

République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Ecole Nationale Polytechnique



Département d'Automatique
Laboratoire de Commande des
Processus



master en Automatique intitulé :

**Modélisation et Commande du Robot Delta ISIR88,
Commandes Linéaires Basées Modèle**

Présenté par

Hatem Ibrahim KADRI

Encadré par

Pr. Djamel BOUKHETALA

M. Hakim ACHOUR

Composition du jury :

Président	M. Rachid	ILLOUL	ENP
Encadreur	M. Djamel	BOUKHETALA	ENP
Co-Encadreur	M. Hakim	ACHOUR	ENP
Examineur	M. Hachemi	CHEKIREB	ENP
Examineur	M. Omar	STIHI	ENP

ENP 2017

République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Ecole Nationale Polytechnique



Département d'Automatique
Laboratoire de Commande des
Processus



master en Automatique intitulé :

**Modélisation et Commande du Robot Delta ISIR88,
Commandes Linéaires Basées Modèle**

Présenté par

Hatem Ibrahim KADRI

Encadré par

Pr. Djamel BOUKHETALA

M. Hakim ACHOUR

Composition du jury :

Président	M. Rachid	ILLOUL	ENP
Encadreur	M. Djamel	BOUKHETALA	ENP
Co-Encadreur	M. Hakim	ACHOUR	ENP
Examineur	M. Hachemi	CHEKIREB	ENP
Examineur	M. Omar	STIHI	ENP

ENP 2017

ملخص

إن الهدف الذي ترمي إليه هذه المذكرة هو النمذجة و التحكم في الآلة المتوازية دلتا ازير88. فهي تعتبر آلة مشهورة اجتاحت في الآونة الأخيرة الساحة الصناعية نظرا لأدائها السريع و الدقيق .تبدأ الدراسة بالنمذجة الهندسية الحركية والديناميكية . قارنا بعد ذلك مختلف آليات التحكم (طريقة العامل المبرمج مسبقا و طريقة العزم المحسوب). اختبرنا مسارات مختلفة و عرضنا نتائج المحاكاة.

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 with a global modeling for the Delta robot ISIR88 which is presented and validated. Two control structures are synthesized: Gain scheduling and Computed Torque. Then, we proceeded by doing a comparison between the two methods. To ensure the utmost reliability of the work done, complex trajectories have been tested. Simulation results are illustrated using Matlab-Simulink.

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 modélisation globale du robot Delta ISIR88 qui a été faite et validée. Une fois le modèle conçu, la synthèse de la commande par Gain préprogrammé et par Couple calculé ont été synthétisés, les différentes structures de commande ont été simulés, et enfin le travail a été conclu par une étude comparative.

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

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. Hichem 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

Résumé.....	
Liste des acronymes.....	
Liste Des Figures.....	
Introduction générale.....	11
Chapitre 1: Modélisation du robot delta ISIR88.....	14
1.1. Introduction.....	14
1.2. Modélisation géométrique.....	14
1.2.1. Indices et paramètres géométriques.....	14
1.2.2. Modèle géométrique direct « MGD ».....	16
1.2.3. Modèle géométrique inverse « MGI ».....	22
1.3. Espace de travail.....	25
1.4. Modélisation cinématique.....	27
1.4.1. Modèle Cinématique Direct « MCD ».....	27
1.4.2. Modèle Cinématique Inverse « MCI ».....	30
1.5. Singularités.....	32
1.6. Modèle des accélérations direct (MAD).....	33
1.7. Modélisation dynamique.....	34
1.7.1. Modèle Dynamique Inverse « MDI ».....	42
1.7.1. Modèle Dynamique Direct « MDD ».....	44
1.8. Représentation d'état.....	44
1.9. Conclusion.....	46
Chapitre 2: Commande Du Robot Delta ISIR88.....	49
2.1. Introduction :.....	49

2.2. Trajectoire de consigne	49
2.3. Commande par la technique du couple calculé [6].....	50
2.3.1. Technique du couple calculé avec retour d'état	52
2.3.1. Technique du couple calculé avec régulateur PD.....	58
2.4. Commande par Gain Pré programmé.....	61
2.4.1. Linéarisation :	62
2.4.1. Commande par Retour d'état :	63
2.5. Etude Comparatif.....	68
2.6. Conclusion	69
Conclusion Générale	70
Bibliographie.....	72

Liste des acronymes

alpha : Position angulaire du bras

BF : Boucle fermée

BO : Boucle ouverte

Ci : 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

Liste Des Figures

Figure 1.1 Schématisation et placement des repères [1] [3].....	15
Figure 1.2 Vues de face et de profil d'une chaine « i » du robot delta [3].	15
Figure 1.3Schéma d'une chaine « i » après simplification [3].	16
Figure 1.4Modèle géométrique direct du Delta [7]	16
Figure 1.5La sphère contenant les points C_i du Delta [3].....	18
Figure 1.6Schéma de simulation SIMULINK pour la validation du MGD.....	20
Figure 1.7 Erreur entre MGD et SimMechanics pour une trajectoire linéaire....	20
Figure 1.8 Erreur entre MGD et SimMechanics pour trajectoire circulaire.....	21
Figure 1.9 Erreur entre la trajectoire désirée et celle résultante du MGD	21
Figure 1.10Modèle géométrique indirect du Delta [7]	22
Figure 1.11 Schéma Simulink de validation MGI.....	24
Figure 1.12 Erreur MGI des trois bras	25
Figure 1.13Espace de travail de chaque chaine cinématique [11].....	25
Figure 1.14Intersection des trois espaces de travail de chaque chaine [11].....	26
Figure 1.15Espace de travail du Delta	26
Figure 1.16 Espace de travail du Delta et ses paramètres [3]	27
Figure 1.17 Schéma de validation MCD.....	29
Figure 1.18 Erreur MCD	30
Figure 1.19 Schéma de validation MCI.....	31
Figure 1.20 Erreur MCI des trois bras	31
Figure 1.21 Les configurations singulières du robot Delta [9] [10]	33
Figure 1.22 Erreur en accélération entre le MAD et SimMechanics.....	34
Figure 1.23 gravité et résultante dynamique de la nacelle.....	35
Figure 1.24Force de gravité et résultantes dynamique des barres parallèles	35
Figure 1.25 Simplification de la force de gravité des barres parallèles.....	36
Figure 1.26 Simplification de la résultante dynamique de translation des barres parallèles	36
Figure 1.27Forces et résultantes dynamiques simplifiées des barres parallèles...	38
Figure 1.28Forces extérieures et résultantes dynamiques générales.....	38
Figure 1.29 Vecteurs unitaires portés sur les barres parallèles	39
Figure 1.30 décomposition des forces et des résultantes dynamiques	40
Figure 1.31 forces extérieures et résultantes dynamiques extérieures d'un bras i.	41

Figure 1.32 Schéma de structure de validation du MDI	43
Figure 1.33 Couples donnés par les deux le MDI et SimMechanics.....	43
Figure 1.34 Erreur entre notre MDI et celui de SimMechanics.	44
Figure 1.35 Réducteur mécanique	45
Figure 2.1 Trajectoire proposée de prise-dépose	50
Figure 2.2 Effet du bouclage linéarisant sur le système [7].....	52
Figure 2.3 Boucle linéarisant et stabilisation [7]	52
Figure 2.4 Schéma de commande Computed Torque avec Retour d'Etat.....	53
Figure 2.5 Suivi de trajectoire articulaire Computed Torque avec Retour d'Etat	53
Figure 2.6 Suivi de trajectoire opérationnelle Computed Torque avec Retour d'Etat.....	54
Figure 2.7 Erreurs opérationnelles Computed Torque avec Retour d'Etat	55
Figure 2.8 Erreurs articulaires Computed Torque avec Retour d'Etat	55
Figure 2.9 Couple calculé avec retour d'état (les trois couples de chaque bras)	56
Figure 2.10 Perturbation pour Computed Torque avec Retour d'Etat	57
Figure 2.11 Erreur Computed Torque et Retour d'Etat avec perturbation	57
Figure 2.12 Schéma de régulation Computed Torque + PID	58
Figure 2.13 Suivi de trajectoire Computed Torque + PID.....	59
Figure 2.14 Erreur de positionnement de la nacelle Computed Torque + PID..	60
Figure 2.15 Erreurs articulaires Computed Torque + PID.....	60
Figure 2.16 Valeurs des couples Computed Torque + PID	61
Figure 2.17 Schéma de la structure de Commande par Gain Préprogrammé.....	64
Figure 2.18 Suivi de Trajectoire Articulaire.....	65
Figure 2.19 Suivi de Trajectoire Operationnel	66
Figure 2.20 Erreur sur la Trajectoire operationnel.....	66
Figure 2.21 Erreurs sur la dynamique des articulations.....	67
Figure 2.22 Couples de la commande par gain préprogrammé.....	68

Introduction générale

Dans ce Mémoire de Master , nous nous sommes intéressés à la commande du Robot Delta disponible au sein du Laboratoire de Commande des Processus de l'Ecole Nationale Polytechnique, notre choix s'est porté sur ce projet car il englobe une partie théorique riche et une partie pratique qui constitue une première en ce qui concerne l'étude de ce robot au sein de l'école , Pour Cela nous avons établi comme objectif , La mise en œuvre des commandes en pratique pour améliorer les performances du robot en poursuite.

Cette objectif a été atteint dans les travaux de notre Projet de Fin d'étude du diplôme d'ingénieur, Donc pour ce Mémoire de Master nous avons décidé de travailler sur la simulation de nouvelles commandes afin que ce travail soit une base pour les travaux expérimentaux futures sur le Robot delta ISIR88.

Après avoir implémenté avec succès la commande PD avec incorporation du Modèle dynamique inverse, notre choix s'est porté sur les commandes linéaires basées modèle.

Il existe plusieurs méthodes pour utiliser les commandes linéaires sur ce Système, nous avons choisi de simuler deux méthodes, La Commande par Gain Préprogrammé et la technique du couple calculé, puis nous avons comparé entre ces deux techniques pour voir laquelle est la plus performante.

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

Le Premier 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 deuxième chapitre expose les deux schémas de commande qui ont été testés et comparés en simulation à l'aide du logiciel « Matlab-Simulink » pour en tirer le meilleur.

Chapitre 1

Modélisation du robot delta

ISIR88

Chapitre 1: Modélisation du robot delta ISIR88

1.1. Introduction

Dans ce chapitre, nous allons faire Modélisation 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 [1].

