient :

$$A_i \cos(\alpha_i) + B_i \sin(\alpha_i) = C_i \quad (3.13)$$

Avec les paramètres A_i , B_i et C_i connus à chaque instant. Afin de résoudre cette équation, on fait le changement de variable suivant :

$$\beta = \tan\left(\frac{\alpha_i}{2}\right) \quad \text{et donc} \quad \cos(\alpha_i) = \frac{1 - \beta^2}{1 + \beta^2} \quad \text{et} \quad \sin(\alpha_i) = \frac{2\beta}{1 + \beta^2}$$

On aura :

$$A_i \frac{1 - \beta^2}{1 + \beta^2} + B_i \frac{2\beta}{1 + \beta^2} = C_i$$

Ou

$$(1 - \beta^2)A_i + 2\beta B_i = (1 + \beta^2)C_i$$

Ou

$$(-A_i - C_i)\beta^2 + 2B_i\beta + (A_i - C_i) = 0 \quad (3.14)$$

La résolution de cette équation en prenant β comme inconnu donne :

$$\beta = \frac{-2B_i \pm \sqrt{4B_i^2 + 4(A_i + C_i)(A_i - C_i)}}{-2(A_i + C_i)}$$

En revenant à l'expression de β :

$$\frac{\alpha_i}{2} = \text{atan}\left(\frac{-2B_i \pm \sqrt{4B_i^2 + 4(A_i + C_i)(A_i - C_i)}}{-2(A_i + C_i)}\right)$$

Qui donne finalement :

$$\alpha_i = 2 \text{atan}\left(\frac{2B_i \pm \sqrt{4B_i^2 + 4(A_i + C_i)(A_i - C_i)}}{2(A_i + C_i)}\right)$$

Cette expression donne deux solutions qui vérifient toutes les deux le modèle géométrique. Le choix de l'une d'entre elles varie selon le critère de sélection. Un critère raisonnable serait de choisir la solution pour laquelle le robot bouge le moins, c'est-à-dire de prendre les angles α_i les plus petits possible. Ceci se traduit mathématiquement par en choisissant l'équation avec le signe « - » ; Ce qui donne :

$$\alpha_i = 2 \text{atan}\left(\frac{2B_i - \sqrt{4B_i^2 + 4(A_i + C_i)(A_i - C_i)}}{2(A_i - C_i)}\right) \quad (3.15)$$

Il s'agit maintenant de comparer le MGI développé antérieurement avec celui présenté dans [8] qui a déjà été validé avec le bloc SimMechanics du robot delta. En injectant la même trajectoire à l'entrée des deux modèles, nous constatons que la différence entre les deux blocs est très faible et peut être identifiée au zéro numérique. Ce résultat valide notre MGI.

3.3. Espace de travail

L'espace du travail est la zone que le point P, centre de la nacelle, peut atteindre. C'est une caractéristique très importante pour les robots et un critère indispensable pour le choix des solutions industrielles. En premier lieu, il est déterminé par l'intersection des volumes atteignables par les trois chaînes cinématiques constituant le Delta prises indépendamment [3] comme le montre la figure suivante :

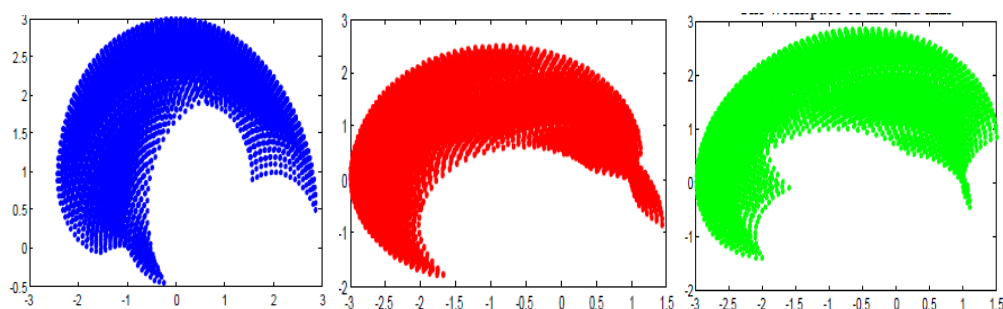


Fig.3.8 Espace de travail de chaque chaîne cinématique [11]

L'intersection donne :

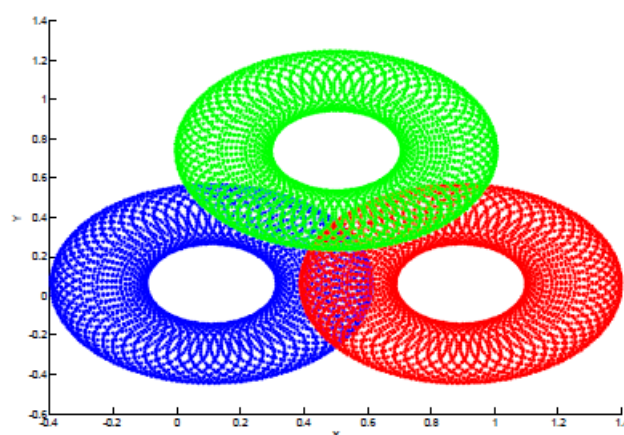


Fig.3.9 Intersection des trois espaces de travail de chaque chaîne [11]

Cette intersection est donnée dans [12] par :

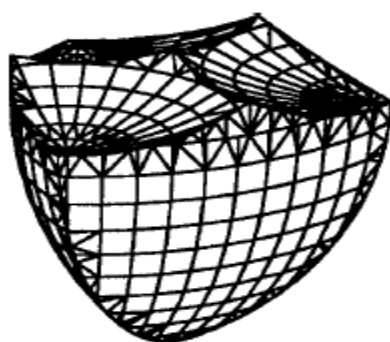


Fig.3.10 Espace de travail du Delta

Cet espace est ensuite limité en prenant en considération les contraintes et les singularités suivantes [9] [3] :

- Les contraintes de construction des articulations aux 2 extrémités des barres parallèles limitent l'angle γ . Pratiquement, nous avons $\gamma_{Max} = 40^\circ$.
- Afin d'éviter les interférences entre le bras et les barres parallèles ainsi que entre les barres parallèles et les moteurs nous prenons $\alpha_i + \beta_i > 30^\circ$. Or, pour éviter les ambiguïtés de transformation de coordonnées qui se produisent lorsque l'angle $(\alpha_i + \beta_i)$ devient supérieur à 180° (2 consignes articulaires pour un seul point de l'espace opérationnel). Nous concluons que $30^\circ < \alpha_i + \beta_i < 180^\circ$.

En prenant compte de ces contraintes, l'espace de travail du Delta sera considéré tel un cylindre de diamètre « D » et de hauteur « H » éloigné de « Zh » de la base fixe du robot comme le montre la figure suivante [3] :

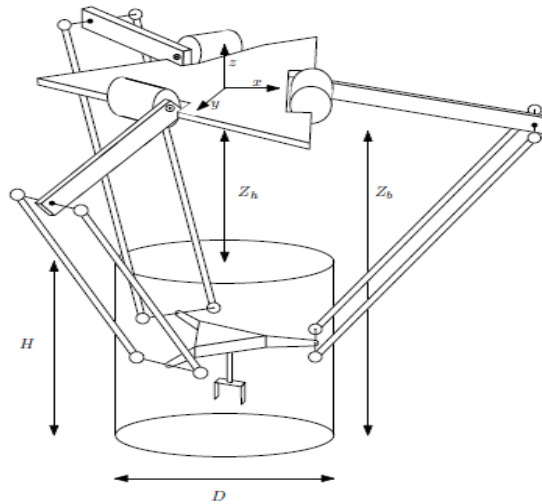


Fig.3.11 Espace de travail du Delta et ses paramètres [3]

Les expressions des paramètres H D et Zh sont les suivantes [9] :

$$D = 2 l_2 \sin \gamma_{MAX} = 385.6726 \text{ mm} ;$$

$$H = l_1 \sqrt{\left(1 + \frac{l_2}{l_1}\right)^2 - \left(\frac{r}{l_1} + 2 \frac{l_2 \sin \gamma_{Max}}{l_1 \sqrt{3}}\right)^2} - l_1 \sqrt{1 + \left(\frac{l_2}{l_1}\right)^2 - \frac{l_2}{l_1} \sqrt{3}} = 196.8884 \text{ mm} ;$$

$$Zh = -l_1 \sqrt{1 + \left(\frac{l_2}{l_1}\right)^2} - 2 \frac{l_2}{l_1} \cos(\alpha_i + \beta_i)_{MIN} = -161.4836 \text{ mm} ;$$

N.B. A noter que le montage d'un outil (Par exemple : Tournevis) provoquera un décalage du point de travail vers le bas et ainsi une translation correspondante de l'espace du travail.

3.4. Modélisation cinématique

La modélisation cinématique permet d'établir la relation entre les vitesses articulaires et les vitesses de l'organe terminal. Cette étude est indispensable car elle met en valeur la matrice Jacobienne « J » qui permet de déterminer les singularités du robot, c'est-à-dire les points de l'espace dans lesquels il y a ambiguïté de calcul et/ou de commande.

3.4.1. Modèle Cinématique Direct « MCD »

Le modèle cinématique direct est le passages des vitesses articulaires $\dot{\alpha}_i$ (ou $\dot{\theta}_i$) vers les vitesses de la nacelle $\dot{X}, \dot{Y}, \dot{Z}$. Nous reprenons les équations (3.8) :

$$\begin{aligned} X &= Z \frac{H5}{H2} + \frac{H4}{H2} = F1 \\ Y &= Z \frac{H1}{H2} + \frac{H3}{H2} = F2 \\ Z &= \frac{-M \pm \sqrt{M^2 - 4LN}}{2L} = F3 \end{aligned}$$

Ces équations relient X Y et Z avec des paramètres H_i , M, L et N qui dépendent à leur tour des α_i . Afin d'obtenir une relation sous la forme (3.9)

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = J \cdot \begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix}$$

Dérivons les équations de (3.8) par rapport au temps, nous obtenons ;

$$\begin{aligned} \dot{X} &= \frac{dF1}{dt} = \frac{dF1}{d\alpha} \frac{d\alpha}{dt} = \frac{dF1}{d\alpha} \dot{\alpha} \\ \dot{Y} &= \frac{dF2}{dt} = \frac{dF2}{d\alpha} \frac{d\alpha}{dt} = \frac{dF2}{d\alpha} \dot{\alpha} \\ \dot{Z} &= \frac{dF3}{dt} = \frac{dF3}{d\alpha} \frac{d\alpha}{dt} = \frac{dF3}{d\alpha} \dot{\alpha} \end{aligned}$$

Ce qui nous donne :

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \frac{dF1}{d\alpha} \\ \frac{dF2}{d\alpha} \\ \frac{dF3}{d\alpha} \end{pmatrix} \cdot \begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix}$$

Avec la matrice Jacobienne J (3.10) :

$$J = \begin{pmatrix} \frac{dF1}{d\alpha} \\ \frac{dF2}{d\alpha} \\ \frac{dF3}{d\alpha} \end{pmatrix}$$

A noter que J est une matrice 3x3 et se présente sous la forme matricielle générale suivante (3.11) :

$$J = \begin{pmatrix} \frac{dF1}{d\alpha1} & \frac{dF1}{d\alpha2} & \frac{dF1}{d\alpha3} \\ \frac{dF2}{d\alpha1} & \frac{dF2}{d\alpha2} & \frac{dF2}{d\alpha3} \\ \frac{dF3}{d\alpha1} & \frac{dF3}{d\alpha2} & \frac{dF3}{d\alpha3} \end{pmatrix}$$

Cette matrice n'est pas simple à calculer analytiquement comme c'est généralement le cas des robots séries. En effet, un calcul symbolique à l'aide du logiciel Matlab est nécessaire.

Nous comparons notre Modèle Cinématique Direct avec le bloc SimMechanics représentatif du robot delta. Nous leur injectons les mêmes angles alpha et nous observons l'erreur de résultats entre les deux blocs. Celle-ci est représentée sur la figure suivante :

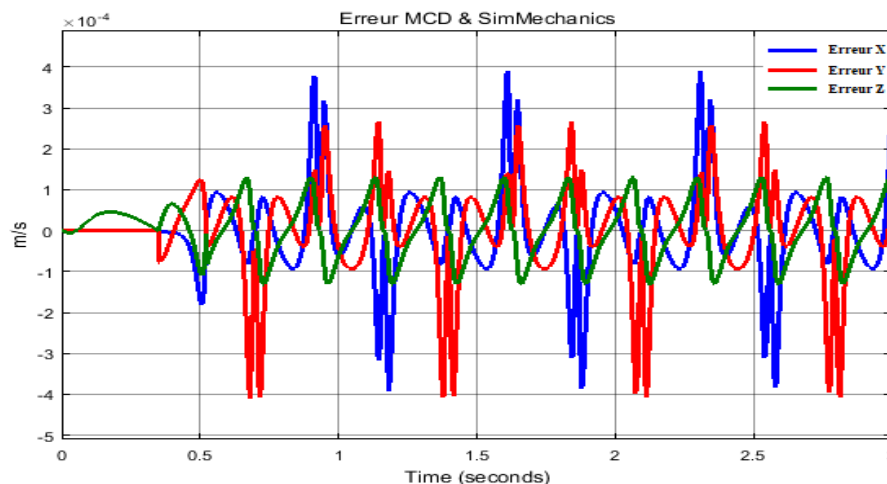


Fig.3.12 Erreur MCD

Comme nous pouvons le remarquer, l'erreur est de l'ordre de 10^{-4} pour une trajectoire variée. Nous pouvons donc valider notre Modèle Cinématique Direct et l'utiliser dans les schémas de simulation qui suivent.

3.4.2. Modèle Cinématique Inverse « MCI »

Le modèle cinématique inverse offre le passage des vitesses de la nacelle $\dot{X}, \dot{Y}, \dot{Z}$. vers les vitesses articulaires $\dot{\alpha}_i$ (ou $\dot{\theta}_i$).

Nous reprenons les équations (3.9) en multipliant les deux membres de l'équation par J^{-1} . Nous obtenons (3.10) :

$$\begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix} = J^{-1} \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix}$$

Cette relation (3.10) impose l'invisibilité de la matrice Jacobienne J ($\det(J) \neq 0$).

En fait, ceci introduit l'étude des singularités ou point singuliers qui sont définis comme les points de l'espace pour lesquels la matrice J n'est pas inversible (Et donc à un déterminant nul).

Finalement, nous comparons le MCI obtenu avec SimMechanics et nous visualisons les résultats.

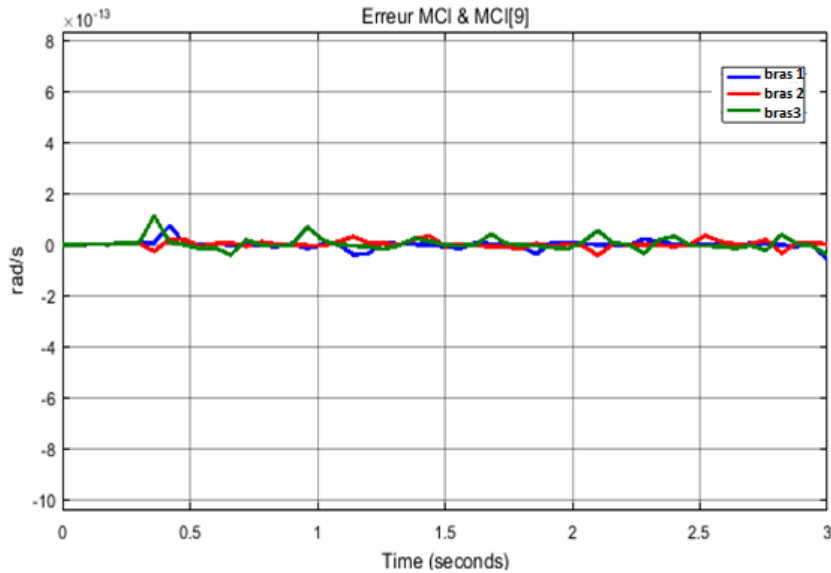


Fig.3.13 Erreur MCI

En effet, l'erreur de modélisation cinématique est très faible et peut être identifiée au zéro numérique. Cette erreur très faible entre le modèle cinématique inverse obtenu par dérivation symbolique et le bloc SimMechanics valide notre MCI obtenu.

3.5. Singularités

Pour un manipulateur parallèle, les singularités résultent du gain de un ou plusieurs degrés de liberté. En général, la résolution du déterminant de la matrice Jacobienne permet de déterminer facilement les singularités. Mais, vu la complexité de la matrice Jacobienne (Vue précédemment), la résolution de son déterminant est quasi-impossible. Ce qui a poussé le créateur de robot Delta, comme présenté dans [9] [10], à faire une analyse des positions des barres parallèles (parallélogrammes) les une par rapport aux autres. Ceci lui a permis de définir les configurations pour lesquelles la nacelle gagne en degrés de liberté. Ces configurations sont les suivantes :

1. Les trois paires de barres parallèles sont parallèles l'une à l'autre à une même direction (Fig.3.21.a).
2. Les trois parallélogrammes sont sur le même plan horizontal ou sur trois plans parallèles (Fig.3.21.b).
3. Deux des trois parallélogrammes ont leurs barres sur le même plan ou sur deux plans parallèles (Fig.3.21.c).
4. Deux des trois parallélogrammes ont leurs barres parallèles (Fig.3.21.d).