1.2. 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 considérer comme un seul bras situé au milieu de ces dernières.

1.2.1. 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].

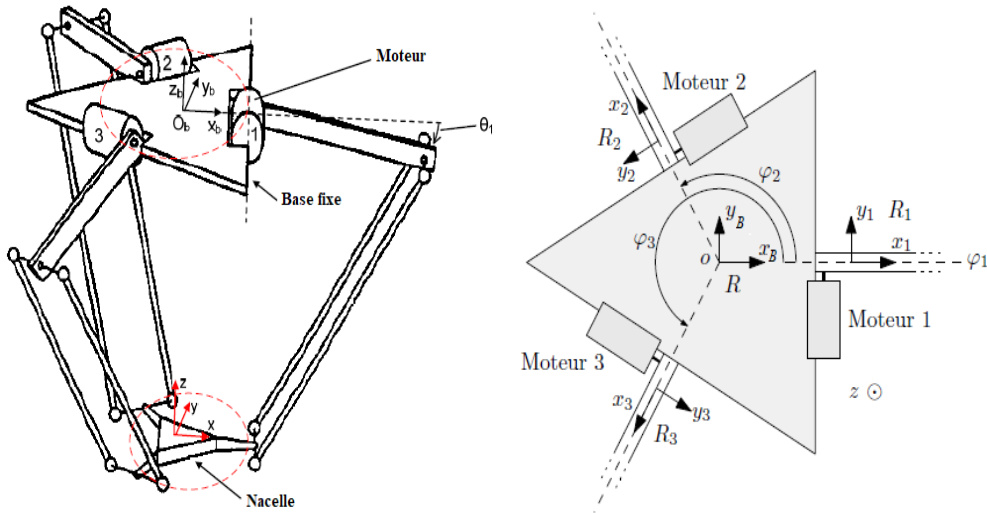


Figure 1.1 Schématisation et placement des repères [1] [3]

Les matrices de rotations R_i^T sont données par :

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

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 :

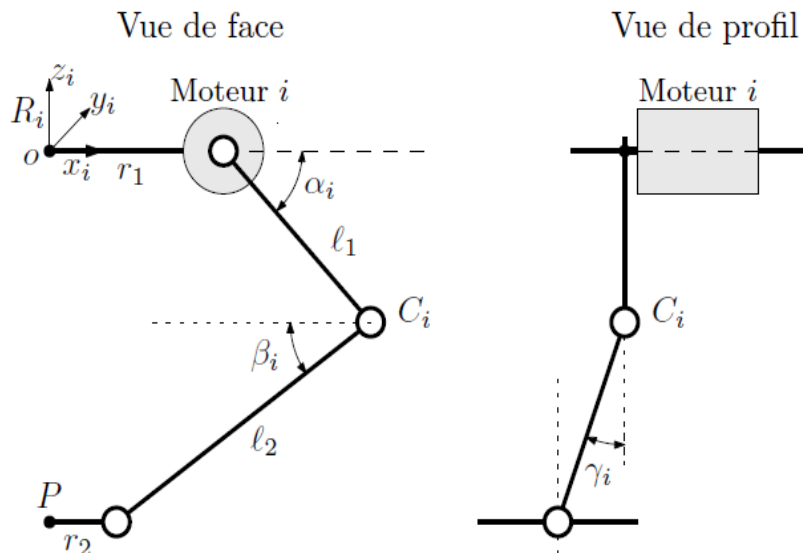


Figure 1.2 Vues de face et de profil d'une chaîne « i » du robot delta [3].

Une dernière simplification est de faire une translation horizontale de toute la chaîne de « r2 » dans la direction du point O. Ceci donne :

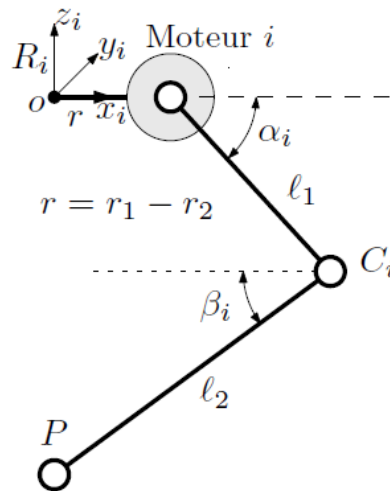


Figure 1.3 Schéma d'une chaîne « i » après simplification [3].

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).

1.2.2. 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].



Figure 1.4 Modèle géométrique direct du Delta [7]

Comme le montre la figure 1.3, les coordonnées du point Ci dans le repère Ri est donné par :

$$\overrightarrow{OC_{i/Ri}} = \begin{pmatrix} r + l1 \cos(\alpha_i) \\ 0 \\ -l1 \sin(\alpha_i) \end{pmatrix} \quad (3.2)$$

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 :

$$\overrightarrow{OC_i} = {}^R_iT \times \overrightarrow{OC_{i/Ri}} \quad (3.3)$$

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

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 quelques 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.

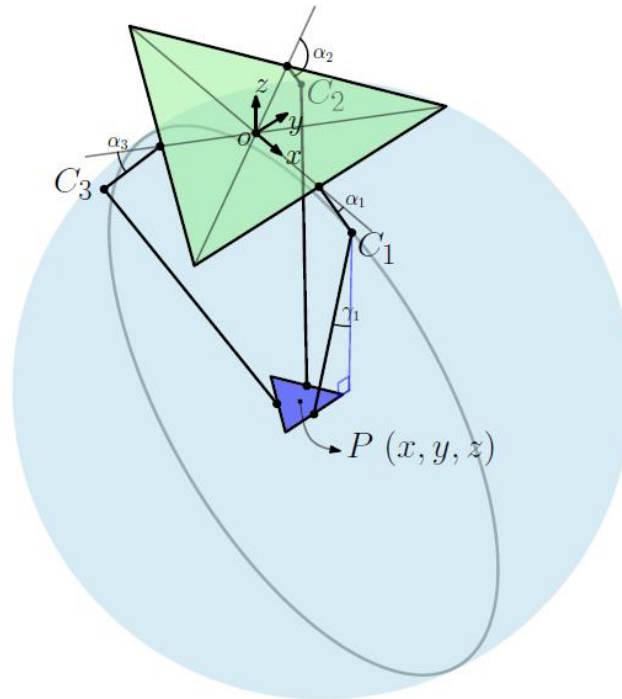


Figure 1.5 La sphère contenant les points C_i du Delta [3]

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) :

$$\begin{cases} ((r + l1 \cos(\alpha1)) \cos\varphi_1 - X)^2 + ((r + l1 \cos(\alpha1)) \sin\varphi_1 - Y)^2 + (-l1 \sin(\alpha1) - Z)^2 = l2^2 \\ ((r + l1 \cos(\alpha2)) \cos\varphi_2 - X)^2 + ((r + l1 \cos(\alpha2)) \sin\varphi_2 - Y)^2 + (-l1 \sin(\alpha2) - Z)^2 = l2^2 \\ ((r + l1 \cos(\alpha3)) \cos\varphi_3 - X)^2 + ((r + l1 \cos(\alpha3)) \sin\varphi_3 - Y)^2 + (-l1 \sin(\alpha3) - Z)^2 = l2^2 \end{cases}$$

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 :

$$D_i = -l2^2 + l1^2 + r^2 + 2 \times r \times l1 \times \cos(\alpha_i)$$

$$E_i = 2 \times (r + l1 \times \cos(\alpha_i)) \times \cos(\varphi_i)$$

$$F_i = 2 \times (r + l1 \times \cos(\alpha_i)) \times \sin(\varphi_i)$$

$$G_i = -2 \times l1 \times \sin(\alpha_i)$$

Et en notant :

$$H1 = E1G2 - E1G3 - E2G1 + E2G3 + E3G1 - E3G2$$

$$H2 = -E1F2 + E1F3 + E2F1 - E2F3 - E3F1 + E3F2$$

$$H3 = -E1D2 + E1D3 + E2D1 - E2D3 - E3D1 + E3D2$$

$$H4 = F1D2 - F1D3 - F2D1 + F2D3 + F3D1 - F3D2$$

$$H5 = -F1G2 + F1G3 + F2G1 - F2G3 - F3G1 + F3G2$$

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

$$X = Z \frac{H5}{H2} + \frac{H4}{H2}$$

$$Y = Z \frac{H1}{H2} + \frac{H3}{H2}$$

$$Z = \frac{-M \pm \sqrt{M^2 - 4LN}}{2L}$$

Avec :

$$L = \frac{H5^2 + H1^2}{H2^2} + 1$$

$$M = 2 \frac{H5H4 + H1H3}{H2^2} - \frac{H5E_1 + H1F_1}{H2} - G_1$$

$$N = \frac{H4^2 + H3^2}{H2^2} - \frac{H4E_1 + H3F_1}{H2} + D_1$$

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 [1]. 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 comme le montre la figure ci-dessous. Ces entrées sont des angles α assurant une trajectoire circulaire.

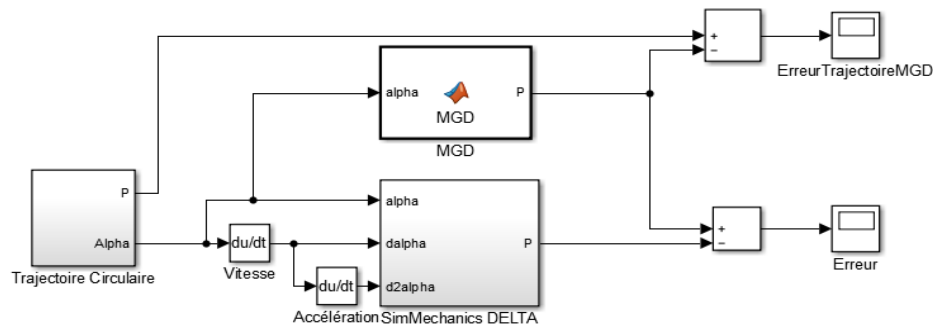


Figure 1.6 Schéma de simulation SIMULINK pour la validation du MGD

L'erreur entre le MGD et le bloc SimMechanics est représentée sur Fig.1.7.