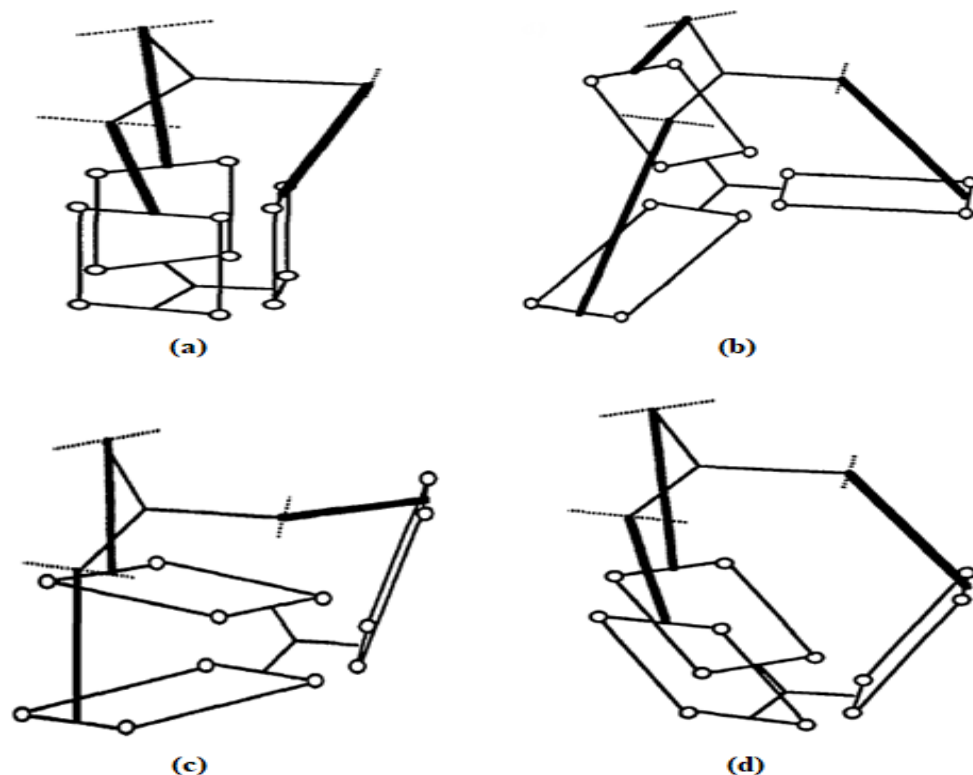


Fig.3.14 Les configurations singulières du robot Delta [9] [10]

3.6. Modèle des accélérations direct (MAD)

Le modèle des accélérations direct nous permet le passage des accélérations articulaires vers les accélérations de la nacelle. Soit le modèle géométrique déjà établi :

$$\overrightarrow{OP} = f(\alpha)$$

C'est une fonction analytique, donc pour trouver le modèle des accélérations, nous avons utilisé le calcul symbolique, qui permet de dériver analytiquement les fonctions explicite, cette méthode nous donne rapidement la deuxième dérivée du model géométrique, qui correspond au modèle des accélérations direct.

$$\ddot{\overrightarrow{OP}} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{d^2 \overrightarrow{OP}}{dt^2} (\alpha, \dot{\alpha}, \ddot{\alpha})$$

Pour valider ce modèle nous l'avons comparé avec bloc créé sur SimMechanics. Les résultats de la simulation montrent l'erreur en accélération :

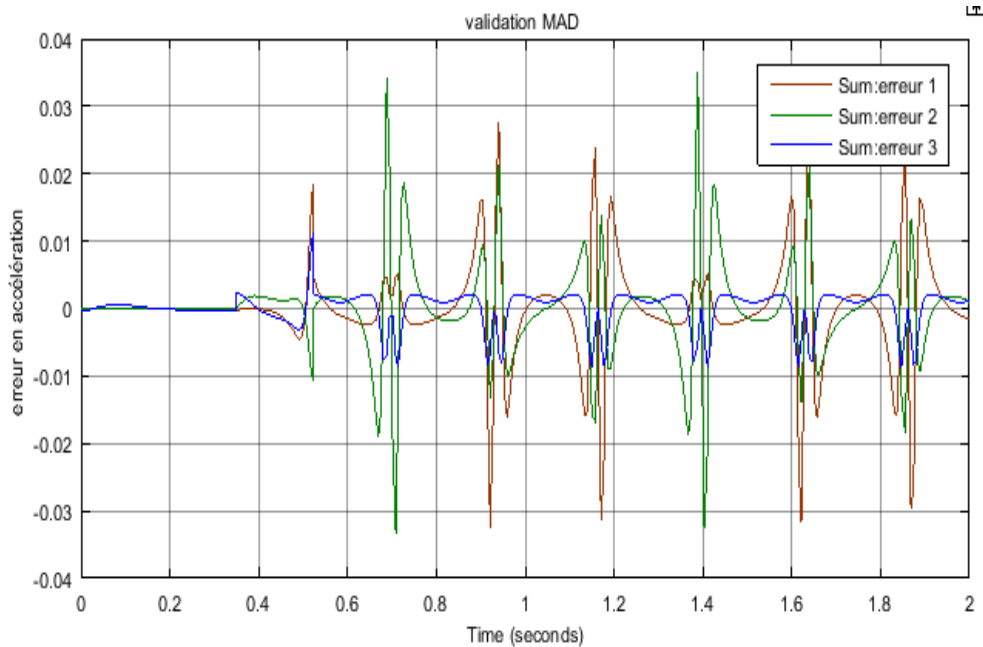


Fig.3.15 Erreur en accélération entre le MAD et SimMechanics

On constate que l'erreur est vraiment petite, l'erreur relative est de 0.16%, ce qui valide notre modèle.

3.7. Modélisation dynamique

Dans la Robotique on trouve trois méthodes utilisées pour trouver le model dynamique du robot : l'utilisation des équations de Lagrange, la méthode basée sur le travail virtuel, et la méthode classique des équations de Newton Euler. Dans ce qui suit, nous allons utiliser cette dernière méthode, qui consiste à mettre en relation le mouvement du solide avec le torseur des forces et des moments appliqué sur le solide.

Dans notre cas nous allons appliquer cette loi sur les bras de notre robot, pour cela nous devons rapporter toutes les forces appliquées sur les barres parallèles et la nacelle ainsi que leurs résultantes dynamiques, aux points C_i des bras.

Forces appliqué sur le robot :

Chaque corps du robot est soumis à sa propre force de gravitation, et des forces de liaisons qui régissent son mouvement.

Nacelle :

- Forces de gravité et de liaisons :

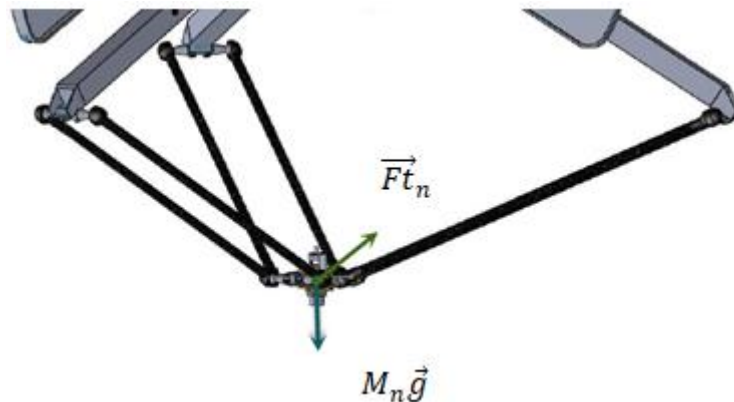


Fig.3.16 de gravité et résultante dynamique de la nacelle

- Résultantes dynamiques :

La nacelle dans notre cas à 3 degré de liberté, elle n'admet pas de rotation. Il n'existe pas de moment appliqué sur la nacelle, d'où le moment de la résultante dynamique est nul, donc la seule résultante dynamique est celle de la translation, et on a :

$$\vec{Ft}_n = M\ddot{OP}$$

Figure 3.7.1 Force de gravité et résultantes dynamique des barres parallèles

Barres parallèles :

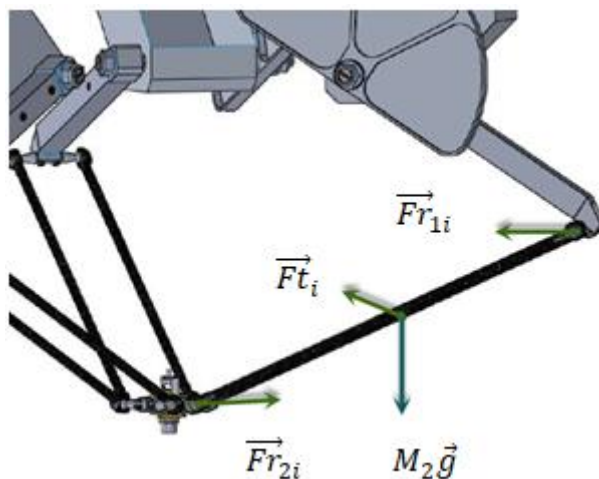


Fig.3.17 Force de gravité et résultantes dynamique des barres parallèles

- Forces de gravité :

Afin de simplifier le calcul nous avons supposé que la masse de chaque paire est répartie aux extrémités, ce qui est équivalent au cas réel, ce n'est donc pas une approximation de cette force.

La force de gravitation devient comme indiqué dans la Fig.3.23 :

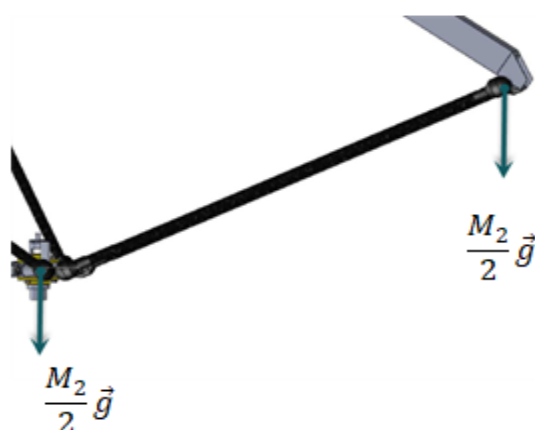


Fig.3.18 Simplification de la force de gravité des barres parallèles

- Résultantes dynamiques :

Chaque paire de barres parallèles subit, en plus de la gravité , des forces de liaisons aux deux extrémités engendrant un mouvement de translation et un mouvement de rotation autour du centre de gravité

(loi fondamentale de la dynamique) , donc on a décomposé le tenseur dynamique de ces barres en la somme d'un tenseur glisseur et d'un tenseur couple. Le tenseur glisseur a comme résultante :

$$\vec{F}t_i = M_2 \ddot{OG}_1$$

Nous définissons 2 vecteurs à l'extrémité des barres équivalents à la résultante du tenseur glisseur :

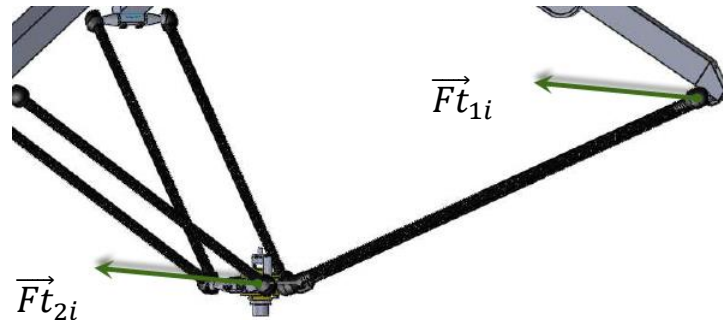


Fig.3.19 Simplification de la résultante dynamique de translation des barres parallèles

On a :

$$\vec{F}t_{1i} = \vec{F}t_{2i} = \frac{\vec{F}t_i}{2} = \frac{M_2}{2} \ddot{OG}_1$$

On a donc :

$$\vec{F}t_{2i} + \vec{F}t_{1i} = \vec{F}t_i$$

Le moment de réduction du tenseur couple au centre de gravité de la paire de barres parallèles est :

$$\overline{M(G_1)} = J_2 \ddot{\theta} \vec{u}$$

Avec :

\vec{u} : Direction de l'axe de rotation.

J_2 : Moment d'inertie de la paire des barres parallèles au point G_1 .

Nous allons maintenant construire un couple de vecteurs équivalents au tenseur couple, à l'extrémité des barres, ces vecteurs sont donc perpendiculaires à l'axe de rotation et au vecteur $\overrightarrow{PC_1}$. On a :

$$\begin{aligned} \overrightarrow{Fr_{1i}} + \overrightarrow{Fr_{2i}} &= 0 \\ \overrightarrow{Fr_{1i}} &= Fr_i \\ \overrightarrow{G_1C_1} \wedge \overrightarrow{Fr_{1i}} + \overrightarrow{G_1P} \wedge \overrightarrow{Fr_{2i}} &= \overline{M(G_1)} \\ \Rightarrow Fr_i \cdot l_2 &= J_2 \ddot{\theta} \\ Fr_i &= \frac{J_2 \ddot{\theta}}{l_2} \end{aligned}$$

Et :

$$\begin{aligned} \overrightarrow{G_1C_1} &= \frac{l_2}{2} \frac{\overrightarrow{Fr_{1i}}}{Fr_i} \\ \Rightarrow \overrightarrow{Fr_{1i}} &= 2 \frac{J_2}{l_2} \overrightarrow{G_1C_1} \end{aligned}$$

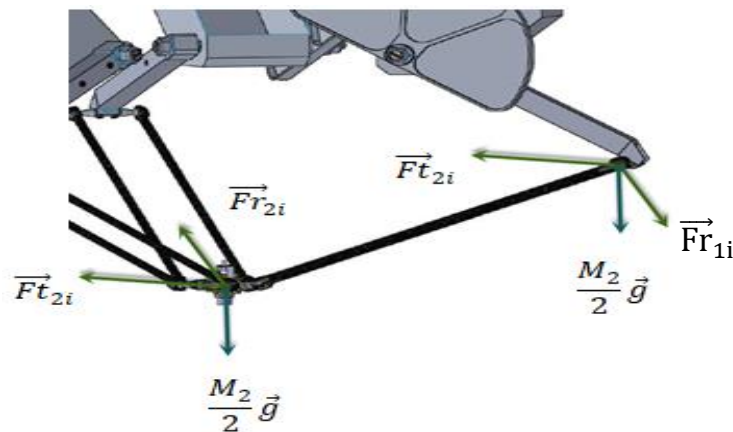


Fig.3.20 Forces et résultantes dynamiques simplifiées des barres parallèles

Finalement les forces extérieures et les résultantes dynamiques de la nacelle et des avants bras sont regroupées dans la figure suivante :

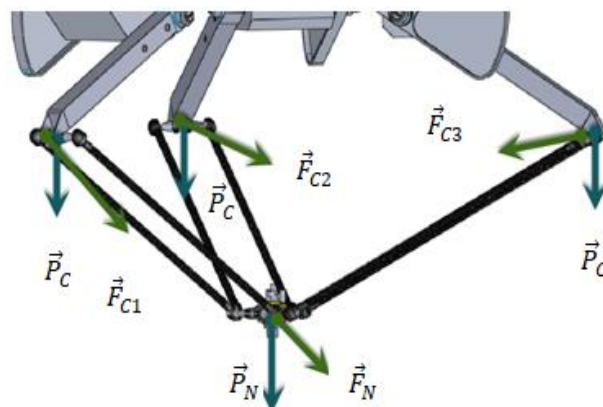


Fig.3.21 Forces extérieures et résultantes dynamiques générales

Avec :

$$\vec{P}_C = \frac{M_2}{2} \vec{g}$$

$$\vec{P}_N = \left(\frac{3M_2}{2} + M_3 \right) \cdot \vec{g}$$

$$\vec{F}_{C_1} = \frac{M_2}{2} \ddot{\vec{O}G_1} + 2 \frac{J_2}{l_2^2} \ddot{\vec{G}_1C_1}$$

$$\vec{F}_N = M_n \ddot{\vec{O}P} + \sum_{i=1}^3 \left(-2 \frac{J_2}{l_2^2} \ddot{\vec{G}_1C_1} + \frac{M_2}{2} \ddot{\vec{O}G_1} \right)$$

Forces et résultantes dynamiques aux points Ci :

Afin de pouvoir écrire l'équation de Newton Euler pour les bras du robot, nous devons avoir tous les forces extérieures et les résultantes dynamiques des autres corps du robot rapportés aux points de contact avec ces bras, ces forces et résultantes seront représentées par des vecteurs aux points C_i . Pour cela nous allons décomposer les vecteurs \vec{P}_N et \vec{F}_N dans une nouvelle base $\{\vec{U}_1, \vec{U}_2, \vec{U}_3\}$. Chaque vecteur de cette base est porté sur un avant-bras comme c'est montré sur la figure 3.29.

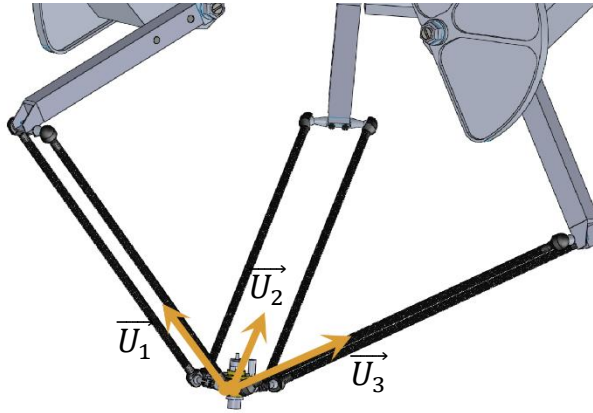


Fig.3.22 Vecteurs unitaires portés sur les barres parallèles

Cela nous donnera les composantes des vecteurs \vec{P}_N et \vec{F}_N portés sur les avant-bras.

On a la matrice de transformation :

$$U = [\vec{U}_1 \ \vec{U}_2 \ \vec{U}_3]$$

$$\vec{P}_{N1} = \vec{U}_1 (1 \ 0 \ 0) U^{-1} \vec{P}_N$$

$$\overrightarrow{P_{N2}} = \overrightarrow{U_2} (0 \ 1 \ 0) U^{-1} \overrightarrow{P_N}$$

$$\overrightarrow{P_{N3}} = \overrightarrow{U_3} (0 \ 0 \ 1) U^{-1} \overrightarrow{P_N}$$

$$\overrightarrow{F_{N1}} = \overrightarrow{U_1} (1 \ 0 \ 0) U^{-1} \overrightarrow{F_N}$$

$$\overrightarrow{F_{N2}} = \overrightarrow{U_2} (0 \ 1 \ 0) U^{-1} \overrightarrow{F_N}$$

$$\overrightarrow{F_{N3}} = \overrightarrow{U_3} (0 \ 0 \ 1) U^{-1} \overrightarrow{F_N}$$

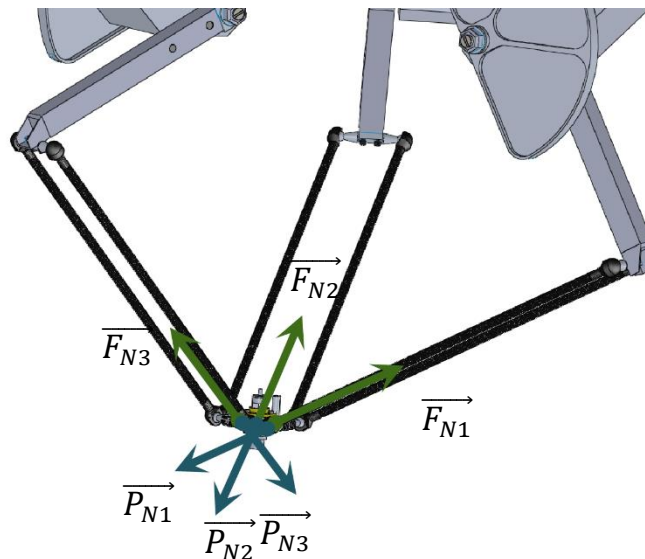


Fig.3.23 décomposition des forces et des résultantes dynamiques

Le principe de l'action réaction nous permet de transporter ces forces et résultantes aux points C_i .

On a donc à chaque point C_i :

$$\overrightarrow{P_i} = \overrightarrow{P_{N1}} + \overrightarrow{P_C}$$

$$\overrightarrow{F_i} = \overrightarrow{F_{N1}} + \overrightarrow{F_{C1}}$$

-pour chaque bras nous nous intéressons qu'aux composantes des forces et des résultantes dynamiques se trouvant sur le plan (OX_iZ_i) car l'axe de rotation de chaque bras i est normal au plan (OX_iZ_i) .

Donc on tire ces composantes comme indiqué ci-dessous :

$$\overrightarrow{P_{1_{oxz}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot {}^R_i T \cdot \overrightarrow{P_1}$$

$$\overrightarrow{F_{1_{oxz}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot {}^R_i T \cdot \overrightarrow{F_1}$$

Equation de Newton-Euler :

Nous allons maintenant écrire l'équation de Newton-Euler pour chaque bras.

Forces appliquées sur les bras :

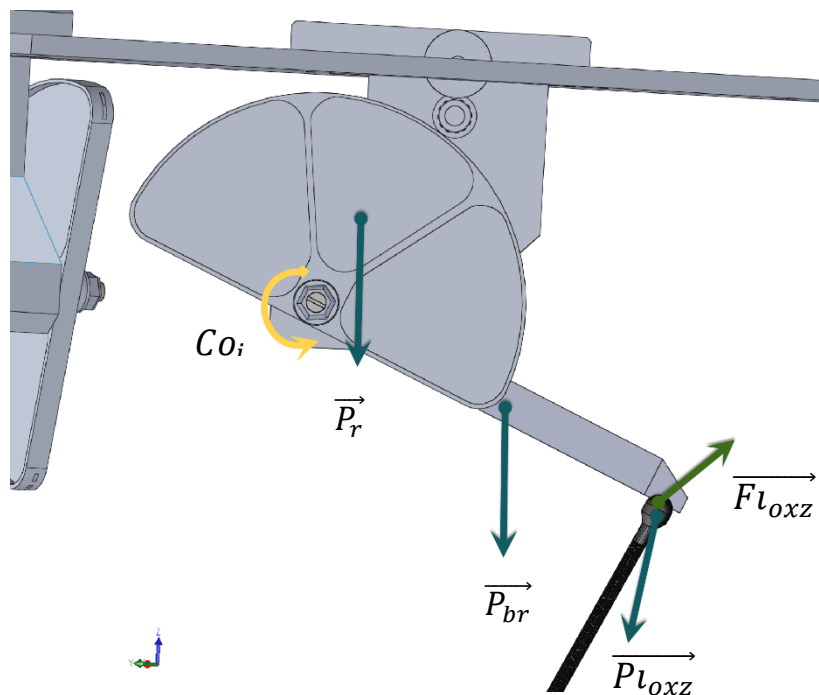


Fig.3.24 forces extérieures et résultantes dynamiques extérieures

Avec :

$$\overrightarrow{P_{br}} = M_{br} \vec{g}$$

$$\overrightarrow{P_r} = M_{br} \vec{g}$$

Les bras font un mouvement de rotation autours du point Ai et dans le plan $(OX_i Z_i)$.

L'équation de Newton-Euler s'écrit comme il suit :

$$\begin{aligned} \text{Co}_i + (010)\overrightarrow{A_1C_1} \wedge \overrightarrow{P1_{\text{oxz}}} + (010)\overrightarrow{A_1G_{\text{br}}} \wedge \overrightarrow{P_{\text{br}}} + (010)\overrightarrow{A_1G_r} \wedge \overrightarrow{P_r} \\ = J\dot{\alpha}_i + (010)\overrightarrow{A_1C_1} \wedge \overrightarrow{F1_{\text{oxz}}} \end{aligned}$$

Avec :

$$\begin{aligned} \overrightarrow{A_1C_1} &= \begin{pmatrix} l1 \cos(\alpha_i) \\ 0 \\ -l1 \sin(\alpha_i) \end{pmatrix} \\ \overrightarrow{A_1G_{\text{br}}} &= \begin{pmatrix} lbr \cos(\alpha_i - 3.75) \\ 0 \\ -lbr \sin(\alpha_i - 3.75) \end{pmatrix} \\ \overrightarrow{A_1G_r} &= \begin{pmatrix} lr \sin(\alpha_i - 3.75) \\ 0 \\ lr \sin(\alpha_i - 3.75) \end{pmatrix} \end{aligned}$$

Maintenant qu'on a l'équation du modèle dynamique on peut avoir le modèle inverse et direct du robot.

3.7.1. Modèle Dynamique Inverse « MDI »

La réécriture du model dynamique nous donne :

$$\text{Co} = J\ddot{\alpha} + f_1(\alpha, \dot{\alpha}, \ddot{\alpha}) - f_2(\alpha)$$

Avec :

$$\begin{aligned} f_1(\alpha, \dot{\alpha}, \ddot{\alpha}) &= \begin{pmatrix} (010)\overrightarrow{A_1C_1} \wedge \overrightarrow{F1_{\text{oxz}}} \\ (010)\overrightarrow{A_2C_2} \wedge \overrightarrow{F2_{\text{oxz}}} \\ (010)\overrightarrow{A_3C_3} \wedge \overrightarrow{F3_{\text{oxz}}} \end{pmatrix} \\ f_2(\alpha) &= \begin{pmatrix} (010)(\overrightarrow{A_1C_1} \wedge \overrightarrow{P1_{\text{oxz}}} + \overrightarrow{A_3G_{\text{br}1}} \wedge \overrightarrow{P_{\text{br}}} + \overrightarrow{A_1G_{r1}} \wedge \overrightarrow{P_r}) \\ (010)(\overrightarrow{A_2C_2} \wedge \overrightarrow{P2_{\text{oxz}}} + \overrightarrow{A_2G_{\text{br}2}} \wedge \overrightarrow{P_{\text{br}}} + \overrightarrow{A_2G_{r2}} \wedge \overrightarrow{P_r}) \\ (010)(\overrightarrow{A_3C_3} \wedge \overrightarrow{P3_{\text{oxz}}} + \overrightarrow{A_3G_{\text{br}3}} \wedge \overrightarrow{P_{\text{br}}} + \overrightarrow{A_3G_{r3}} \wedge \overrightarrow{P_r}) \end{pmatrix} \end{aligned}$$

Pour valider ce modèle, nous avons comparé notre model dynamique inverse avec le model généré par SimMechanics, pour une trajectoire désirée. Le schéma de simulation SIMULINK est le suivant :

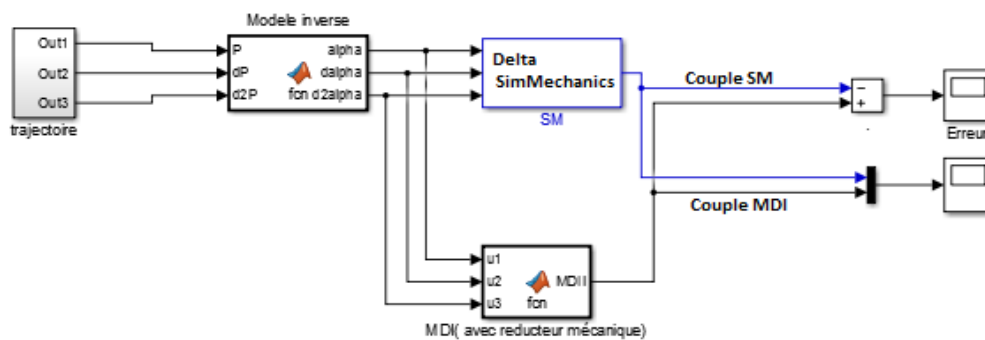


Fig.3.25 Schéma de structure de validation du MDI

Les résultats de la simulation sont :

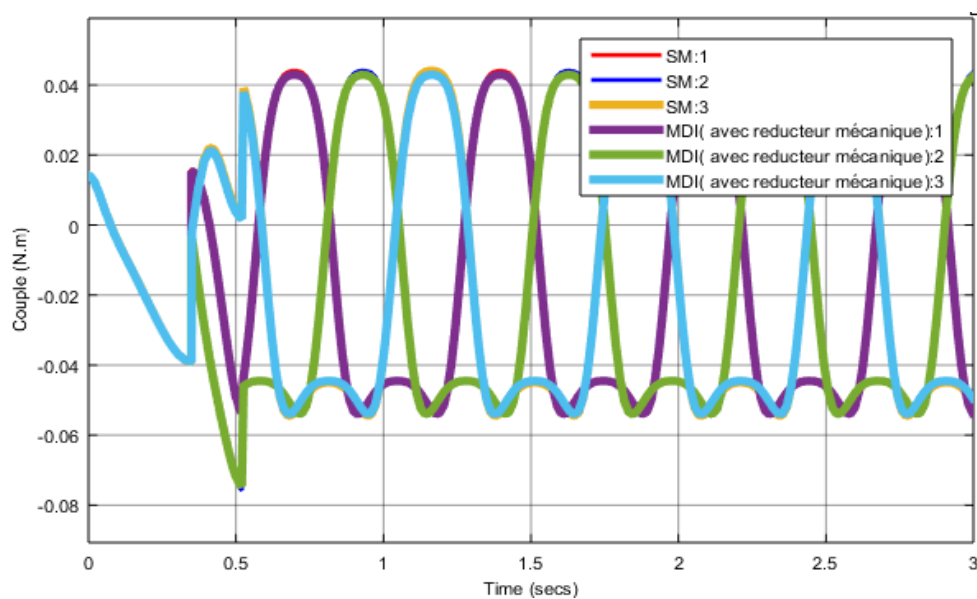


Fig.3.26 Couples donnés par les deux le MDI et SimMechanics

L'erreur est donnée par :

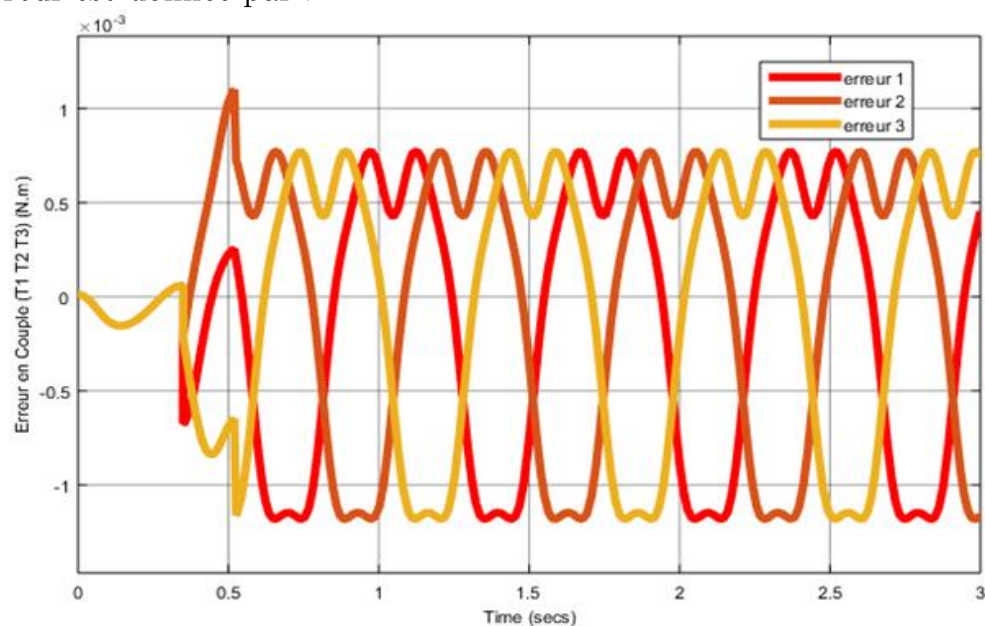


Fig.3.27 Erreur entre notre MDI et celui de SimMechanics.

L'erreur est de l'ordre de 10^{-3} , et se traduit à une erreur relatif de 2.5%, ce qui est bon comme résultat, donc notre modèle est validé.

3.7.1. Modèle Dynamique Direct « MDD »

La réécriture du model dynamique nous donne :

$$\ddot{\alpha} = \text{inv} \left(\begin{pmatrix} J & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & J \end{pmatrix} + A(\alpha) \right) \cdot (C_0 + f_2(\alpha) - B(\alpha, \dot{\alpha}))$$

Avec :

$$A(\alpha) \cdot \ddot{\alpha} + B(\alpha, \dot{\alpha}) = f_1(\alpha, \dot{\alpha}, \ddot{\alpha})$$

3.8. Représentation d'état

Pour avoir le model d'état, nous devons définir notre commande, et les sorties du système, dans le cas du robot ISIR88, la commande du couple des bras se fait moyennant un réducteur mécanique et un moteur à courant continu. Donc à travers la tension d'entrée de notre système nous commandons le courant désiré du moteur, et vue le fait que la constante de temps électrique est très petite par rapport à la constante mécanique, et qu'on a pas de capteurs de courant, la dynamique du courant est négligée, et l'asservissement du courant est représenté par un gain.

$$I = K \cdot U$$

Ce qui nous donne le couple mécanique délivré par le moteur :

$$C_{mo} = K_I \cdot I$$

Avec :

U : Tension d'entrée de notre système.

K : Gain entre la tension et le courant.

K_I : Constante de courant du moteur.

Ces couples délivrés par les moteurs sont transmis aux axes de rotation des bras du robot à travers des réducteurs, qui amplifient ces couples et nous donnent les actuels couples aux axes de rotation des bras que nous commandons.

$$C = K_r \cdot C_{mo}$$

C Étant le couple aux axes de rotation des bras.

$K_r = 12$ Gain du réducteur mécanique.

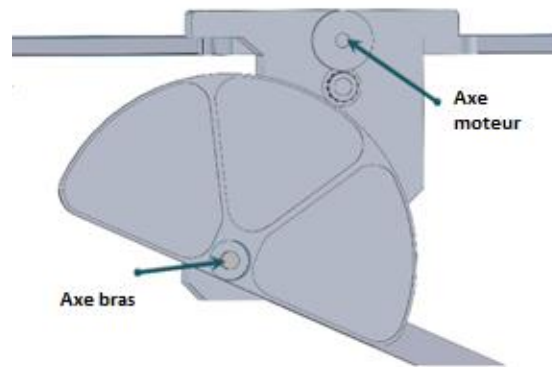


Fig.3.28 Réducteur mécanique

Ce qui nous donne enfin la relation entre notre commande et les couples aux axes de rotation des bras :

$$C = K_r \cdot K_l \cdot K \cdot U$$

Un travail d'identification a été fait sur le robot pour trouver ce gain final entre les tensions de commande et les couples aux axes de rotation des bras et le résultat est le suivant :

$$C = K_f \cdot U \quad \text{avec} \quad K_f = 3.08 \text{ N.m/V}$$

En robotique on s'intéresse presque toujours aux positions et aux vitesses des articulations, donc nous allons prendre comme sortie de notre système ces deux grandeurs.

On a donc :

$$\begin{aligned} X_1 &= \alpha \\ X_2 &= \dot{\alpha} \end{aligned}$$

Le model d'état finale est le suivant :

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \text{inv} \left(\begin{pmatrix} J & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & J \end{pmatrix} + A(X_1) \right) \cdot (3.08 U + f_2(X_1) - B(X_1, X_2)) \end{aligned}$$

3.9. Conclusion

Dans ce chapitre, nous avons présenté les caractéristiques géométriques et cinématiques du robot Delta ISIR88.

Le modèle géométrique du robot Delta établi par R. Clavel que nous avons adopté nous permet de passer de l'espace articulaire vers l'espace opérationnel et inversement. Puis, en exploitant ce modèle géométrique on a pu aboutir à un modèle cinématique par dérivation numérique.

Ensuite, une modélisation dynamique a été effectuée selon la méthode de Newton Euler en prenant compte de la roue de réduction existant entre le bras et l'actionneur. Le modèle dynamique direct sera utile pour simuler les algorithmes de commande et le modèle inverse sera utilisé en régulation (couple a priori). De cette modélisation découle aussi la représentation d'état indispensable à la synthèse de plusieurs algorithmes de commande en effectuant un choix convenable des variables d'état.

Pour une meilleure exploitation du robot Delta ISIR88, nous avons défini son volume opérationnel de travail et ses singularités ce qui nous permettra de respecter les contraintes physiques lors de la phase expérimentale.

A noter que tout au long de ce chapitre, la validation des modèles établis s'est faite en les comparant avec le bloc SimMechanics conçu par N. I. Mokhtari et H. Nadour.