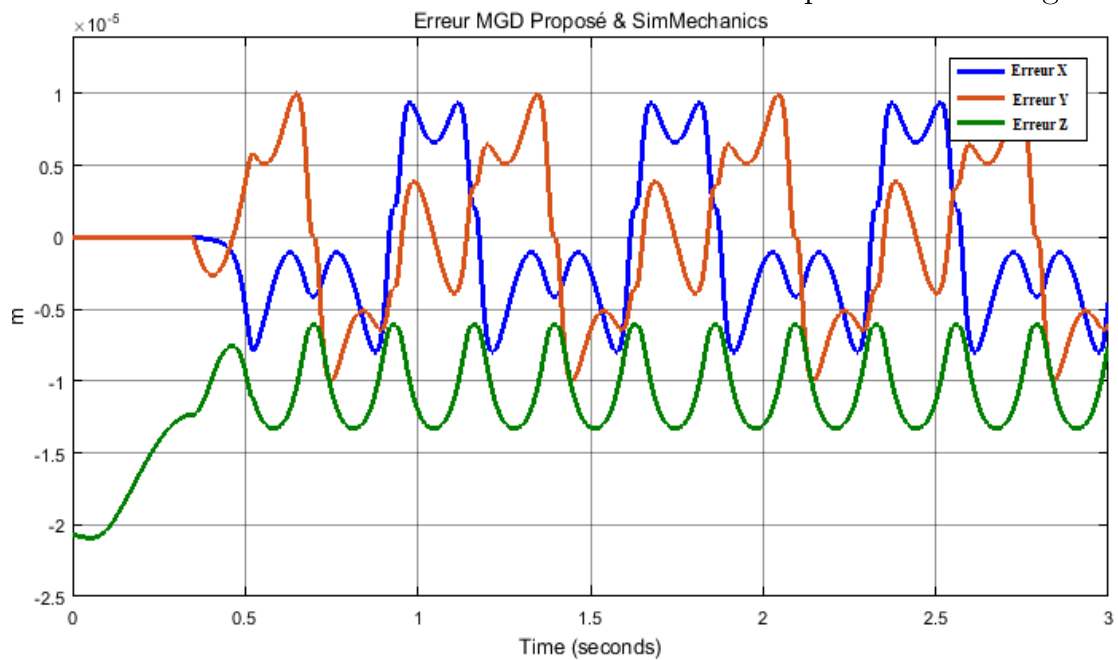


Figure 1.7 Erreur entre MGD et SimMechanics pour une trajectoire linéaire

Les erreurs sur la Fig.1.7 sont de l'ordre de 10^{-5} d'où la validation du MGD.

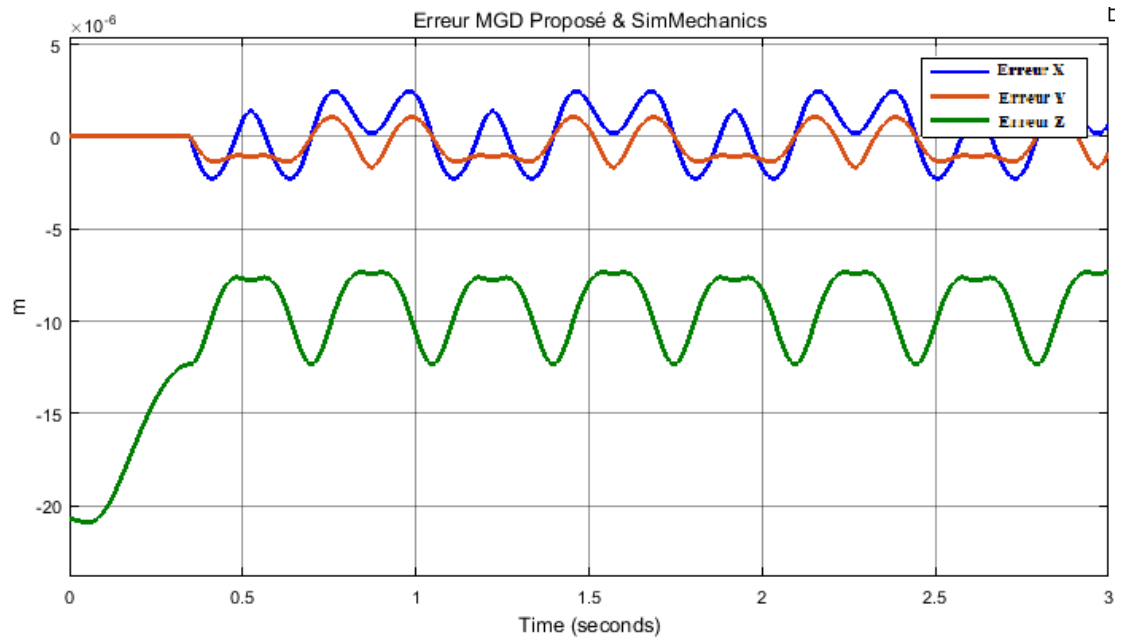


Figure 1.8 Erreur entre MGD et SimMechanics pour trajectoire circulaire

Pour la trajectoire linéaire, l'erreur est de l'ordre de 10^{-5} . Nous comparons ensuite avec une trajectoire circulaire et nous obtenons les résultats suivants :

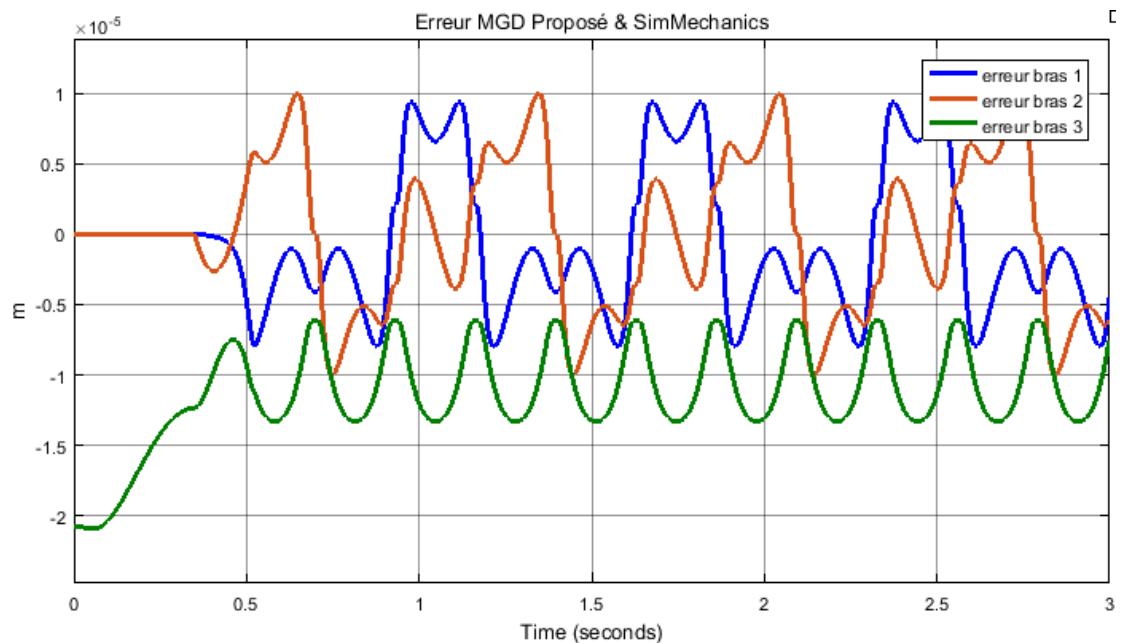


Figure 1.9 Erreur entre la trajectoire désirée et celle résultante du MGD

Toutes les erreurs trouvées sont très petites d'où la validation du MGD.

1.2.3. 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.



Figure 1.10 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 analytique de résolution est celle proposée dans [2] [3] [4]. En développant l'équation de la sphère (3.6) et en remplaçant x_i, y_i et z_i par leurs valeurs, on obtient :

$$\left((r + l1 \cos \alpha_i) \cos \varphi_i - X \right)^2 + \left((r + l1 \cos \alpha_i) \sin \varphi_i - Y \right)^2 + (-l1 \sin \alpha_i - Z)^2 = l2^2 \quad (3.9)$$

Qui, en la développant devient :

$$\begin{aligned} & (r + l1 \cos(\alpha_i))^2 \cos^2 \varphi_i + X^2 - 2 X (r + l1 \cos(\alpha_i)) \cos \varphi_i + (r + \\ & l1 \cos(\alpha_i))^2 \sin^2 \varphi_i + Y^2 - 2 X (r + l1 \cos(\alpha_i)) \sin \varphi_i + Z^2 + (l1 \sin(\alpha_i))^2 + \\ & 2 Z l1 \sin(\alpha_i) = 2^2 \end{aligned} \quad (3.10)$$

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

$$\begin{aligned} & (r + l1 \cos(\alpha_i))^2 + X^2 - 2 X (r + l1 \cos(\alpha_i)) \cos \varphi_i + Y^2 - 2 Y (r + \\ & l1 \cos(\alpha_i)) \sin \varphi_i + Z^2 + (l1 \sin(\alpha_i))^2 + 2 Z l1 \sin(\alpha_i) = l2^2 \end{aligned} \quad (3.11)$$

On

sépare les inconnus d'un coté 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 :

$$\frac{\alpha_i}{2} = \text{atan}\left(\frac{-2Bi \pm \sqrt{4Bi^2 + 4(Ai + Ci)(Ai - Ci)}}{-2(Ai + Ci)}\right)$$

Qui donne finalement :

$$\alpha_i = 2 \text{atan}\left(\frac{2Bi \pm \sqrt{4Bi^2 + 4(Ai + Ci)(Ai - Ci)}}{2(Ai + Ci)}\right)$$

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 :

$$\alpha_i = 2 \text{atan}\left(\frac{2Bi - \sqrt{4Bi^2 + 4(Ai + Ci)(Ai - Ci)}}{2(Ai - Ci)}\right) \quad (3.15)$$

Pour la validation du MGI, nous proposons le schéma de simulation suivant :