CHAPITRE 4

Commande et Simulation

Chapitre 4: Commande et Simulation

4.1. Introduction

Théoriquement si le modèle dynamique était parfait, le suivi de trajectoire devrait l'être aussi car le modèle dynamique direct du robot exprime les équations du mouvement naturel du robot [3]. Le problème est que le modèle dynamique du robot, aussi réaliste soit-il, ne peut jamais être parfait à cause des hypothèses simplificatrices introduisant des erreurs de modélisation, les perturbations qui peuvent agir sur le robot et les erreurs du calcul numérique. Afin de vérifier ce phénomène, le schéma de commande suivant est proposé :

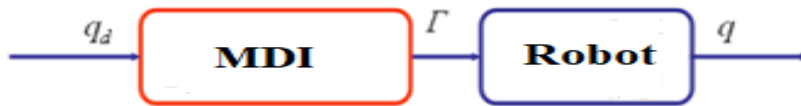


Fig.4.1 Schéma de commande en boucle ouverte

Nous visualisons la trajectoire désirée et celle prise par le robot :

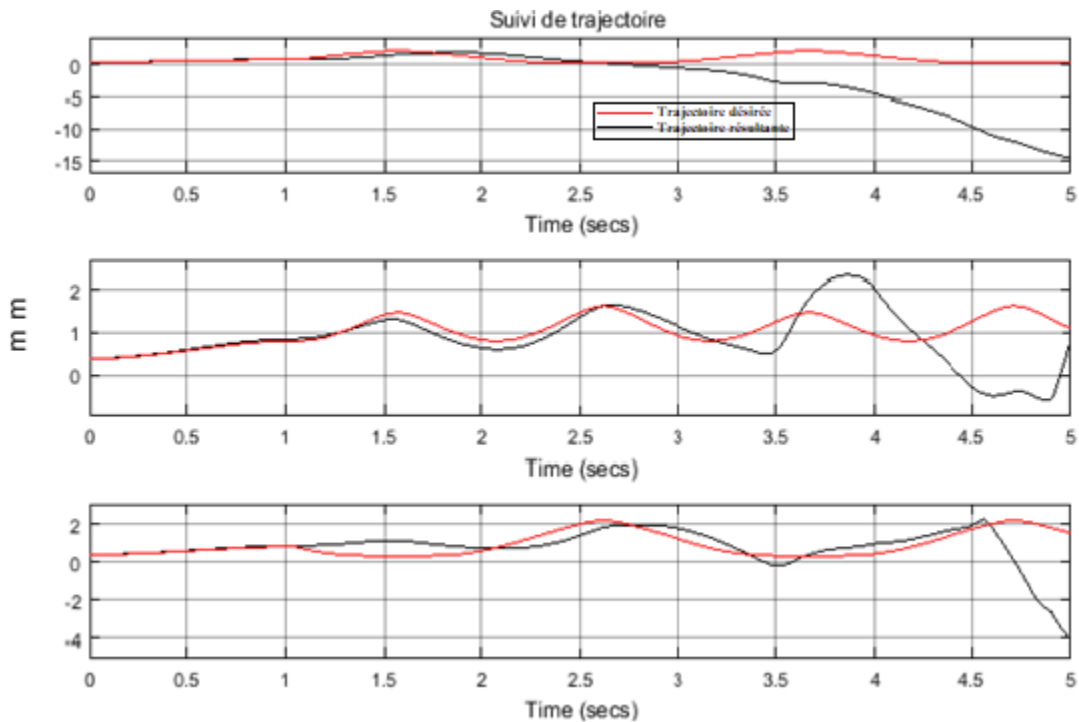


Fig.4.2 Suivi de trajectoire en BO (PID)

Au début, le robot suit assez bien la trajectoire, mais au bout de quelques secondes, il commence à dévier de la trajectoire de consigne. Ce comportement résulte des erreurs de modélisation du MDI et nous montre

clairement que la commande en boucle ouverte n'est pas suffisante pour satisfaire les performances désirées. Pour régler ce problème, la solution serait d'introduire un régulateur qui reçoit la sortie du système et la consigne et délivre le couple à appliquer. Le schéma de commande suivant montre bien l'action du régulateur :

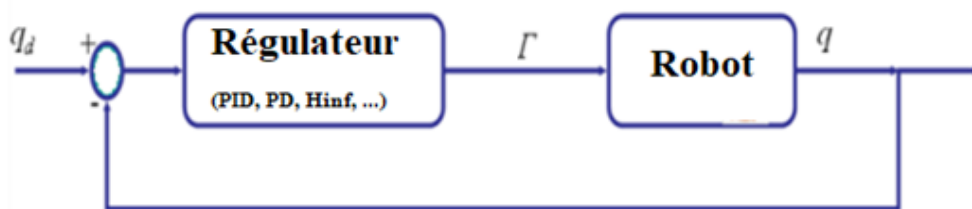


Fig.4.3 Schéma de commande avec régulateur

Les résultats obtenus ainsi sont :

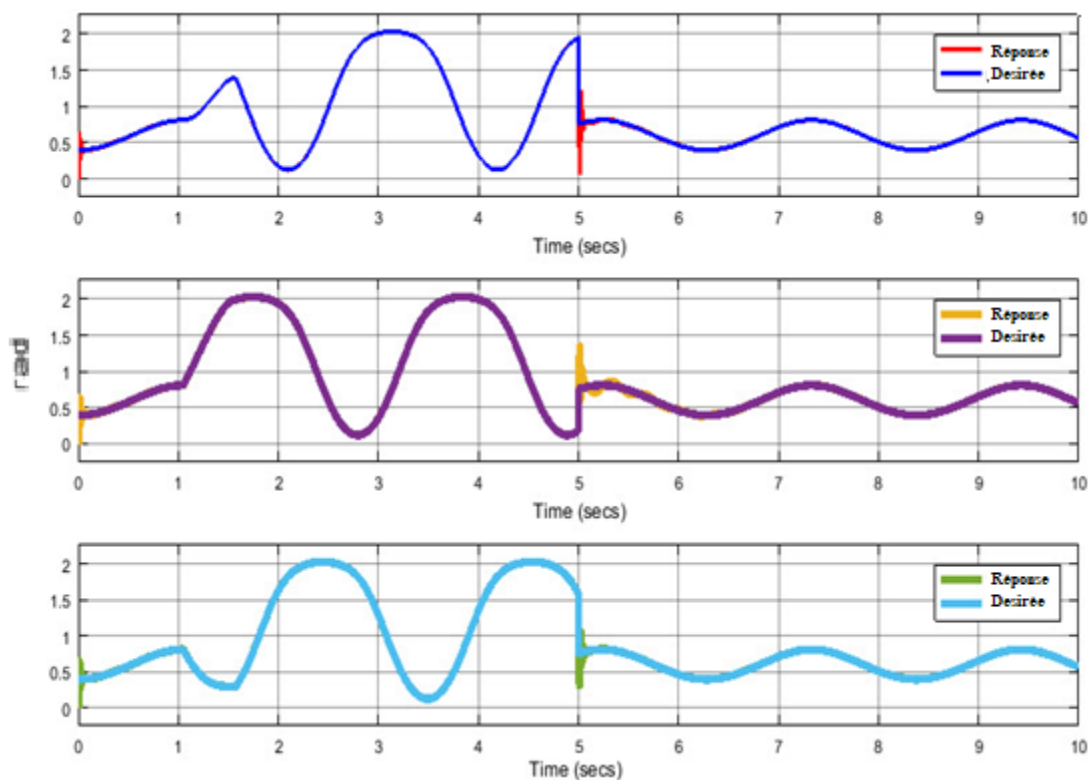


Fig.4.4 Suivi de trajectoire articulaire pour la commande avec régulateur (PID)

Le robot suit bien la trajectoire désirée, mais on constate certains pics indésirables et une erreur qui tend à converger mais lentement comme le montre la figure suivante :

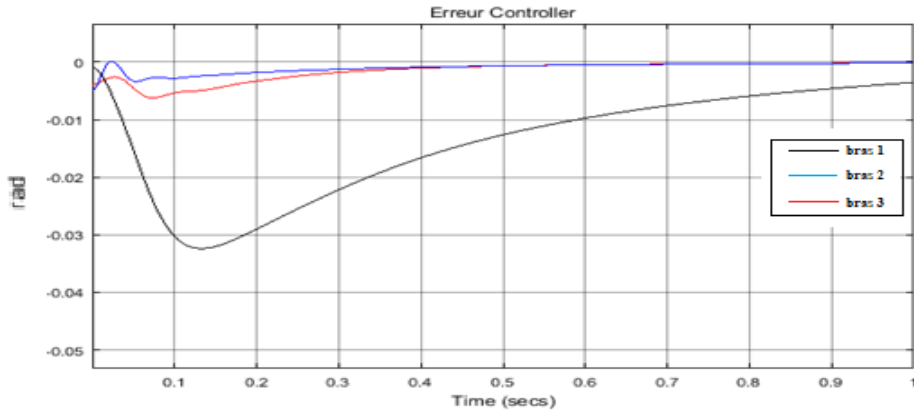


Fig.4.5 Erreur de Suivi de trajectoire pour la commande avec régulateur

Ces erreurs sont principalement dues aux problèmes de couplage et de non linéarité du système. L'introduction du modèle dynamique inverse s'impose donc afin d'améliorer le comportement du régulateur. Le schéma de régulation devient :

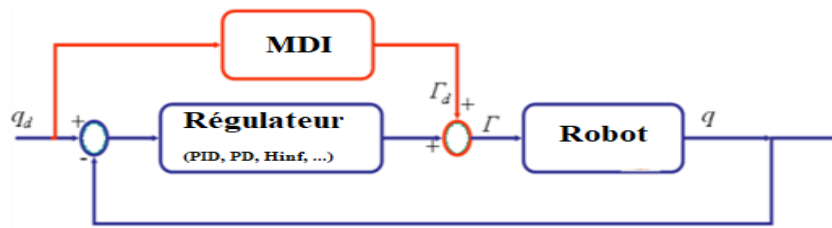


Fig.4.6 Schéma de commande Régulateur avec MDI a priori

En fait, si nous prenons un régulateur PID par exemple, L'ajout du MDI lui permet au de travailler autour d'un point de fonctionnement. Ce point de fonctionnement est ainsi déplacé de façon non linéaire suivant la dynamique inverse du robot. Ceci améliore nettement le travail du régulateur comme le montre la figure suivante :

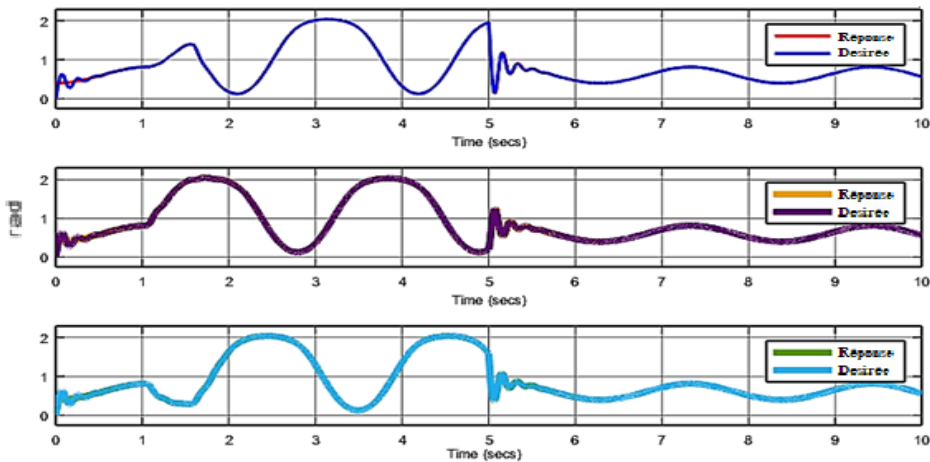


Fig.4.7 Suivi de trajectoire pour la commande avec régulateur et MDI à priori

Le suivi de la consigne est ainsi meilleur pour deux trajectoires différentes.

Afin de mieux voir la différence entre les trois techniques de commandes, nous comparons les trois erreurs comme suit :

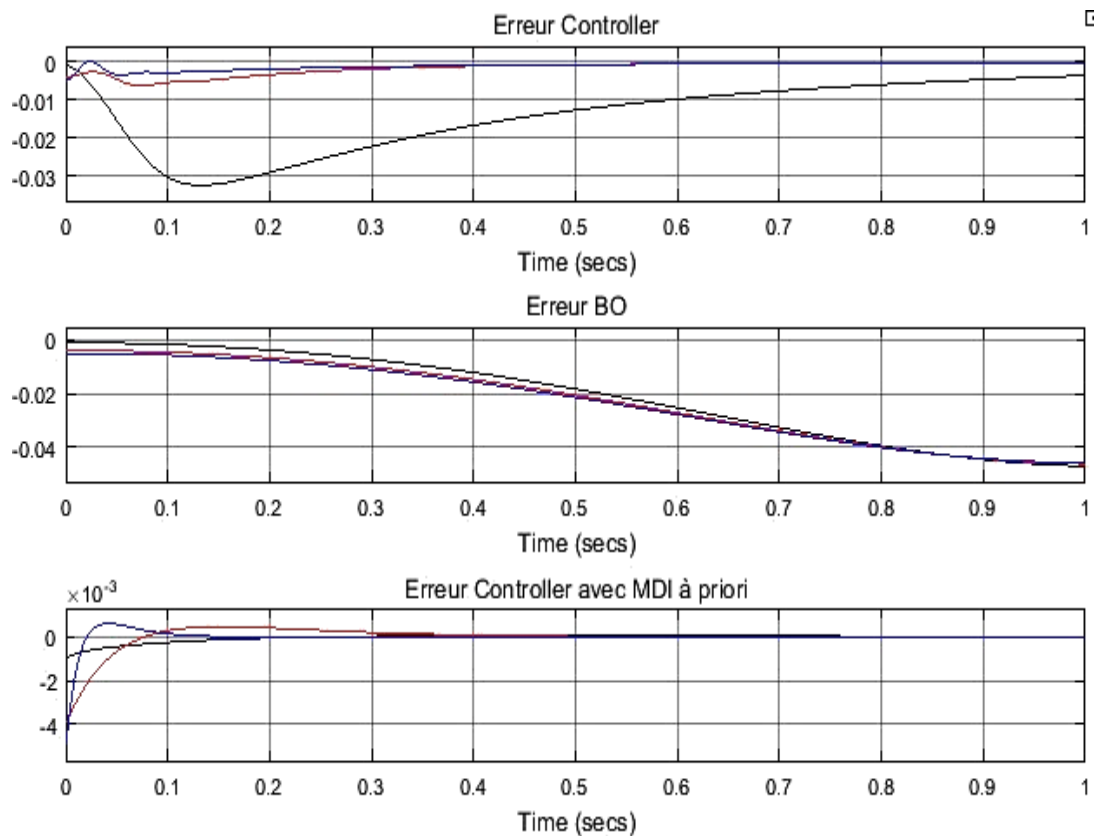


Fig.4.8 Erreurs de position pour les trois techniques de commande (en m)

Nous constatons qu'en boucle ouverte, l'erreur augmente avec le temps. Avec l'introduction d'un régulateur, l'erreur converge vers 0 mais assez lentement et les problèmes de non linéarité et de couplage posent toujours problème. Enfin, avec l'ajout du MDI, le suivi de trajectoire est excellent et l'erreur tend rapidement vers 0. A partir des résultats obtenus on conclut que le meilleur schéma de commande est « Le régulateur avec MDI à priori ».

Dans les paragraphes à venir, nous allons nous intéresser aux méthodes de synthèses de certains types de régulateurs parmi les plus utilisés en robotique. Une étude comparative est ensuite primordiale à la fin pour en choisir le/les meilleur(s).

4.2. Trajectoire de consigne

Avant d'entamer la synthèse des régulateurs, il faut tout d'abord choisir une trajectoire (consigne) adéquate. En fait, en industrie, le robot delta est principalement utilisé dans les opérations de prise-dépose. En observant plusieurs exemples de ce type d'opérations, nous proposons la trajectoire suivante avec une cadence d'une opération par seconde (aller et retour):

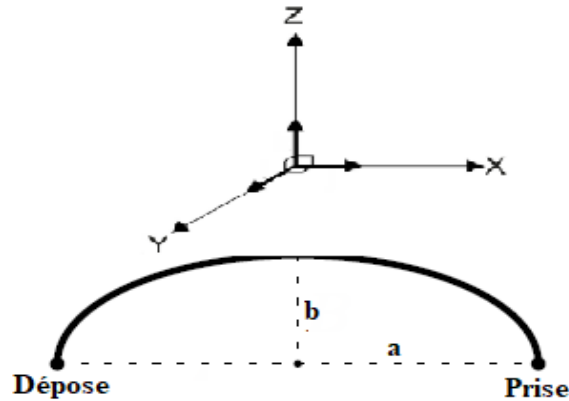


Fig.4.9 Trajectoire proposée de prise-dépose

Cette trajectoire peut être assimilée à une semi ellipse et ses lois de mouvements sont connues (voir [3]).

4.3. Commande avec régulateur PD

La commande classique PD est largement utilisée dans le domaine de la robotique industrielle. C'est d'ailleurs le cas du robot ISIR88. En effet, un tel régulateur est implémenté par défaut pour le robot Delta ISIR88 avec comme paramètres $Kp=2,2$ et $Kd=0,0145$ sous l'équation suivante :

$$u = Kp.e + Kd.\dot{e}$$

Or, comme expliqué dans le début de ce chapitre, les performances d'un régulateur deviennent meilleures et plus intéressantes lors de l'incorporation du couple à priori. De ce fait, deux schémas de commande sont envisagés; sans et avec l'incorporation du modèle dynamique inverse du robot (voir figure suivante).

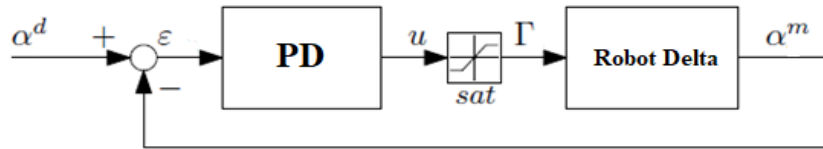


Fig.4.10.a Schéma de commande PD

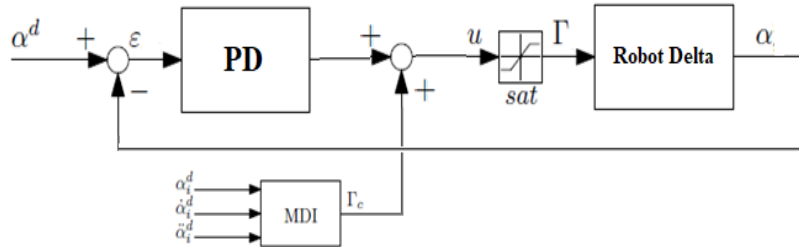


Fig.4.10.b Schéma de commande PD avec MDI

Pour la détermination des paramètres Kp et Kd , la solution la plus courante en robotique consiste à choisir les gains de façon à obtenir un pôle triple réel négatif, ce qui donne une réponse la plus rapide possible sans oscillations [13] (voir [1] [14] [13]). Dans notre cas, nous allons en premier lieu visualiser les résultats donnés par le PD déjà implémenté. Ensuite, nous incorporons le Modèle Dynamique Inverse établi en 5.5 et nous comparons les résultats obtenus.

4.3.1. Commande avec régulateur PD classique

Afin de simuler le comportement du robot avec le régulateur PD implémenté par défaut par l'EPFL, nous proposons le schéma de simulation suivant de Fig.4.10.a. Nous prélevons les coordonnées cartésiennes de la nacelle et nous les comparons avec les coordonnées de la trajectoire désirée :

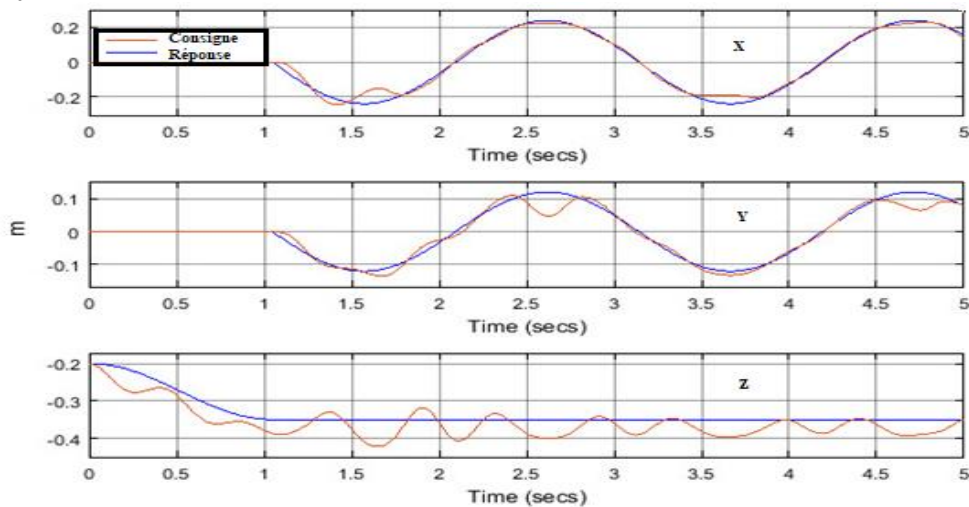


Fig.4.11 Suivi de trajectoire PD

Comme nous pouvons le remarquer, la position réelle de la nacelle n'est pas exactement celle désirée. Certes le système suit assez bien la coordonnée X, mais les résultats sont moins bons sur l'axe Y et encore pire selon Z. Ces observations sont confirmées par la figure suivante :

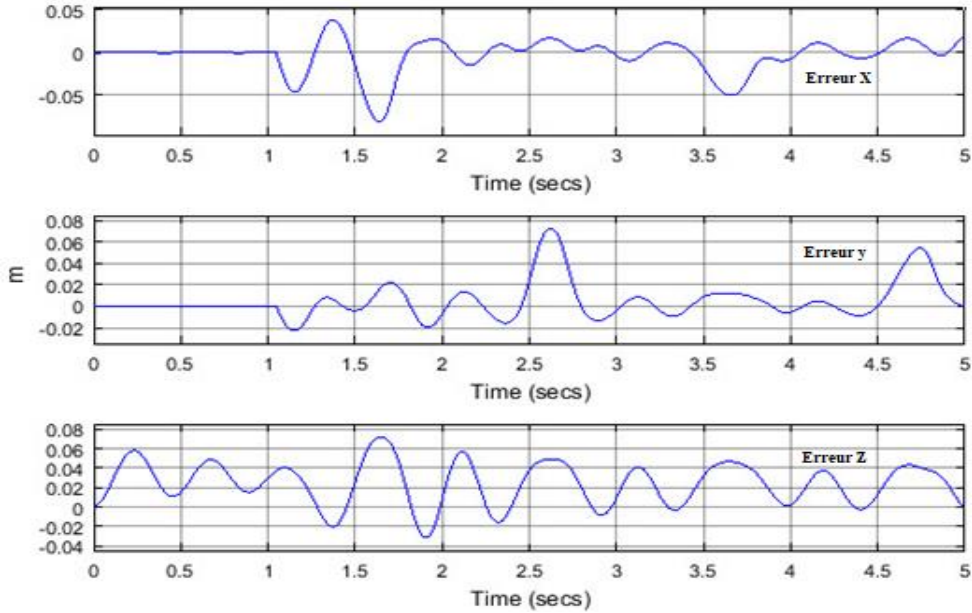


Fig.4.12 Erreur de suivi de trajectoire PD

4.3.2. Commande avec régulateur PD et MDI à priori

Maintenant, nous allons ajouter à la commande obtenue par le régulateur PID, un couple calculé à priori à partir du modèle dynamique inverse du robot. Pour s'assurer d'appliquer une commande admissible et supportée par les actionneurs du robot, nous ajoutons une condition de saturation.

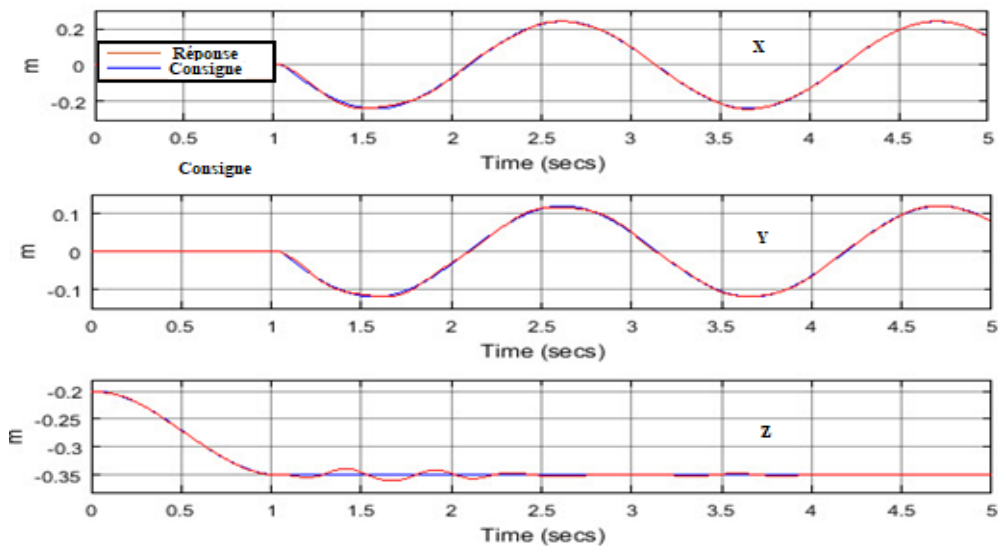


Fig.4.13 Suivi de trajectoire PD+MDI

Comme nous pouvons le remarquer, le suivi de trajectoire est respecté. Pour mieux évaluer cette commande, observons l'erreur :

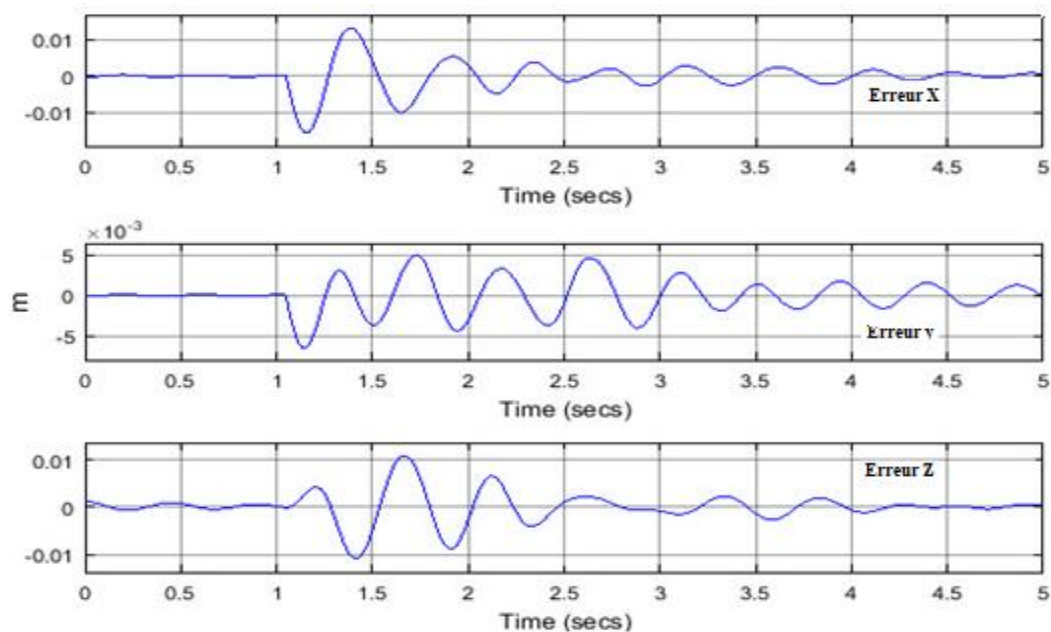


Fig.4.14 Erreur suivi de trajectoire PD+MDI

L'erreur est très faible par rapport à la commande PD précédente. Les objectifs d'asservissement en terme de suivi de la trajectoire ont donc été atteints avec cette méthode, et ceci, bien sûr, en appliquant une commande tout à fait supportée par les trois moteurs comme le montre la figure suivante :

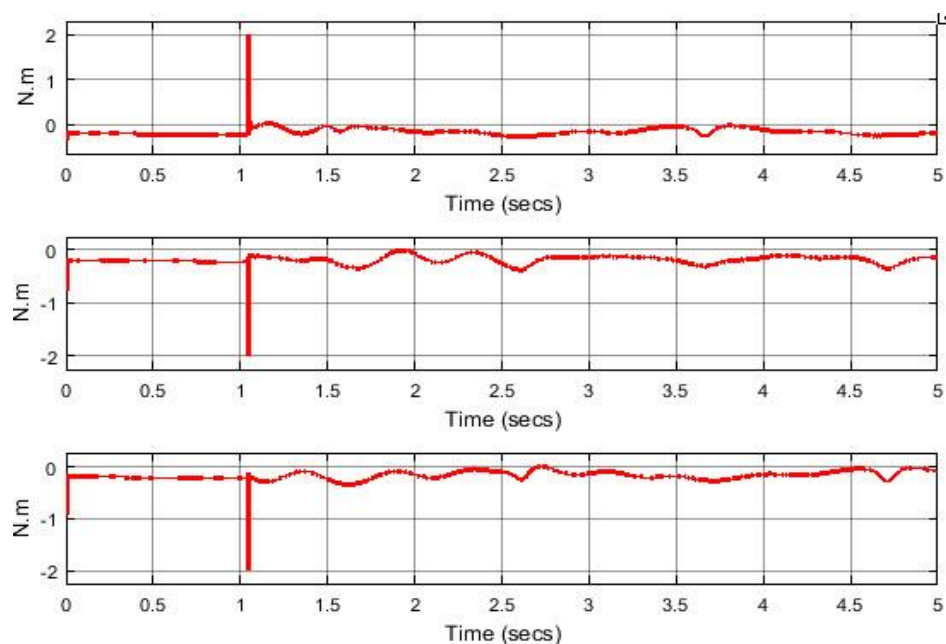


Fig.4.15 Les commandes des trois bras PD+MDI

4.3.3. Comparaison PD et PD-MDI

Comparons les erreurs obtenues par les deux méthodes de régulation :

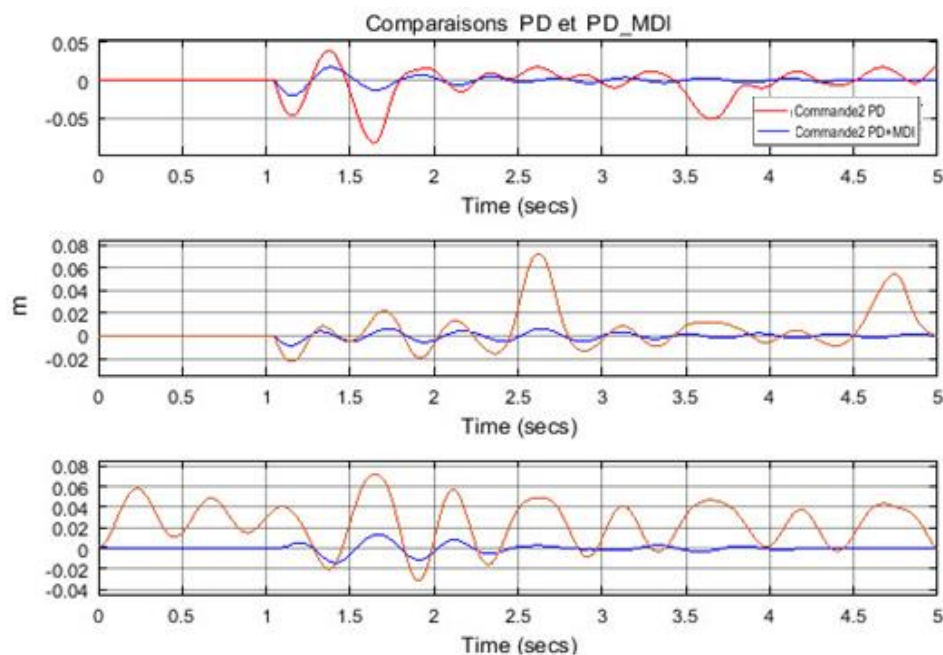


Fig.4.16 Comparaison Erreurs PD et PD+MDI

Comme indiquée sur la figure ci-dessus, l'erreur obtenue par la méthode PD+MDI (en bleu) est nettement plus faible que celle obtenue par la méthode PD (en rouge). La commande PD avec MDI est donc meilleure.

4.4. Tests de robustesse

Afin de tester la robustesse du schéma de régulation PD avec couple a priori et le comparer avec le schéma classique PD, nous injectons à l'entrée du robot la perturbation suivante :

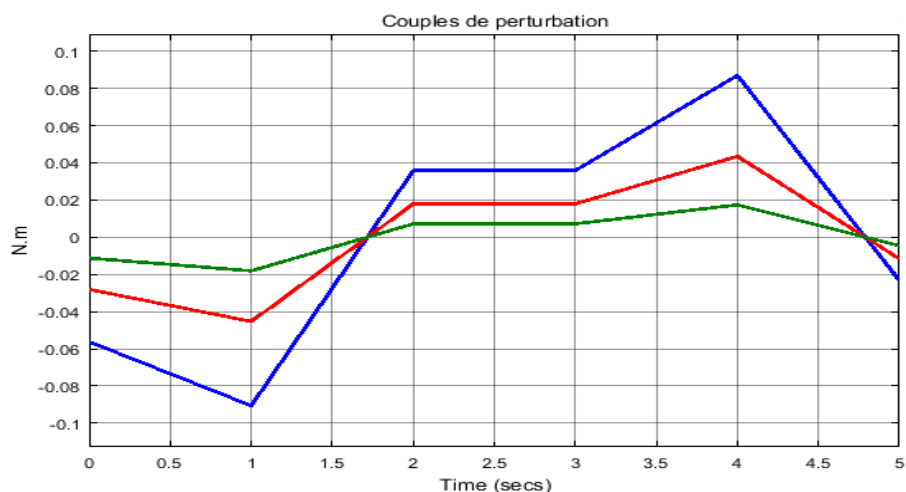


Fig.4.17 Couples de perturbations

Le suivi de trajectoire des deux schémas de régulations est donnée par :

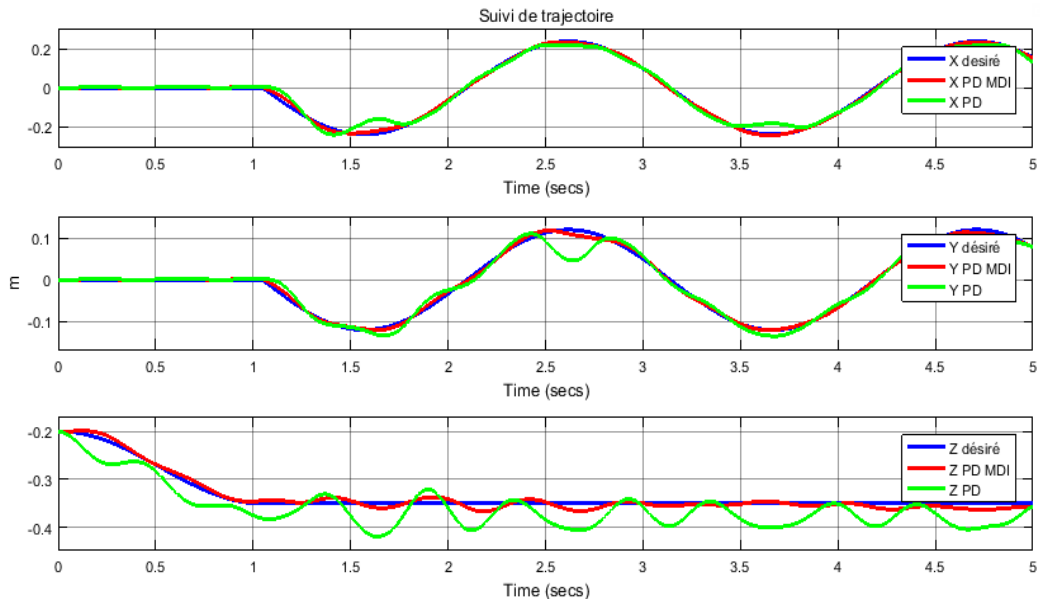


Fig.4.18 Suivi de trajectoire sous l'effet de perturbations

Même si le régulateur PD est connu pour sa robustesse, l'erreur en suivi de trajectoire est assez grande lorsque le système subit des perturbations importantes. Pour y remédier, l'incorporation du modèle dynamique inverse (couple a priori) améliore nettement la robustesse du régulateur et diminue considérablement l'erreur de suivi de trajectoire comme le montre la figure suivante :

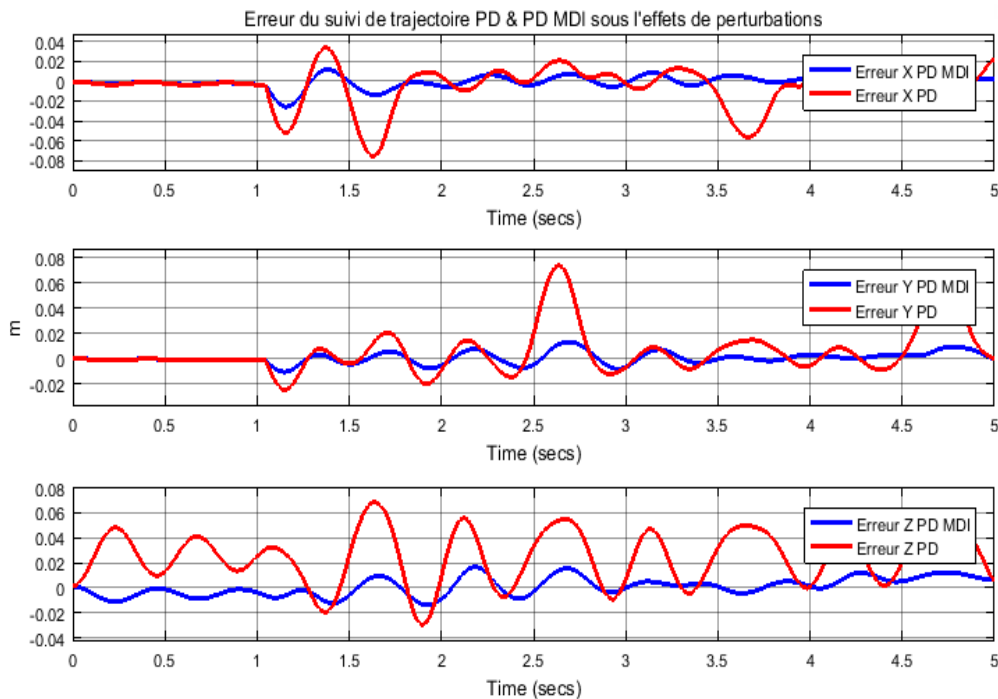


Fig.4.19 Erreur PD & PD MDI sous l'effet de perturbations

Ceci s'explique par ce qu'on appelle la compensation dynamique. En effet, le rôle d'un régulateur est de trouver le couple à appliquer au moteur pour que ce dernier réalise le mouvement désiré. Le régulateur PD étant linéaire, l'idéal serait donc de combiner ces deux outils pour le même objectif: Asservissement au mieux de la position du moteur à la trajectoire désirée. [15]

- Le couple à priori servira à s'approcher rapidement des couples nécessaires à la réalisation du mouvement.
- Le régulateur pour fermer la boucle de position.