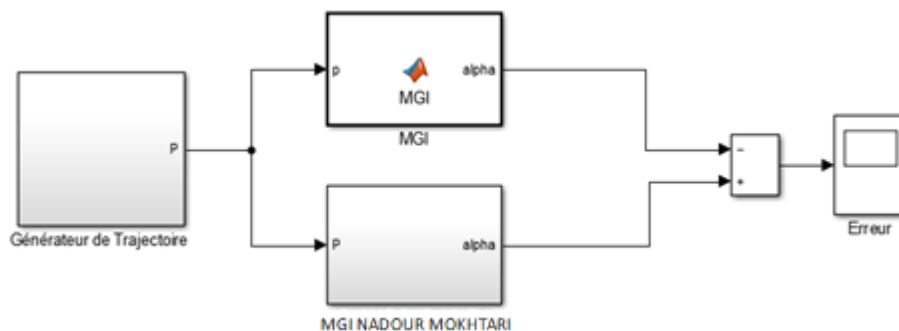


Figure 1.11 Schéma Simulink de validation MGI

Il s'agit 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 les résultats suivants :

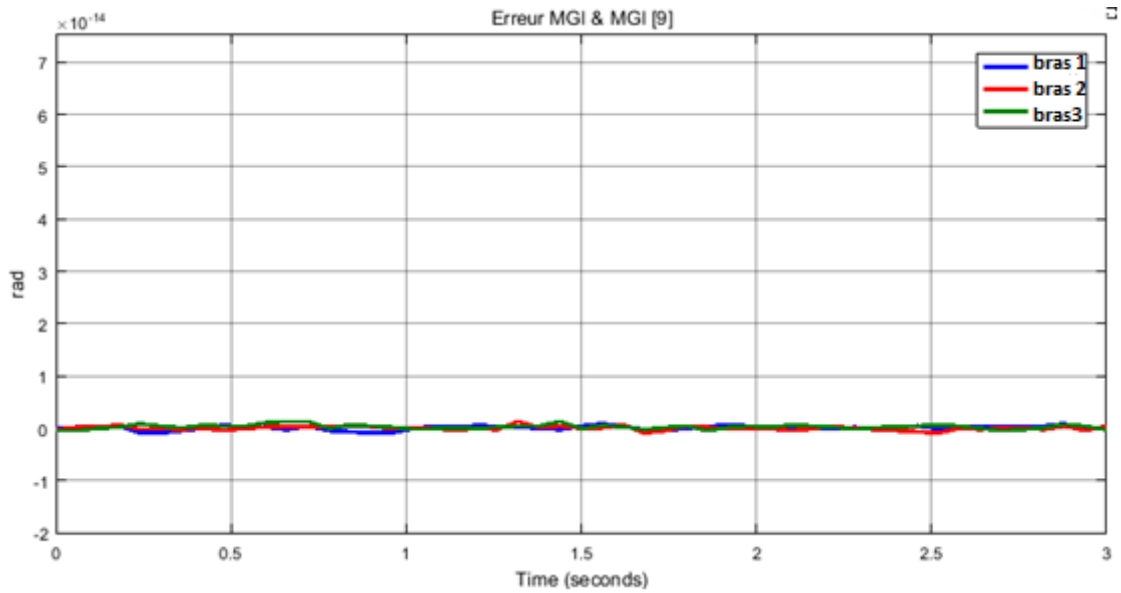


Figure 1.12 Erreur MGI des trois bras

L'erreur obtenue est très faible pour une trajectoire variée (linéaire et circulaire). Ceci nous permet de valider le modèle géométrique Indirect (MGI).

1.3. 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 :

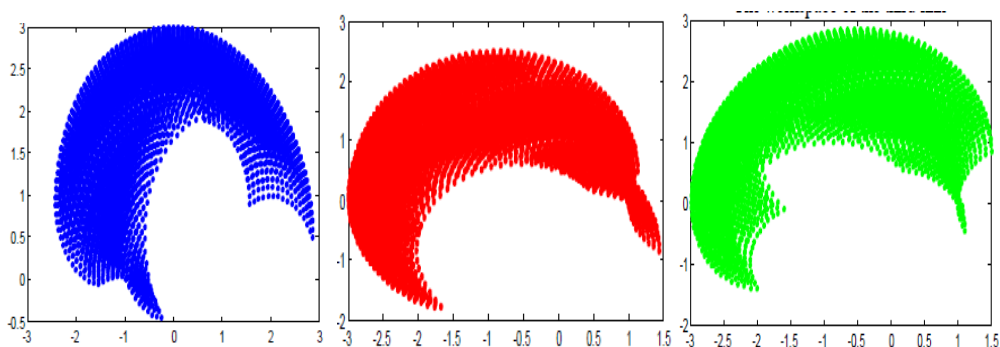


Figure 1.13 Espace de travail de chaque chaîne cinématique [11]

L'intersection donne :

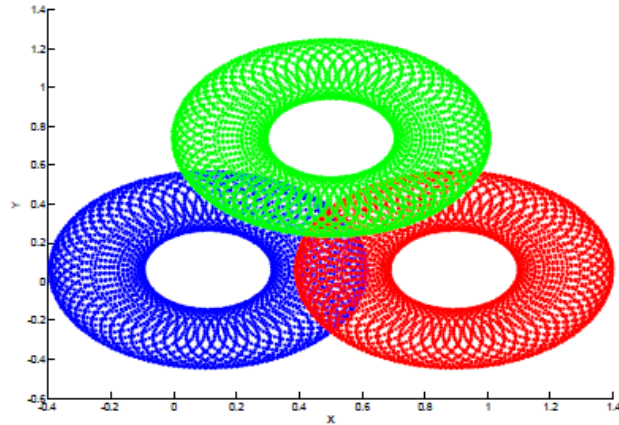


Figure 1.14 Intersection des trois espaces de travail de chaque chaîne [11]

Cette intersection est donnée dans [5] par :

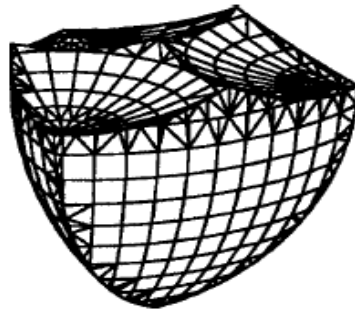


Figure 1.15 Espace de travail du Delta

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 $\gamma_{\text{Max}} = 40^\circ$.
- 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] :

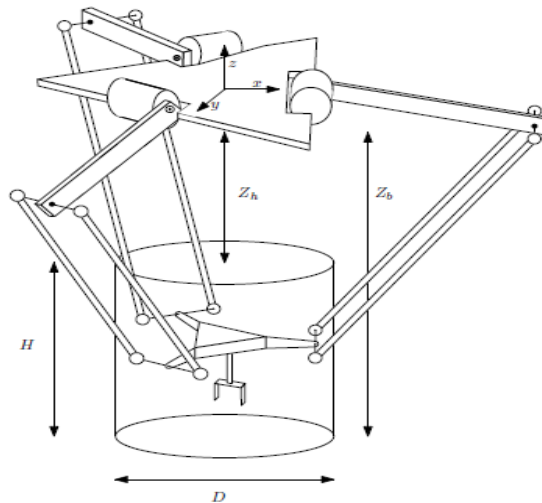


Figure 1.16 Espace de travail du Delta et ses paramètres [3]

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

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

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

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

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.

1.4. 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.

1.4.1. 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) :

$$X = Z \frac{H5}{H2} + \frac{H4}{H2} = F1$$

$$Y = Z \frac{H1}{H2} + \frac{H3}{H2} = F2$$

$$Z = \frac{-M \pm \sqrt{M^2 - 4LN}}{2L} = F3$$

Ces équations relient X Y et Z avec des paramètres Hi, 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 ;

$$\dot{X} = \frac{dF1}{dt} = \frac{dF1}{d\alpha} \frac{d\alpha}{dt} = \frac{dF1}{d\alpha} \dot{\alpha}$$

$$\dot{Y} = \frac{dF2}{dt} = \frac{dF2}{d\alpha} \frac{d\alpha}{dt} = \frac{dF2}{d\alpha} \dot{\alpha}$$

$$\dot{Z} = \frac{dF3}{dt} = \frac{dF3}{d\alpha} \frac{d\alpha}{dt} = \frac{dF3}{d\alpha} \dot{\alpha}$$

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 3×3 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 implémentons (3.11) et nous proposons le schéma de simulation suivant :

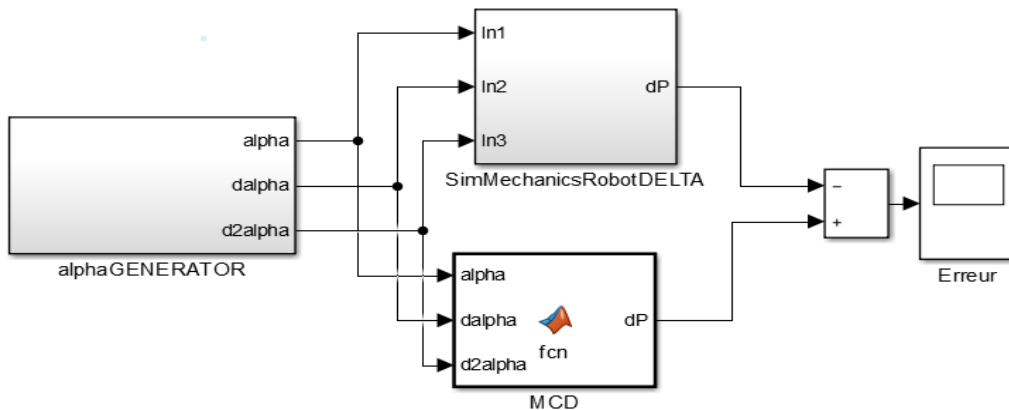


Figure 1.17 Schéma de validation MCD

Ce schéma nous permet de comparer notre Modèle Cinématique Direct avec le bloc SimMechanics représentatif du robot delta.