4.5. Conclusion

Au début de ce chapitre, nous avons comparé les trois structures de commandes usuelles, à savoir la commande en boucle ouverte, la commande en boucle fermée et la commande en boucle fermée avec l'incorporation du modèle dynamique inverse. Pour un régulateur donné, les résultats de simulations montrent que la dernière structure de commande est la plus adaptée et nous expliquons ça par la compensation dynamique.

Ensuite, notre travail a porté sur l'amélioration des performances du régulateur PD implémenté par défaut sur le robot Delta ISIR88. Pour cela, nous avons décidé de tester nos programmes sur une trajectoire de prise dépose qui constitue l'opération la plus courante pour le robot Delta.

Les résultats de la simulation confirment la théorie, l'incorporation de notre modèle dynamique inverse (couple a priori) a amélioré considérablement les performances du régulateur PD en terme de suivi de trajectoire en appliquant une commande tout à fait admissible par les actionneurs du robot Delta ISIR88.

Enfin, pour tester la robustesse de notre régulateur, des couples de perturbations ont été injectés à l'entrée du robot et comme c'était prévu, l'incorporation du MDI offre plus de robustesse au régulateur PD.

CHAPITRE 5

Implémentation et Expérimentations

Chapitre 5: Implémentation et Expérimentations

5.1. Introduction

Dans ce chapitre, nous allons implémenter les commandes établies et simulées précédemment dans l'environnement de programmation FlexCom. Pour cela, il faut tout d'abord écrire les scripts de commande en langage de programmation C++. Ensuite, en suivant les étapes expliquées dans le troisième chapitre, ces commandes sont injectées aux actionneurs du robot. Les valeurs des positions articulaires et angulaires (désirées et obtenus) données par les capteurs sont visualisées et enregistrées. Enfin, après traitement, ces données sont représentées sous forme de figures, une étude comparative est faite pour établir à la fin le meilleur régulateur.

5.2. Considérations pratiques

La pratique diffère toujours de la simulation. En effet, en simulation, les positions, vitesses, accélérations et commandes sont obtenues facilement en introduisant un simple bloc de visualisation. Ce n'est pas le cas en pratique, où il faut des capteurs et tout un système d'acquisition.

Au Laboratoire de Commande des Processus (LCP), le robot delta ISIR88 est muni de trois capteurs de positions articulaires θ_m , les vitesses et accélérations sont donc calculées par dérivation numérique. Quant au passage de l'espace articulaire vers l'espace opérationnel, il se fait grâce aux modèles géométriques et cinématiques. Ces derniers sont programmés au préalable sous FlexWare.

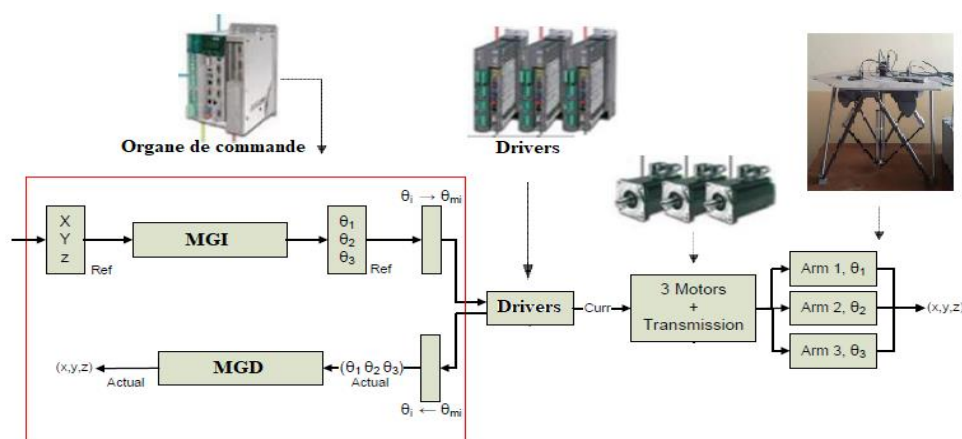


Fig.5.1 Schéma pratique général de commande [1]

Un autre point très important est la dynamique des actionneurs. Les actionneurs étant des moteurs à courant continu, leurs schémas seraient :

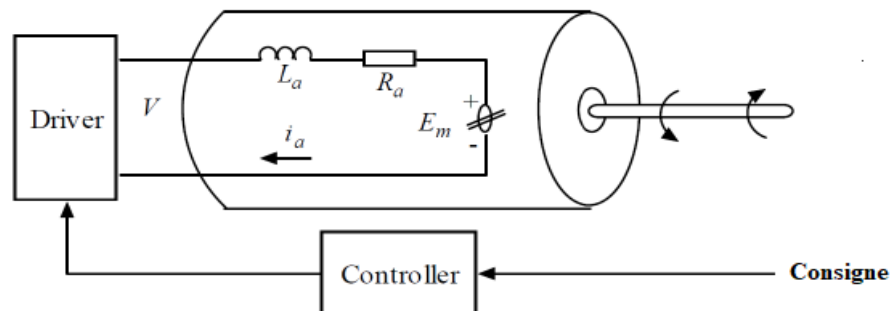


Fig.5.2 Schéma moteur à courant continu [3]

En appliquant la loi des mailles, l'équation électrique du moteur à courant continu s'écrit :

$$V = Ra i_a + La \frac{di_a}{dt} + E_m$$

Le rapport moteur bras étant :

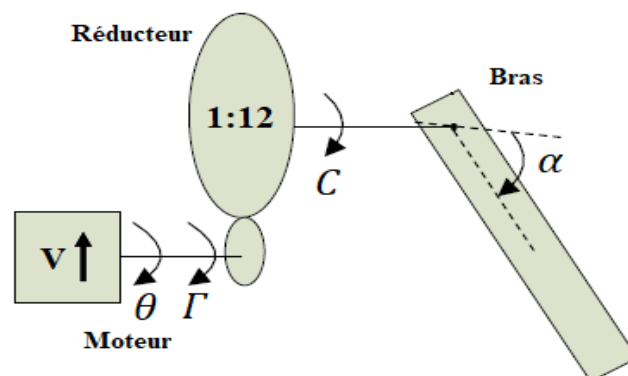


Fig.5.3 Rapport Moteur Bras [1]

L'équation électrique du moteur devient :

$$V = Ra \frac{\Gamma}{Ki} + \frac{La}{\Gamma} \frac{d\Gamma}{dt} + Ke \theta$$

Avec Ke constante de tension et Ki constante de couple (voir Fig.2.9).

Or, en pratique, la relation entre le couple et la tension a déjà été établie expérimentalement au sein du LCP. Il s'agit d'une relation linéaire $C = k.V$. Le gain k a été identifié égal à $k = 3.038 (N.m)/5$.

5.3. Environnement d'implémentation

FlexCom est un environnement de programmation des robots de type « ouvert », développé au sein du Laboratoire de Systèmes Robotiques à l'École Polytechnique Fédérale de Lausanne. Il est dit « ouvert » car il permet de modifier les fichiers et les codes pour les rendre adéquats avec les nouveaux paramètres du robot utilisé qui sont : l'architecture de la structure, les paramètres des actionneurs et des capteurs, la trajectoire désirée, les paramètres dynamiques et l'algorithme de commande personnalisé. Il offre aussi la possibilité d'effectuer les opérations en temps réel et de visualiser les différentes variables mises en jeu.



Fig.5.4 Le contrôleur muni de FlexCom encerclé

Le système de commande envoie les consignes en temps réel à chaque moteur comme la montre la figure ci-dessous :

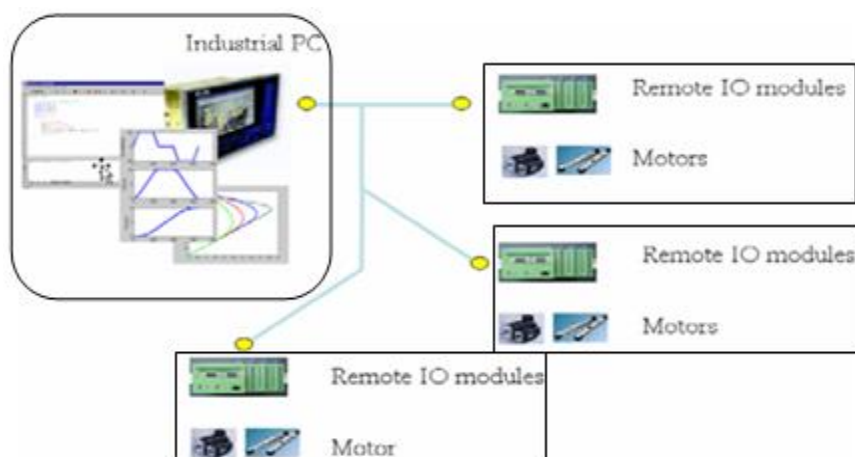


Fig.5.5 Liaison PC – Moteurs [7] [14] [15]

L'acquisition des données se fait via un capteur de position qui communique les positions angulaires des moteurs θ_i . Les vitesses et les accélérations angulaires sont obtenues par dérivation numérique. Quant à la position de la nacelle ainsi que sa dynamique, on doit passer par les modèles géométriques et cinématiques programmés au préalable sur l'éditeur de FlexCom. A noter que les protocoles de communication utilisés sont **Profibus**, **RS232** et **FireWire IEEE 1394**.

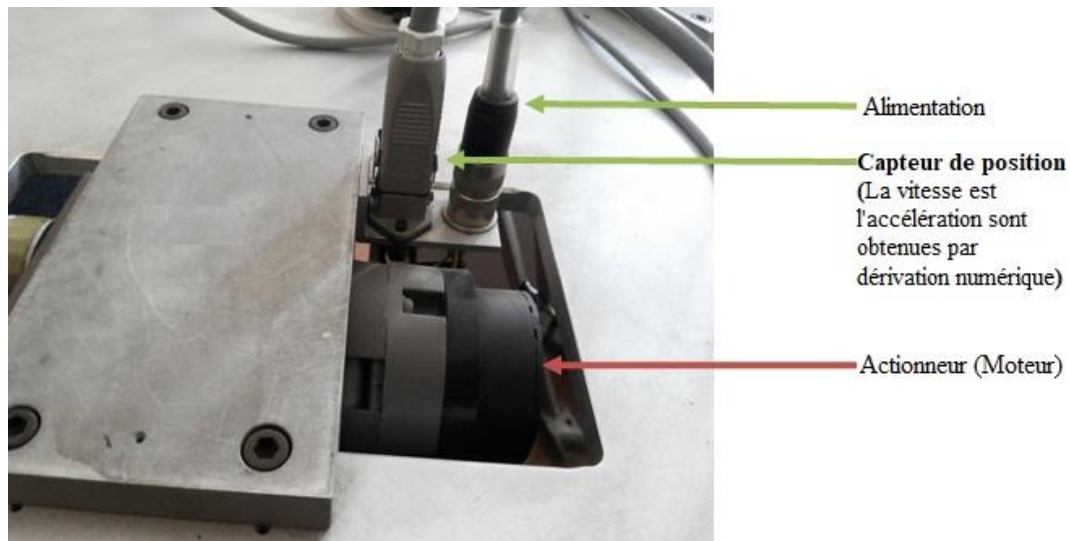


Fig.5.6 Communication avec les actionneurs : Acquisition et Commande

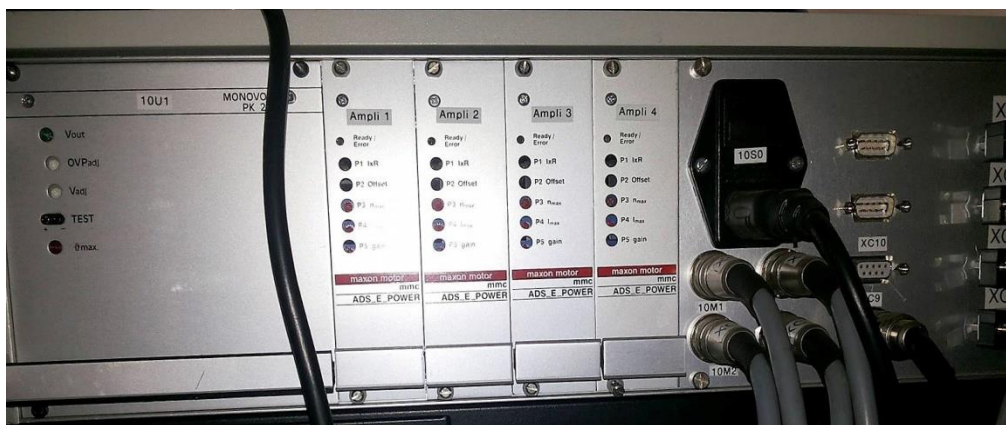


Fig.5.7 Tableau de bord de l'ordinateur de commande du robot ISIR88

Le fait de pouvoir alimenter chaque moteur séparément nous donne la possibilité d'implémenter les stratégies de commande centralisée et décentralisée. C'est-à-dire qu'on peut générer une commande commune aux trois actionneurs comme on peut appliquer à chaque actionneur une commande qui lui est spécifique.

5.4. Description des outils de programmation utilisés

L'environnement de commande FlexCom contient des outils de développements divers. Les importants sont :

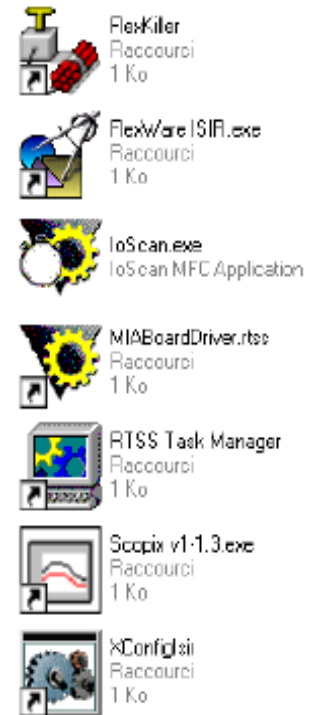
MIABoardDriver : Pilote temps réel de la carte d'axes.

FlexWare : Permet la programmation du robot. Il s'agit d'un éditeur et compilateur des codes écrits en C++.

FlexKiller : Permet de stopper tous les processus en temps réel.

RTSS Task Manager : Permet de visualiser les tâches actives à chaque instant.

Scopix : Permet de visualiser les positions, les vitesses, les accélérations, les signaux de commande, ...



Cet environnement offre toutes les fonctions auxiliaires nécessaires pour le flux de données des capteurs et des actionneurs, il reste à l'utilisateur d'introduire ses fonctions de lois de commande écrites en C++ dans des emplacements bien spécifiés. Nous allons dans ce qui suit citer les étapes nécessaires pour l'introduction d'une loi de commande :

Génération du code C++ de la loi commande :

Une loi de commande écrite sous Matlab peut être automatiquement convertie en une fonction C++, pour cela, sur Matlab la loi de commande doit être écrite sous forme d'une fonction et enregistré dans un fichier avec l'extension « .m », puis on utilise l'outil « Matlab Coder » pour la génération des fonctions et leur header en C++.

Placement des fichiers C++ de la commande :

Pour utiliser les fonctions générées de la commande, ces fichiers doivent être dans le dossier du projet qui se trouve dans le chemin suivant :

OpérationRobotISIR_\SWaresISIR\GreenCom_ISIR_CDK\RtxController_13

Introduction des fonctions de commande dans le programme :

Après l'ouverture du projet avec le fichier suivant : Opération RobotISIR_\SWaresISIR\GreenCom_ISIR_CDK

Dans le script du fichier « AlgorithmesControle.cpp » il faut ajouter les headers de la fonction de la loi de commande.

Dans ce fichier il existe une subroutine : « double ControllerUserDefined (int x_idx_, double* Param___) »

A l'intérieur de cette dernière, est appelée la fonction qui actualise les valeurs de la commande.

Donc on écrit dans ce fichier une fonction qui utilise la fonction de la loi de commande pour actualiser le vecteur « Computed_control », et on fait appel à cette dernière dans la subroutine « ControllerUserDefined ».

Enfin il faut compiler ce projet.

Mise en marche du Robot :

Après avoir programmé la loi de commande, La mise en marche du robot avec FlexCom passe par les étapes suivantes [15] [16]:

- Lancer MIABoardDriver
- Lancer FlexWare
- Initialiser le robot en utilisant le bouton ci-dessous :

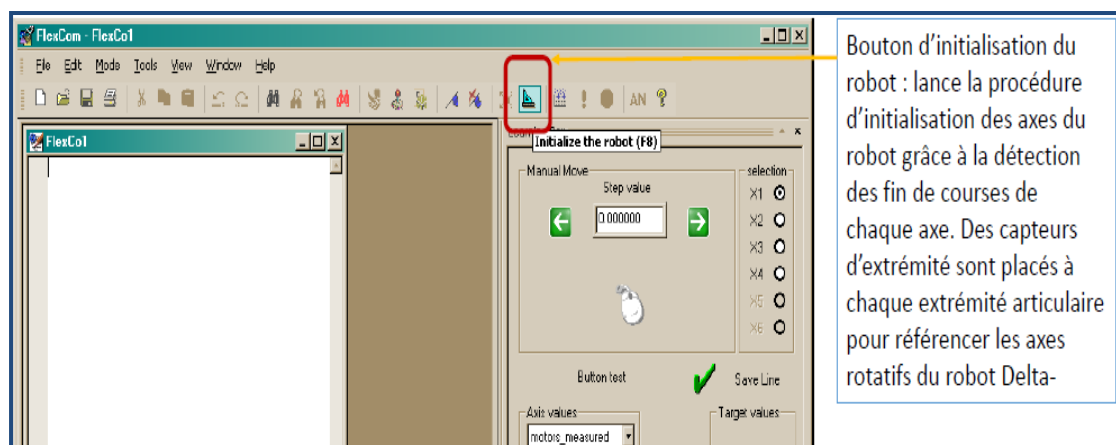


Fig.5.8 Processus d'initialisation [15]

-Après l'initialisation, c'est dans cette interface qu'on programme la trajectoire souhaitée en utilisant des fonctions prédéfinies, en effet des sous-routines déjà programmées permettent de générer des trajectoires basiques du point opérationnel selon la dynamique entrée en arguments à ces fonctions. Ces dernières seront donc combinées dans un algorithme pour générer la trajectoire désirée.

En plus de la génération de trajectoire, c'est dans cette interface que nous programmons les paramètres éventuellement utilisés dans le programme de loi de commande.

- Après écriture du programme dans l'interface, il faut vérifier le code avec le bouton «check ».

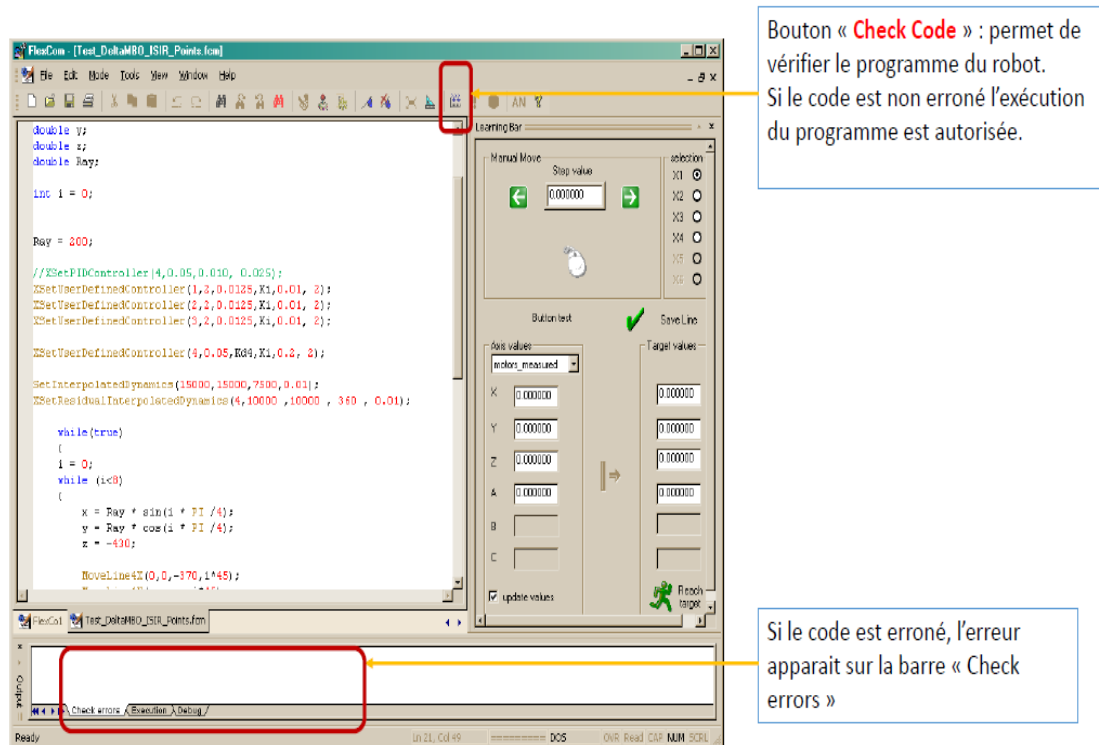


Fig.5.9 Bouton « Check Code » [15]

- Passer ensuite en mode exécution dans le menu « Mode ».
- Ouvrir « Scopix » pour pouvoir préparer la visualisation et l'acquisition des données de la manipulation.
- Finalement, Exécuter le programme.

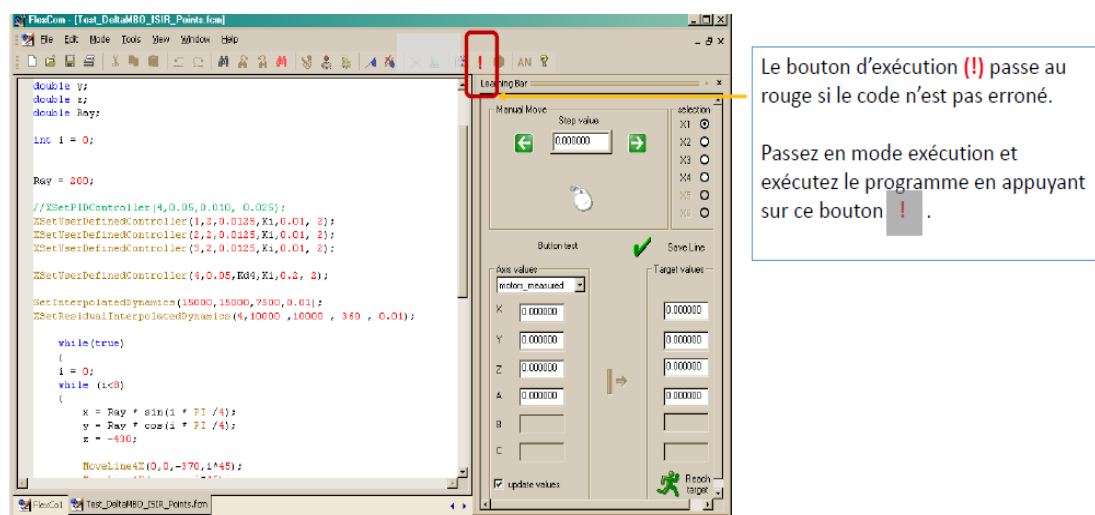


Fig.5.10 Exécution du programme [15]

Une fois que le robot se met en marche, nous pouvons visualiser les réponses avec l'outil de visualisation « **Scopix** ».

Remarques :

- A l'instant où le robot s'arrête, il faut cliquer sur « stop » sur l'outil de visualisation « **Scopix** », car quand le robot s'arrête le logiciel de visualisation continue l'acquisition des données du robot en arrêt, et écrase donc les anciennes valeurs car l'archivage des données est limité à 1500 échantillons.
- Pour des raisons de sécurité, il est indispensable de lancer la commande de commande en mode simulation avant de la tester sur le robot pour s'assurer de l'admissibilité de la commande injectée. Cela se fait en changeant le paramètre « version_debug » de 999 à 0 dans l'outil « XConfig ». L'exécution du programme et la visualisation des résultats se fait comme expliqué précédemment.

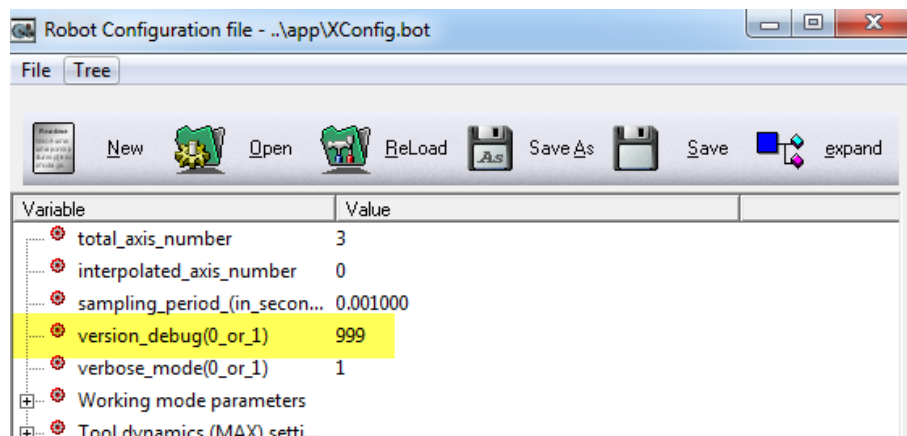


Fig.5.11 Procédure pour passer au mode simulation

5.5. Implémentation du régulateur PD et PD avec MDI

Dans ce paragraphe, nous allons implémenter les régulateurs PD et PD avec MDI simulés dans le chapitre précédent. Après l'acquisition des données, une étude comparative est établie.

5.5.1. Méthodologie

Afin de mieux visualiser l'apport de l'incorporation du modèle dynamique inverse, nous avons choisi de faire 3 essais différents. Le premier consistait à tester notre robot pour une trajectoire variée et complexe, puis, une trajectoire de prise dépose a été testée pour deux dynamiques différentes (faible et élevée).

5.5.2. Données enregistrées

L'outil Scopix nous permet d'avoir l'erreur de position en degrés et la tension de commande en Volts.

Trajectoire complexe

Considérons la trajectoire suivante :

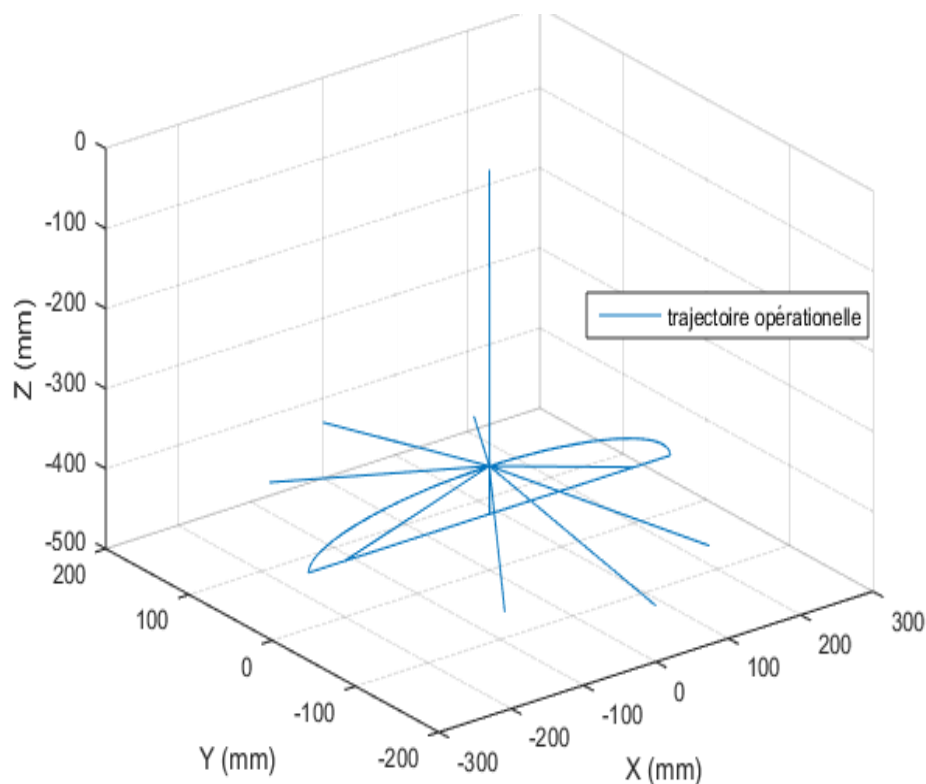


Fig.5.12 Trajectoire opérationnelle complexe

Les erreurs de suivi de trajectoire sont données par :

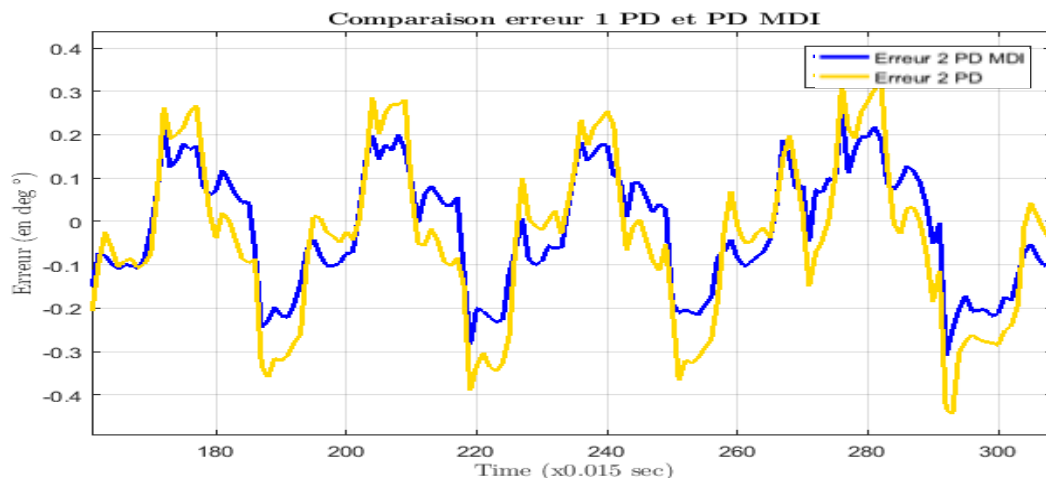


Fig.5.13.a Comparaison erreur 1 PD et PD MDI en pratique pour trajectoire complexe

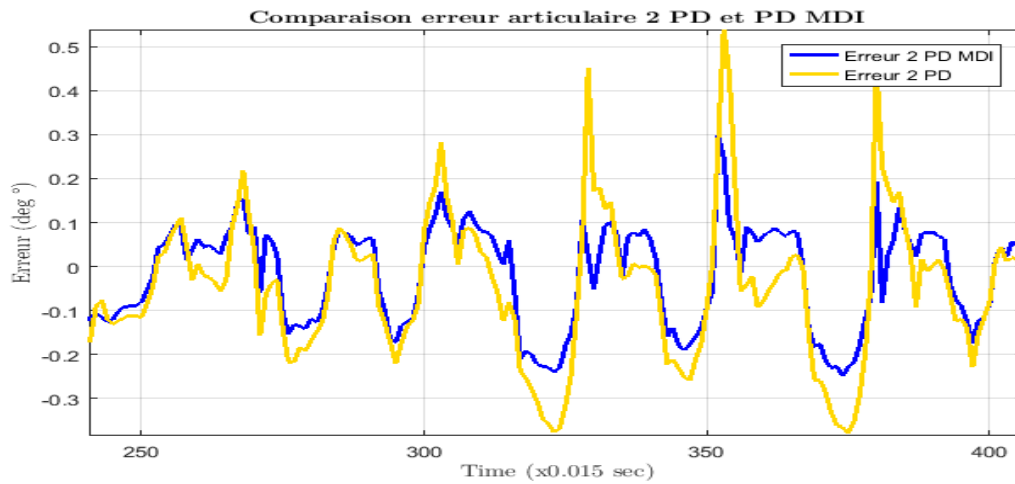


Fig.5.13.b Comparaison erreur 2 PD et PD MDI en pratique pour trajectoire complexe

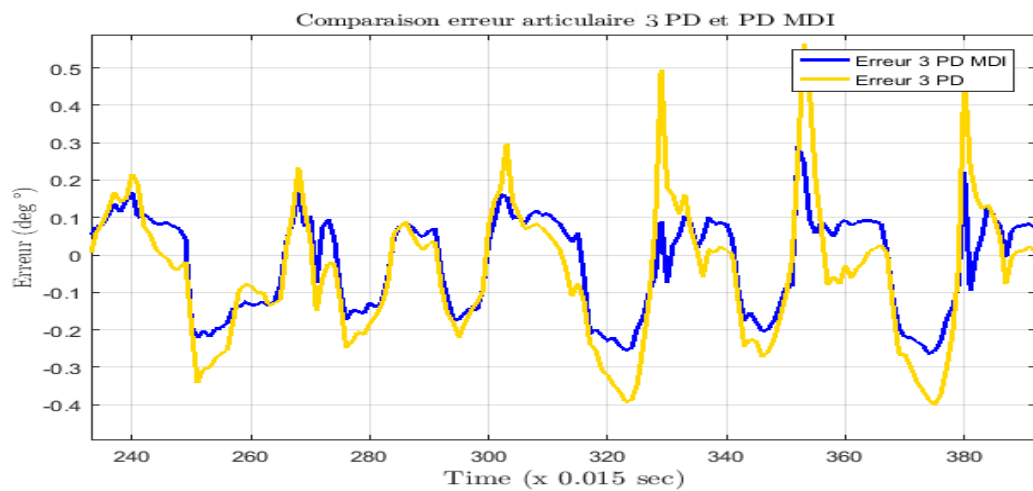


Fig.5.13.c Comparaison erreur 3 PD et PD MDI en pratique pour trajectoire complexe

Comme nous pouvons le visualiser, l'erreur en position articulaire du PD diminue considérablement avec l'incorporation du modèle dynamique inverse. Pour mieux comparer, considérons le tableau suivant :

	PD	PD MDI
Erreur Max bras 1 (en °)	2,5311	2,3567
Erreur Max bras 2 (en °)	6,7735	0,2982
Erreur Max bras 3 (en °)	6,6131	0,2909

Tab.5.1 Comparaison Erreurs PD et PD MDI pour trajectoire complexe

Il s'agit des valeurs maximales des erreurs pour chaque structure de commande. Avec le régulateur PD, l'erreur en certains points de trajectoire atteint des valeurs assez élevées. Ces valeurs de crêtes, comme le montre le tableau, sont beaucoup plus faibles lors de l'incorporation du modèle dynamique inverse.