Nous leurs injectons les mêmes angles α et nous observons l'erreur de résultats entre les deux blocs. Celle-ci est représentée sur la figure suivante :

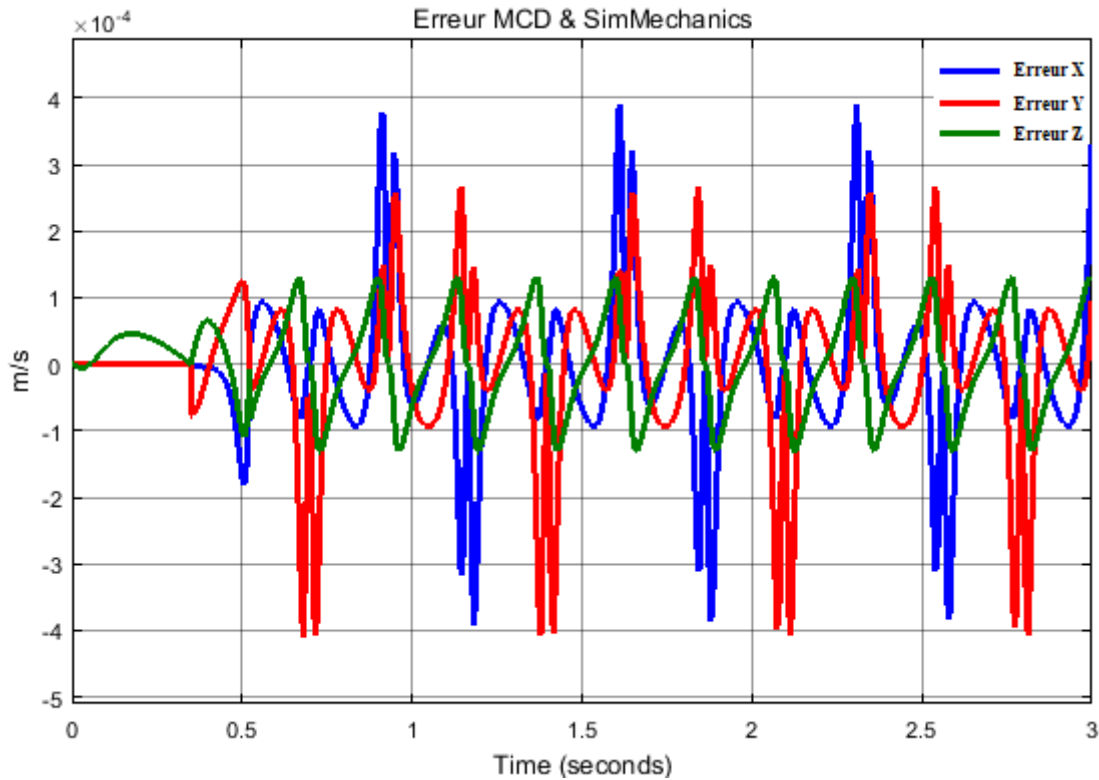


Figure 1.18 Erreur MCD

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.

1.4.2. 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'inversibilité 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)

Afin de la valider, nous proposons le schéma de simulation suivant :

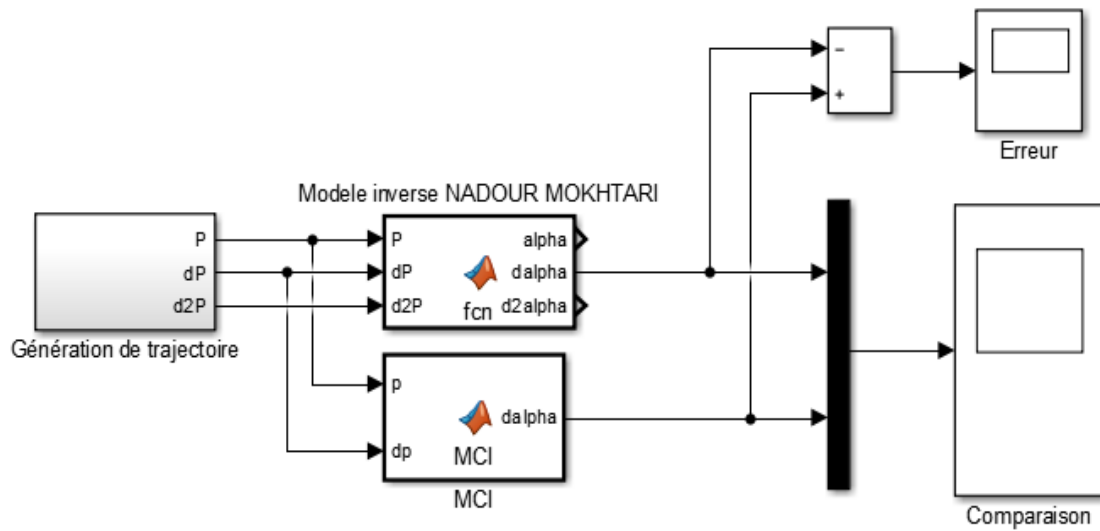


Figure 1.19 Schéma de validation MCI.

Et nous visualisons les résultats suivants :

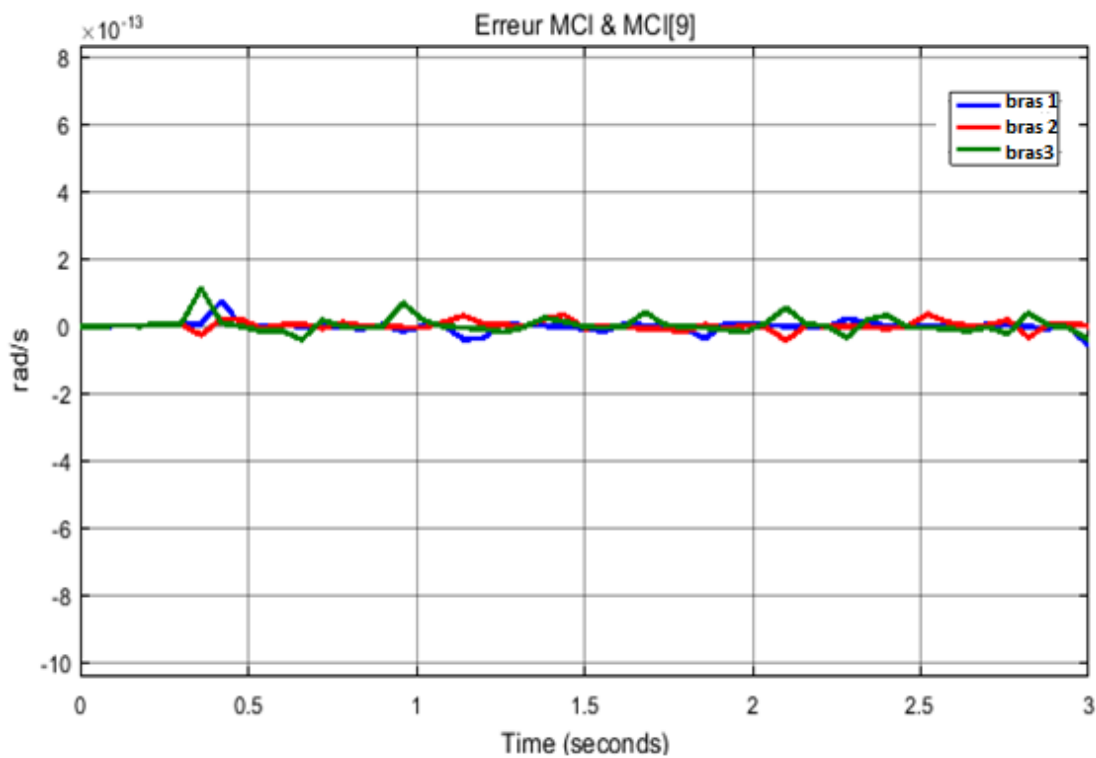


Figure 1.20 Erreur MCI des trois bras

L'erreur de modélisation cinématique inverse est très faible d'où la validation de notre MCI.

1.5. 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.1.21.a).
2. Les trois parallélogrammes sont sur le même plan horizontal ou sur trois plans parallèles (Fig.1.21.b).
3. Deux des trois parallélogrammes ont leurs barres sur le même plan ou sur deux plans parallèles (Fig.1.21.c).
4. Deux des trois parallélogrammes ont leurs barres parallèles (Fig.1.21.d).

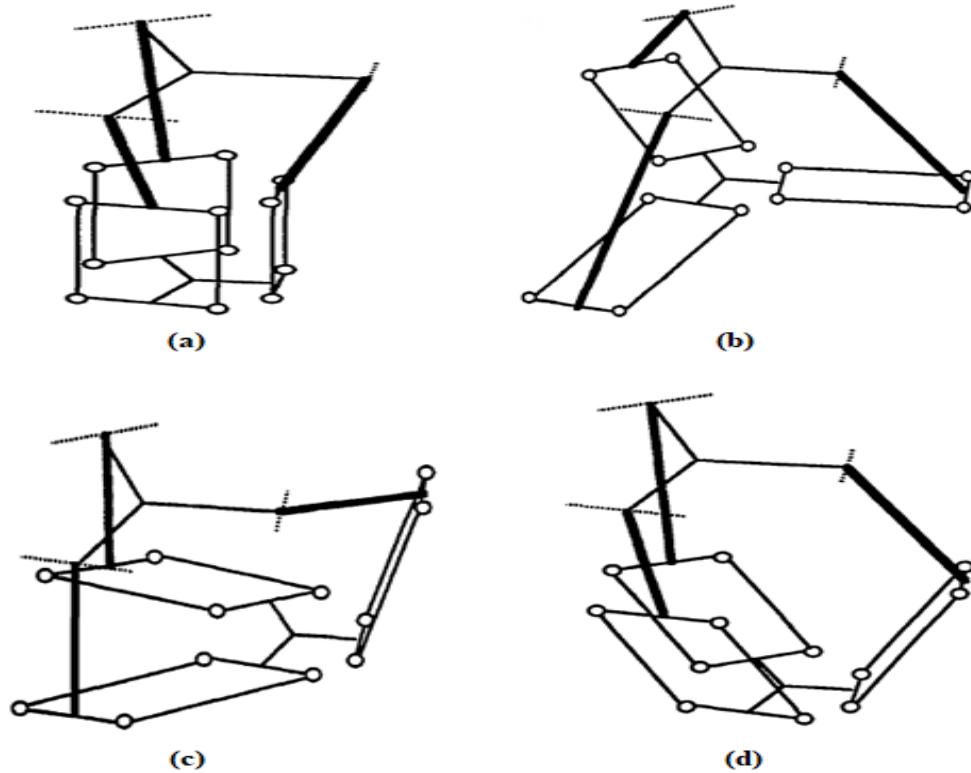


Figure 1.21 Les configurations singulières du robot Delta [9] [10]

1.6. 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 :

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

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.