Trajectoire Prise Dépose avec une dynamique faible

Considérons maintenant la trajectoire suivante de prise dépose avec une dynamique relativement faible :

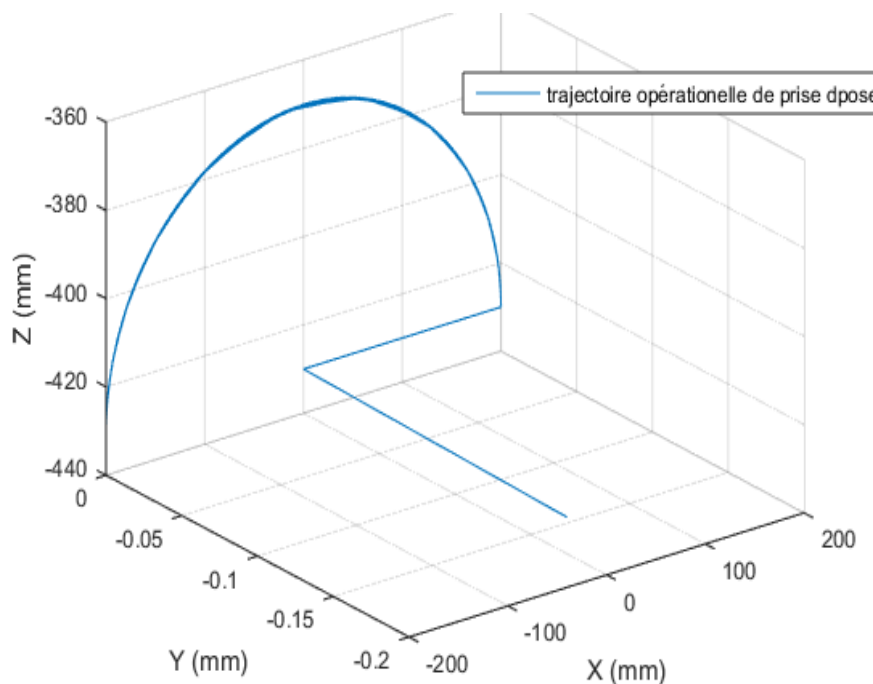


Fig.5.14 Trajectoire opérationnelle de prise dépose

L'erreur articulaire en position du premier bras avec les deux lois de commande est illustrée dans la figure 5.15.

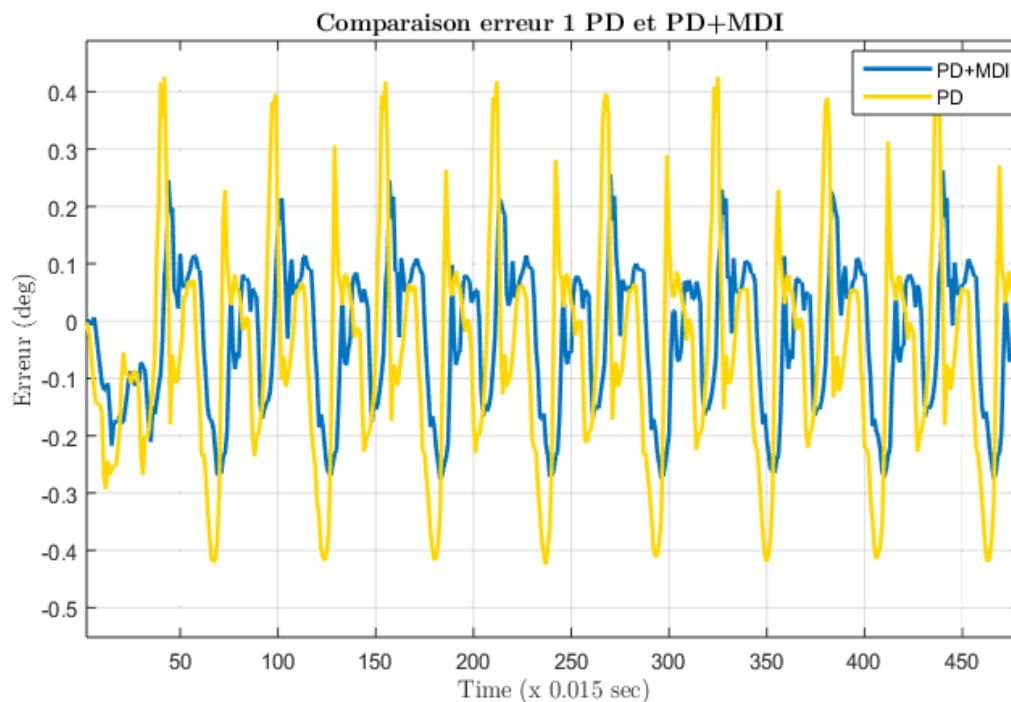


Fig.5.15 Comparaison erreur PD et PD MDI en pratique pour trajectoire prise dépose de faible dynamique

Nous remarquons bien que l'erreur en présence du MDI est nettement plus petite, en effet en calculons la moyenne du carrée de l'erreur on trouve :

$$e_{PD} = \frac{1}{T} \cdot \int e_{1_{PD}}(t)^2 dt = 0.0385$$

$$e_{PD+MDI} = \frac{1}{T} \cdot \int e_{1_{PD+MDI}}(t)^2 dt = 0.0227$$

C'est-à-dire :

$$e_{PD} = 1.7 e_{PD+MDI}$$

Ceci nous donne une évaluation quantitative de la différence entre les 2 erreurs.

Trajectoire Prise Dépose avec une dynamique élevée

Notre troisième manipulation consiste à prendre la même trajectoire prise dépose mais cette fois, avec une dynamique plus forte pour voir l'effet d'une cadence élevée sur les deux lois de commande.

La figure 5.16 montre les erreurs articulaires du premier bras pour les deux lois de commande.

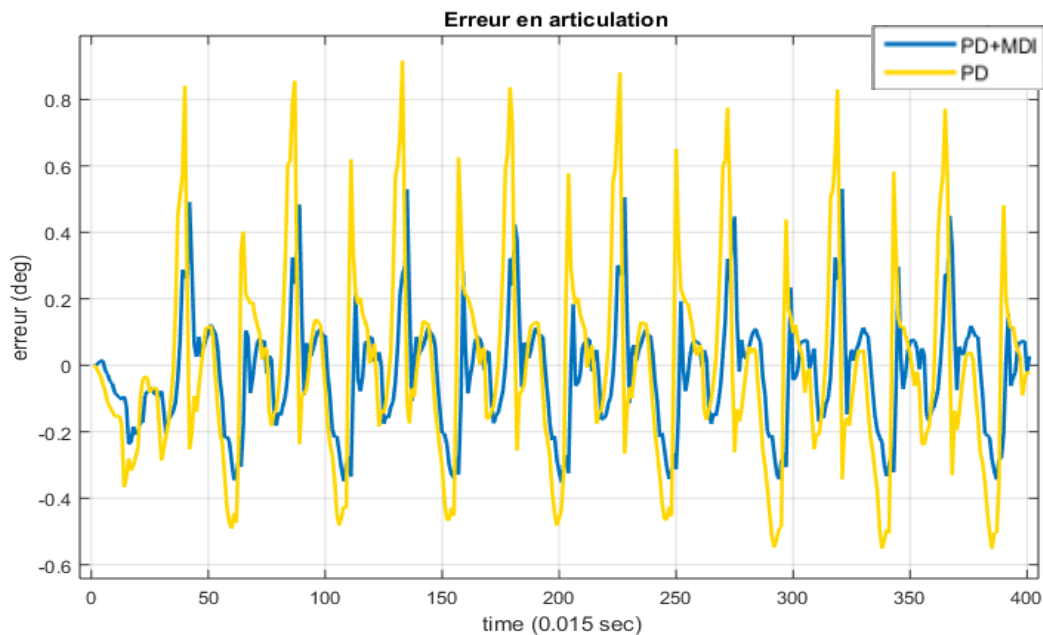


Fig.5.16 Comparaison erreur PD et PD MDI en pratique pour trajectoire prise dépose de forte dynamique

Nous constatons que la commande avec MDI est toujours plus performante que la commande par PD sans MDI. Nous allons maintenant comparer les commandes générées par les deux stratégies en les représentant sur la figure 5.17 .

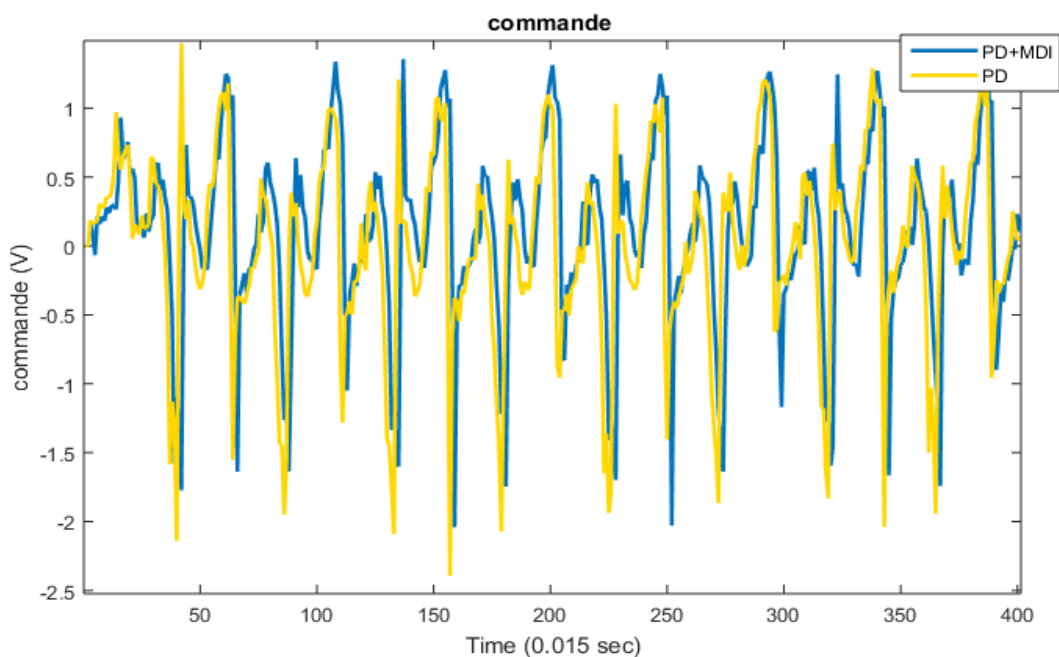


Fig.5.17 Comparaison Commande injectée PD et PD MDI en pratique pour trajectoire prise dépose de forte dynamique

Qualitativement on constate que les commandes ont la même allure.

Quantitativement, le calcul de l'énergie des deux commandes nous donne :

$$E_{PD} = \int u_{PD}(t)^2 dt = 83,1V^2$$
$$E_{PD+MDI} = \int u_{PD+MDI}(t)^2 dt = 75,2 V^2$$

Donc on en conclut qu'en plus de l'amélioration des performances, la commande avec MDI est plus économe énergétiquement.

5.5.3. Comparaison et discussion

- La comparaison entre le premier et le deuxième test montre que l'apport du modèle dynamique inverse est valable indépendamment de la trajectoire suivie.
- La comparaison entre le deuxième et le troisième test montre qu'indépendamment de la dynamique de la trajectoire (faible ou haute), le schéma avec MDI améliore toujours les résultats

5.6. Conclusion

La phase expérimentale constitue notre plus grande contribution. En effet, passer de la simulation à l'expérimentation nécessite la prise en compte de plusieurs considérations pratiques comme les limites physiques des actionneurs et la disponibilité des capteurs.

Le robot Delta ISIR88 étant doté du système de commande FlexCom, la tâche principale consistait à l'implémentation du modèle dynamique inverse en langage C++. Ces étapes ont été clairement expliquées dans ce chapitre.

Ensuite, nous avons testés les deux schémas de commande simulés dans le chapitre précédent pour plusieurs trajectoires de prise-dépose.

Les résultats obtenus sont très intéressants et le MDI implémenté diminue considérablement l'erreur de suivi de trajectoire améliorant ainsi les performances du robot. Ces résultats prouvent la faisabilité de l'implémentation du schéma de commande avec couple a priori en Hardware et confirment la théorie et la simulation.

Conclusion Générale

Notre travail a principalement porté sur deux aspects du robot parallèle Delta ISIR88 : La modélisation et la commande.

Pour ce qui est de la modélisation, nous nous sommes intéressés au modèle géométrique, cinématique et dynamique du robot. Ceci nous a permis de ressortir une particularité des robots parallèles qu'est la complexité de la modélisation. En effet, contrairement aux robots sériels où la modélisation géométrique fait appel à des méthodes systématiques (par exemple les conventions de Khalil & al), la mission s'avère plus compliquée pour les robots parallèles où il faut ressortir une relation géométrique particulière pour chaque structure. Quant à l'étude cinématique, elle n'est pas aisée non plus car elle ne ressort pas explicitement la matrice Jacobienne, un calcul numérique ou approximatif est indispensable. Pour ce qui est du modèle dynamique, le robot Delta ISIR88 étudié contient une roue de réduction de dynamique non négligeable entre le moteur et le bras, la prise en compte de ce constituant rapproche le modèle dynamique de la réalité. Une modélisation dynamique réaliste est très importante car le MDD devient un outil de simulation crédible et le MDI offre un atout indispensable pour les structures de commande avec couple a priori. à l'aide du modèle dynamique direct développé dans ce travail et du bloc SimMechanics conçu par N. I. Mokhtari and H. Nadour, nous avons pu valider la totalité des équations géométriques, cinématiques et dynamiques du robot.

Dans un second temps, nous nous sommes fixés comme objectif la synthèse d'une loi de commande tolérée par les actionneurs et respectant les objectifs d'asservissement à savoir la rapidité et la précision. Pour cela, nous avons décidé de tester nos résultats sur une trajectoire de prise dépose qui constitue l'opération la plus courante pour le robot Delta. Ensuite, pour ce qui est de la structure de régulation, notre choix s'est porté sur le schéma de commande avec couple a priori qui, selon nos simulations, donnait les meilleurs résultats.

Une fois la trajectoire de consigne générée et le schéma de régulation choisi, nous avons comparé le régulateur PD implémenté par défaut par le

Laboratoire de Systèmes Robotiques de l'École Polytechnique Fédérale de Lausanne au même régulateur PD mais avec l'incorporation du modèle dynamique inverse obtenu. Conformément à la théorie, la simulation a montré que ce dernier est plus performant en termes de précision du suivi de trajectoire.

Une partie importante de notre travail a été ensuite consacrée à la pratique. Le robot Delta ISIR88 étant doté du système de commande FlexCom, la tâche principale consistait à l'implémentation du modèle dynamique inverse en langage C++. Ensuite, les deux schémas de régulations (PD sans et avec l'incorporation du modèle dynamique inverse) ont été testés pour des trajectoires différentes de prise dépose. Les résultats obtenus sont très intéressants et le MDI implémenté diminue considérablement l'erreur de suivi de trajectoire améliorant ainsi les performances du robot. A partir des résultats pratiques, on a pu valider les résultats obtenus en simulation.

Ce travail constitue donc une base solide pour les prochains projets du LCP touchant à la robotique parallèle. Les perspectives que l'on peut envisager à moyen terme pour compléter ce travail concernent en premier lieu l'implémentation d'autres lois de commandes qui peuvent être résumées en ces deux points :

- Associer le couple a priori avec d'autres régulateurs linéaires pour améliorer leurs performances surtout lorsqu'on s'éloigne du point de fonctionnement.

- L'implémentation des lois de commande non linéaires et améliorer ainsi incessamment les performances du robot ISIR88 dans le cadre des limites offertes par ses actionneurs.

En second lieu, la notion de perception doit être prise en considération. Il s'agit de la capacité du robot à recueillir, traiter et mettre en forme des informations utiles pour agir et réagir dans l'environnement qui l'entoure. La perception est indispensable en robotique car l'environnement dans lequel le robot opère évolue et il doit avoir la capacité d'estimer de manière autonome le mouvement à effectuer.

Bibliographie

- [1] A. B. Lintott, Geometric Modelling and Accuracy enhancement of parallel manipulators, University of Canterbury: Mechanical Engineering, 2000.
- [2] J. L. Boimond, Cours de robotique, Université Angers: ISTIA.
- [3] K. Oualid et A. Larbi, Modélisation et commande d'un robot parallèle de type delta, ENP Alger, 2010.
- [4] S. Belhabib, Contribution à l'étude du robot Delta, ENP Alger, 2005.
- [5] «Site Web du Laboratoire de Systèmes Robotiques,» [En ligne]. Available: <http://lsro.epfl.ch/>.
- [6] N. I. Mokhtari et H. Nadour, Contribution à la modélisation et à la commande du robot delta ISIR88, ENP Alger, 2016.
- [7] A. Olsson, Modeling and control of a Delta-3 robot, Lund University: Department of Automatic Control, 2009.
- [8] R. Clavel, Conception d'un robot parallèle rapide à 4 degrés de liberté, EPFL Lausanne, 1991.
- [9] R. Clavel et A. Codourey, Contribution à la commande des robots rapides et précis: application au robot delta à entraînement direct, IEEE, 1991.
- [10] R. Clavel, DELTA, a fast robot with parallel geometry, 18th International Symposium on Industrial Robots, p.91-100, IFS Publications, 1988.
- [11] O. Hamdoun, F. Z. Baghli et L. Elbakkali, Inverse Kinematic Modeling of 3RRR Parallel Robot, Tetouan, Morocco: Faculty of Sciences.
- [12] X.-J. Liu, «A new approach to the design of a delta robot with a desired workspace,» *Journal of Intelligent and Robotic Systems*, Février 2004.
- [13] R. Meryem, Modélisation et Commande Robuste de Robots Parallèles: Application au Robot Delta, ENP Alger, 2016.
- [14] H. Alashqar, Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator, The Islamic University of Gaza, 2007.

- [15] M. Bouri, FlexCom A windows PC based robot controller An open architecture, Moveit Automation.
- [16] M. Bouri, Getting started with FlexCom, 2014.
- [17] S.-D. Stan, M. Manic, C. Szep et R. Balan, Performance analysis of 3 DOF Delta parallel, University of Idaho, 2011.
- [18] K. Rosquist, Modelling and Control of a parallel kinematic robot, Lund University: Department of Automatic Control 2013.
- [19] F. Paccot, N. Andreff et P. Martinet, A Review on the dynamic control of parallel kinematics: theory and experiments, Clermont-Ferrant, 2016.
- [20] V. Nabat, S. Krut, O. Company, P. Poignet et F. Pierrot, On the Design of a Fast Parallel Robot based on its Dynamic Model, Université Montpellier.
- [21] N. R. CREȚESCU, KINEMATIC AND DYNAMIC SIMULATION, University of Braşov: Bulletin of the Transilvania, 2015.
- [22] M. Bouri, Commande des robots, 2014.
- [23] C. A. Avizzano, A. Filippeschi, J. Manuel, J. Villegas et E. Ruffaldi, «An Optimal Geometric Model for Clavels Delta Robot,» chez *IEEE European Modelling Symposium*, Pisa, ITALY, 2015.
- [24] R. W. II, The Delta Parallel Robot: Kinematics Solutions, Ohio University: Mechanical Engineering, 2016.
- [25] M. Li, D. Bi et Z. Xiao, Mechanism Simulation and Experiment of 3-DOF Parallel Robot Based on MATLAB, Tianjin: Electronics and Materials Engineering Conference, 2015.
- [26] X. Lu et M. Liu, «Optimal Design and Tuning,» *International Journal of Advanced Robotic Systems*, 2016.