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

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

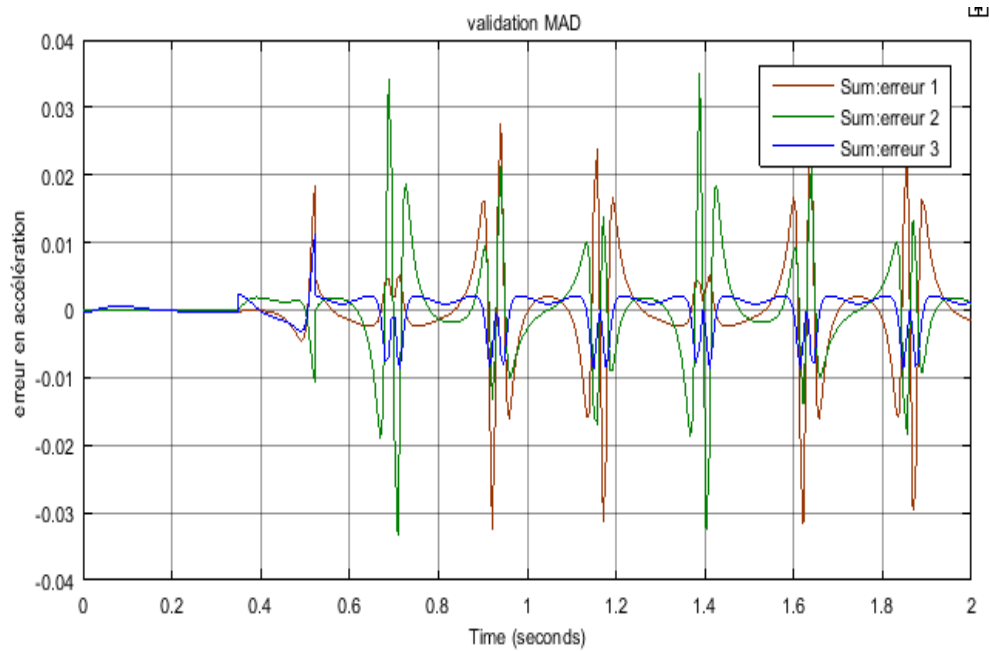


Figure 1.22 Erreur en accélération entre le MAD et SimMechanics

On constate que l'erreur est vraiment petite, l'erreur relative est de .16%, ce qui valide notre modèle.

1.7. 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 :

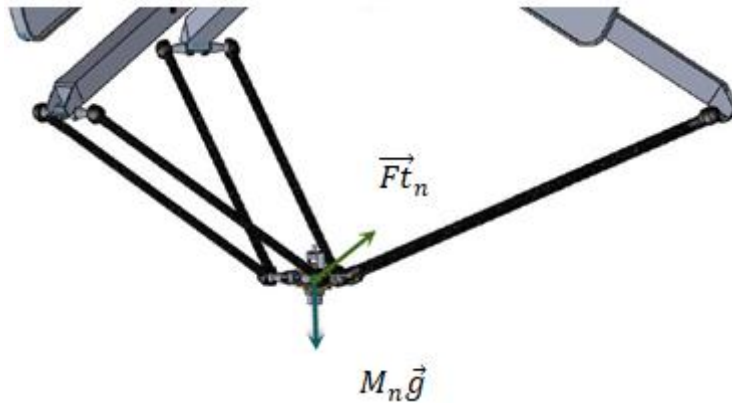


Figure 1.23 gravité et résultante dynamique de la nacelle

- 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 :

$$\vec{Ft}_n = M\ddot{\vec{O}}\vec{P}$$

Barres parallèles :

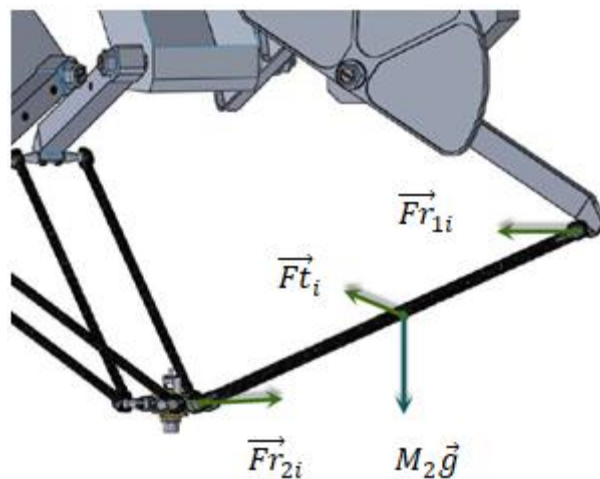


Figure 1.24 Force de gravité et résultantes dynamique des barres parallèles

- 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.1.25 :

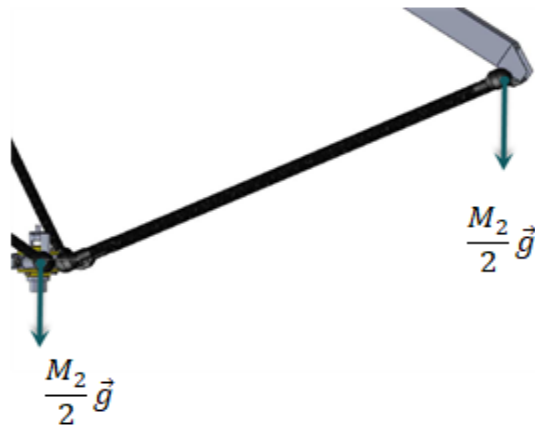


Figure 1.25 Simplification de la force de gravité des barres parallèles

- 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 torseur dynamique de ces barres en la somme d'un torseur glisseur et d'un torseur couple. Le torseur glisseur a comme résultante :

$$\vec{F}t_i = M_2 \ddot{\vec{O}G_1}$$

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

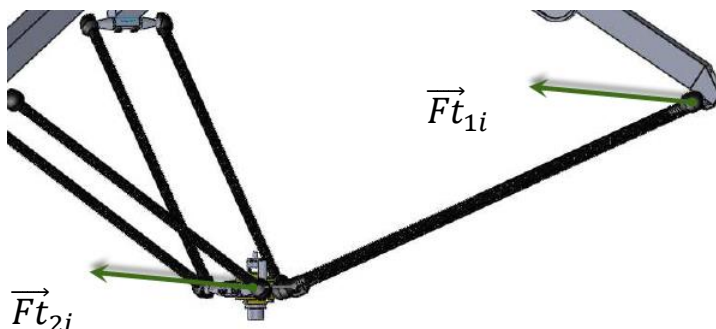


Figure 1.26 Simplification de la résultante dynamique de translation des barres parallèles

On a :

$$\vec{F}_{t_{1i}} = \vec{F}_{t_{2i}} = \frac{\vec{F}_{t_i}}{2} = \frac{M_2}{2} \ddot{\vec{O}G_1}$$

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 $\overline{PC_1}$. On a :

$$\vec{F}_{r_{1i}} + \vec{F}_{r_{2i}} = 0$$

$$\vec{F}_{r_{1i}} = F_{r_i}$$

$$\overline{G_1C_1} \wedge \vec{F}_{r_{1i}} + \overline{G_1P} \wedge \vec{F}_{r_{2i}} = \overline{M(G_1)}$$

$$\Rightarrow F_{r_i} \cdot l_2 = J_2 \ddot{\theta}$$

$$F_{r_i} = \frac{J_2 \ddot{\theta}}{l_2}$$

Et :

$$\overline{G_1C_1} = \frac{l_2}{2} \ddot{\theta} \frac{\vec{F}_{r_{1i}}}{F_{r_i}}$$

$$\Rightarrow \vec{F}_{r_{1i}} = 2 \frac{J_2}{l_2^2} \overline{G_1C_1}$$

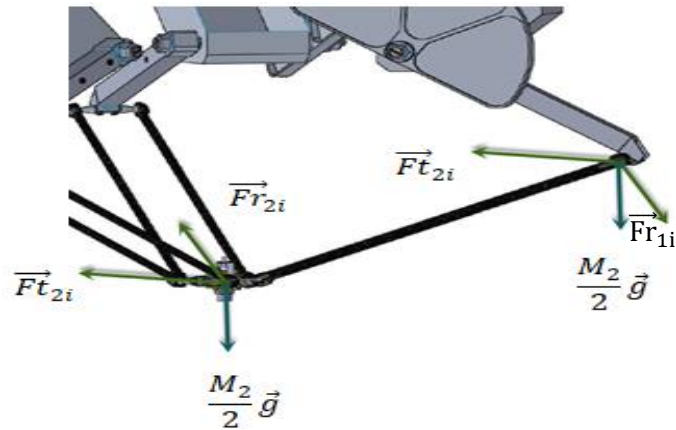


Figure 1.27 Forces et résultantes dynamiques simplifiées des barres parallèles

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

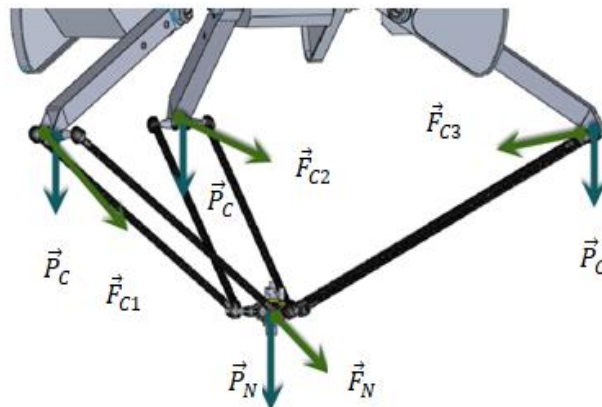


Figure 1.28 Forces extérieures et résultantes dynamiques générales

Avec :

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

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

$$\vec{F}_{C_i} = \frac{M_2}{2} \ddot{\vec{O}G_1} + 2 \frac{J_2}{l_2^2} \ddot{\vec{G}_1 C_1}$$

$$\vec{F}_N = M_n \ddot{\vec{O}P} + \sum_{i=1}^3 \left(-2 \frac{J_2}{l_2^2} \ddot{\vec{G}_1 C_1} + \frac{M_2}{2} \ddot{\vec{O}G_1} \right)$$

Forces et résultantes dynamiques aux points C_i :

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.

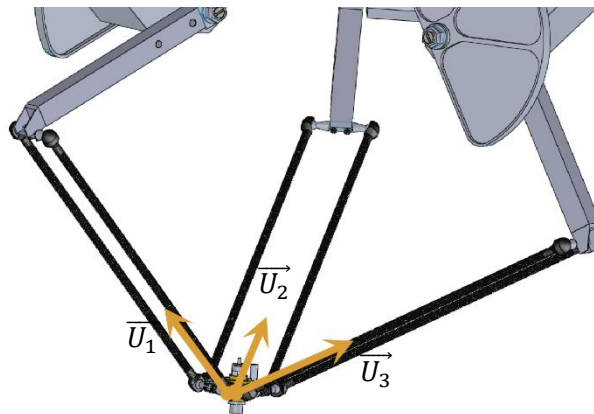


Figure 1.29 Vecteurs unitaires portés sur les barres parallèles

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 :

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

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

$$\vec{P}_{N2} = \vec{U}_2 (0 \ 1 \ 0) U^{-1} \vec{P}_N$$

$$\vec{P}_{N3} = \vec{U}_3 (0 \ 0 \ 1) U^{-1} \vec{P}_N$$

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

$$\vec{F}_{N2} = \vec{U}_2 (0 \ 1 \ 0) U^{-1} \vec{F}_N$$

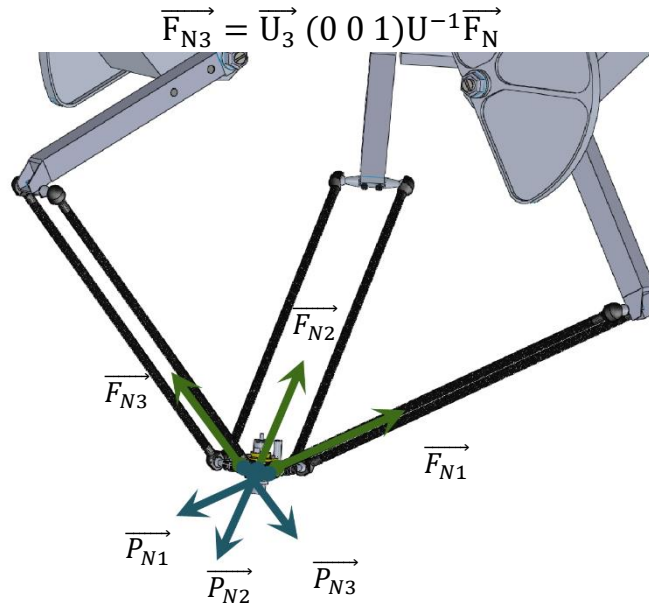


Figure 1.30 décomposition des forces et des résultantes dynamiques

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 :

$$\vec{P}_i = \vec{P}_{N_i} + \vec{P}_C$$

$$\vec{F}_i = \vec{F}_{N_i} + \vec{F}_{C_i}$$

-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 :

$$\vec{P}_{i_{oxz}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot {}^R_i T \cdot \vec{P}_i$$

$$\vec{F}_{i_{oxz}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot {}^R_i T \cdot \vec{F}_i$$

Equation de Newton-Euler :

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

Forces appliquées sur les bras :

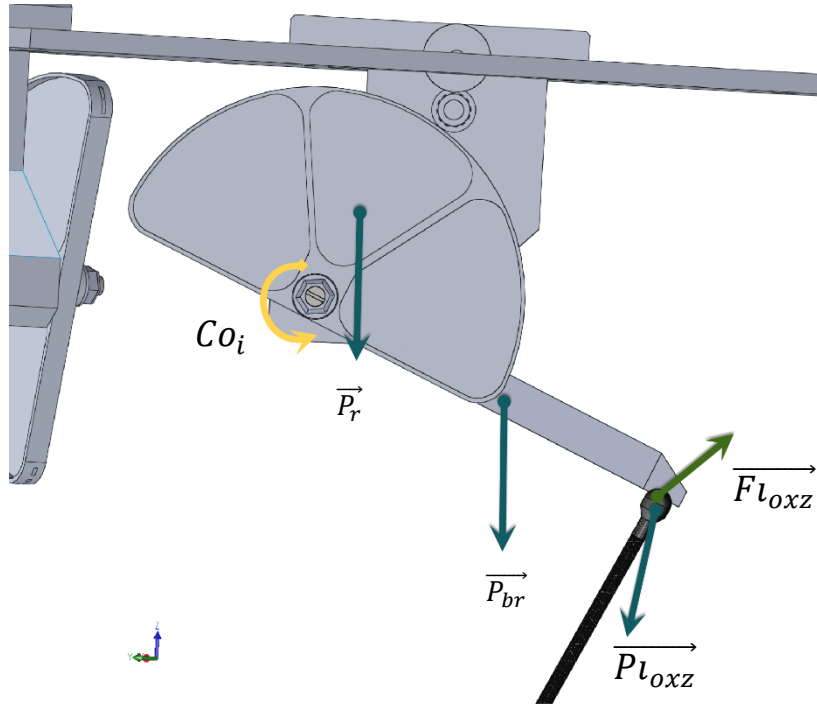


Figure 1.31 forces extérieures et résultantes dynamiques extérieures d'un bras i .

Avec :

$$\vec{P}_{br} = M_{br}\vec{g}$$

$$\vec{P}_r = M_{br}\vec{g}$$

Les bras font un mouvement de rotation autour du point A_i et dans le plan (OX_iZ_i) .

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

$$\begin{aligned} Co_i + (010)\overrightarrow{A_1C_1}\wedge\overrightarrow{P_{1_{oxz}}} + (010)\overrightarrow{A_1G_{br}}\wedge\overrightarrow{P_{br}} + (010)\overrightarrow{A_1G_r}\wedge\overrightarrow{P_r} \\ = J\alpha_i + (010)\overrightarrow{A_1C_1}\wedge\overrightarrow{F_{1_{oxz}}} \end{aligned}$$

Avec :

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

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

1.7.1. Modèle Dynamique Inverse « MDI »

La réécriture du model dynamique nous donne :

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

Avec :

$$\begin{aligned}f_1(\alpha, \dot{\alpha}, \ddot{\alpha}) &= \begin{pmatrix} (010)\overrightarrow{A_1C_1} \wedge \overrightarrow{F1_{oxz}} \\ (010)\overrightarrow{A_2C_2} \wedge \overrightarrow{F2_{oxz}} \\ (010)\overrightarrow{A_3C_3} \wedge \overrightarrow{F3_{oxz}} \end{pmatrix} \\ f_2(\alpha) &= \begin{pmatrix} (010)(\overrightarrow{A_1C_1} \wedge \overrightarrow{P1_{oxz}} + \overrightarrow{A_3G_{br1}} \wedge \overrightarrow{P_{br}} + \overrightarrow{A_1G_{r1}} \wedge \overrightarrow{P_r}) \\ (010)(\overrightarrow{A_2C_2} \wedge \overrightarrow{P2_{oxz}} + \overrightarrow{A_2G_{br2}} \wedge \overrightarrow{P_{br}} + \overrightarrow{A_2G_{r2}} \wedge \overrightarrow{P_r}) \\ (010)(\overrightarrow{A_3C_3} \wedge \overrightarrow{P3_{oxz}} + \overrightarrow{A_3G_{br3}} \wedge \overrightarrow{P_{br}} + \overrightarrow{A_3G_{r3}} \wedge \overrightarrow{P_r}) \end{pmatrix}\end{aligned}$$

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 :

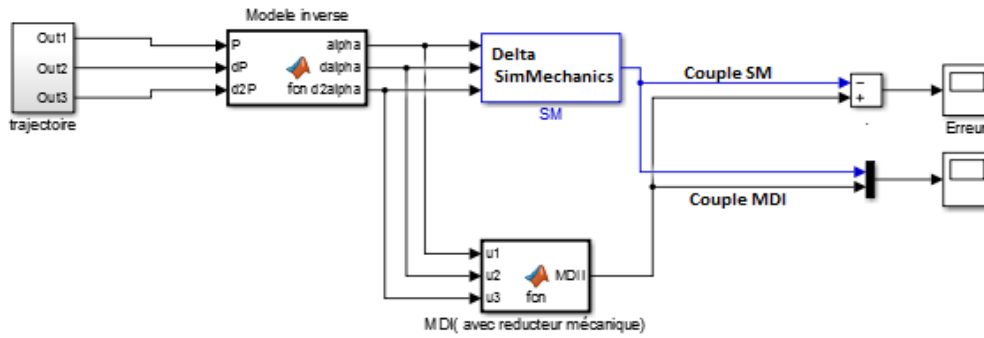


Figure 1.32 Schéma de structure de validation du MDI

Les résultats de la simulation sont :

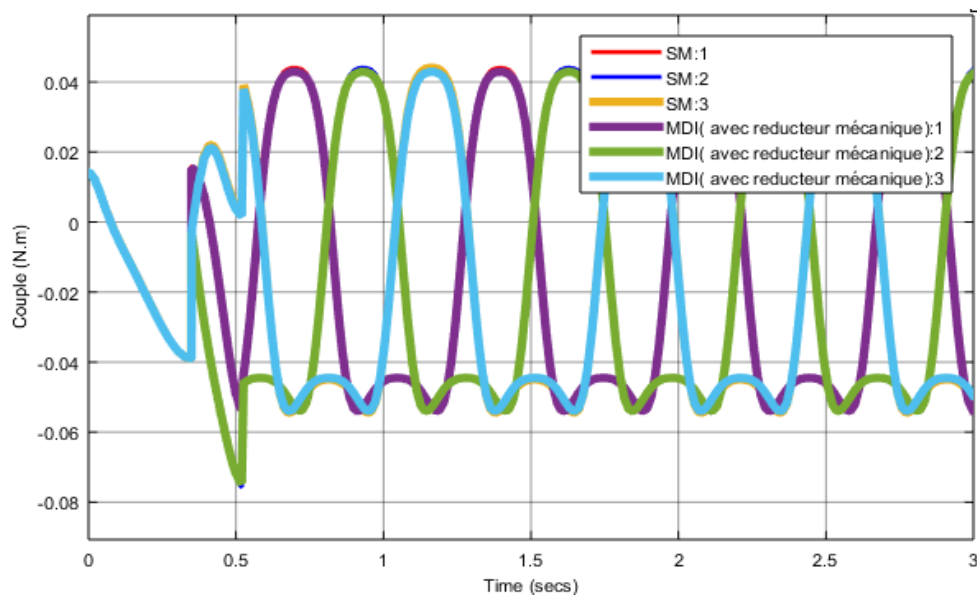


Figure 1.33 Couples donnés par les deux le MDI et SimMechanics

L'erreur est donnée par :

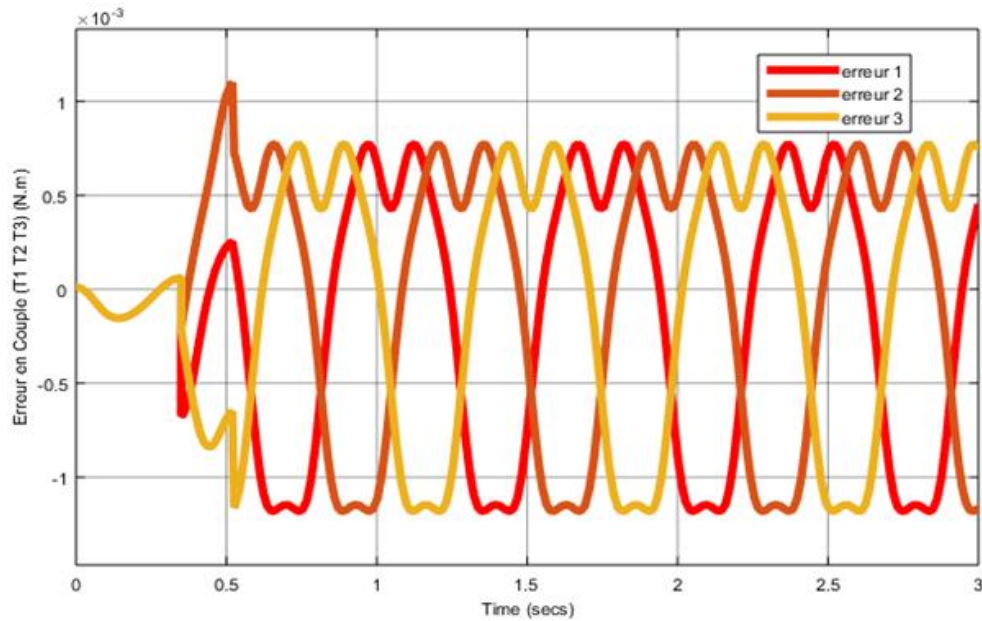


Figure 1.34 Erreur entre notre MDI et celui de SimMechanics.

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.

1.7.1. Modèle Dynamique Direct « MDD »

La réécriture du model dynamique nous donne :

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

Avec :

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

1.8. 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 .

$$I = K.U$$

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.

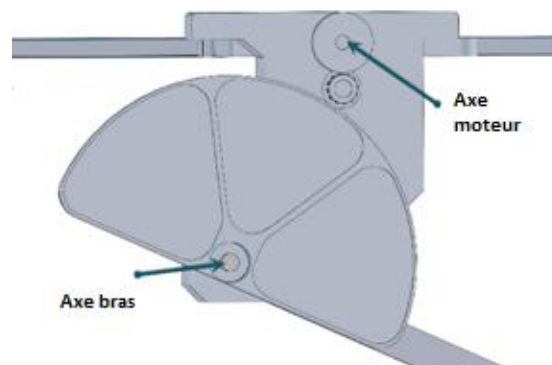


Figure 1.35 Réducteur mécanique

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 :

$$X_1 = \alpha$$

$$X_2 = \dot{\alpha}$$

Le model d'état finale est le suivant :

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

1.9. 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 2

Commande Du Robot Delta

ISIR88

Chapitre 2: Commande Du Robot Delta ISIR88

2.1. Introduction :

Dans ce Chapitre nous abordant deux lois de Commande Linéaire Basées Modèle que nous allons établir puis simuler à l'aide de l'environnement Simulink.

Pour chaque technique, nous exposons la méthode de synthèse de la commande puis ces résultats.

Enfin, une étude Comparatif entre les deux commandes est établie.

2.2. 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):

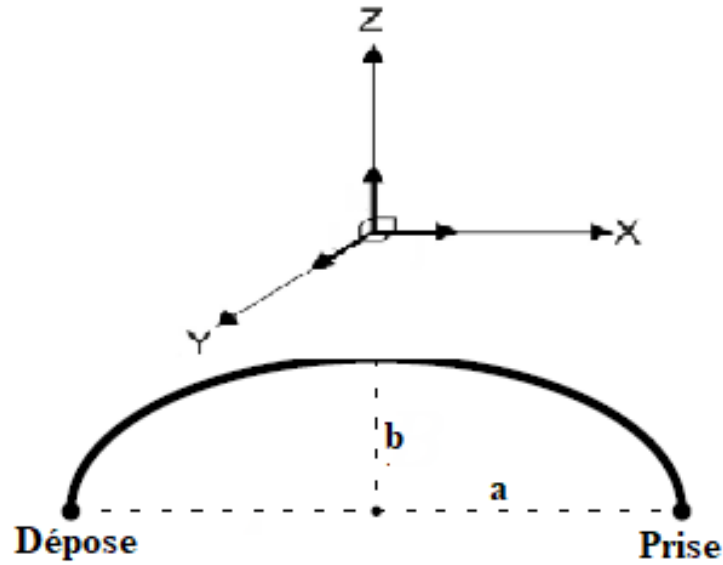


Figure 2.1 Trajectoire proposée de prise-dépose

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

2.3. Commande par la technique du couple calculé [6]

Le robot est un système physique possédant une dynamique entrée sortie non linéaire avec comme vecteur Entrée les couples moteurs et comme vecteur Sortie les positions angulaires moteurs.

La technique du couple calculé (Computed Torque) est une application particulière de la méthode du bouclage linéarisant en robotique. Elle consiste à transformer le système non linéaire en un système linéaire facile à commander. Mathématiquement, ceci se traduit par un changement de base par difféomorphisme.

Considérons le modèle dynamique suivant du robot :

$$\tau = M(q).\ddot{q} + N(q,\dot{q}) + \tau_p$$

Avec $q(t)$ la variable généralisée, $\tau(t)$ le couple de commande, τ_p une perturbation et les matrices M et N matrices du modèle dynamique (voir paragraphe 2.3.2).

Soit $q_d(t)$ une trajectoire de consigne désirée. Une première approche consiste à stabiliser l'erreur de poursuite $e = q_d - q$. Afin d'étudier l'influence du couple de commande τ sur cette erreur, nous la dérivons deux fois pour obtenir :

$$\dot{e} = \dot{q}_d - \dot{q}$$

$$\ddot{e} = \ddot{q}_d - \ddot{q}$$

Et sachant $\ddot{q} = M^{-1}(-N - \tau_p + \tau)$ l'expression de \ddot{e} devient :

$$\ddot{e} = \ddot{q}_d - \ddot{q} = \ddot{q}_d - M^{-1}(N + \tau_p - \tau)$$

En posant :

$$v = \ddot{q}_d + M^{-1}(N - \tau) \quad \text{La nouvelle commande}$$

$$w = M^{-1}\tau_p \quad \text{La nouvelle perturbation}$$

$$X = \begin{pmatrix} e \\ \dot{e} \end{pmatrix} \quad \text{L'état du système}$$

Notre nouvelle représentation du système devient :

$$\begin{cases} \dot{X} = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} X + \begin{pmatrix} 0 \\ I \end{pmatrix} v + \begin{pmatrix} 0 \\ I \end{pmatrix} w \\ Y = (I \quad 0)X \end{cases}$$

Plus précisément la dynamique par rapport à la nouvelle paire (v, e) équivaut à un double intégrateur pour chaque erreur : $\ddot{e} = v$.

Une deuxième approche serait d'utiliser un bouclage particulier. Afin d'obtenir le couple de commande τ , on commence par synthétiser une commande linéaire v (par retour d'état par exemple), et ensuite, le couple τ est obtenu par le bouclage suivant :

$$\tau = M \cdot v + N$$

En remplaçant cette expression du couple dans l'expression de \ddot{q}

$$\ddot{q} = M^{-1}(-N - \tau_p + M \cdot v + N) = v + w$$

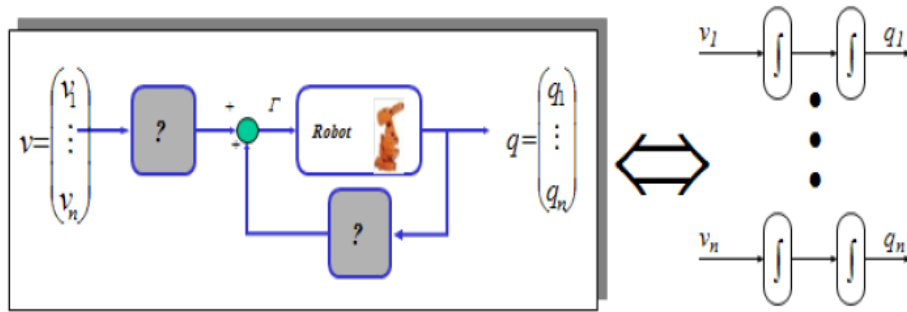


Figure 2.2 Effet du bouclage linéarisant sur le système [7]

Il suffit maintenant de stabiliser q à l'aide d'un régulateur (PD, PID, Retour d'état, ...). En effet, une commande linéaire v qui stabilise l'erreur e nous donne ensuite un couple τ qui assure le suivi de la trajectoire désiré q_d .

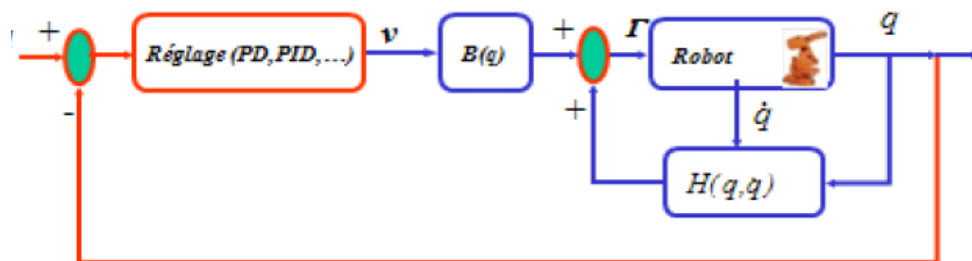


Figure 2.3 Boucle linéarisant et stabilisation [7]

2.3.1. Technique du couple calculé avec retour d'état

Dans ce paragraphe, nous calculons la commande linéaire v par retour d'état, c'est-à-dire $v = -k X$ avec k obtenu grâce à un placement de pôles. Cette commande est ensuite utilisée pour le calcul du couple τ qui est injecté au système. A noter la nécessité d'ajouter un pré compensateur $N = (C(-A + BK)^{-1}B)^{-1}$ pour éliminer l'erreur statique. Le schéma global est le suivant:

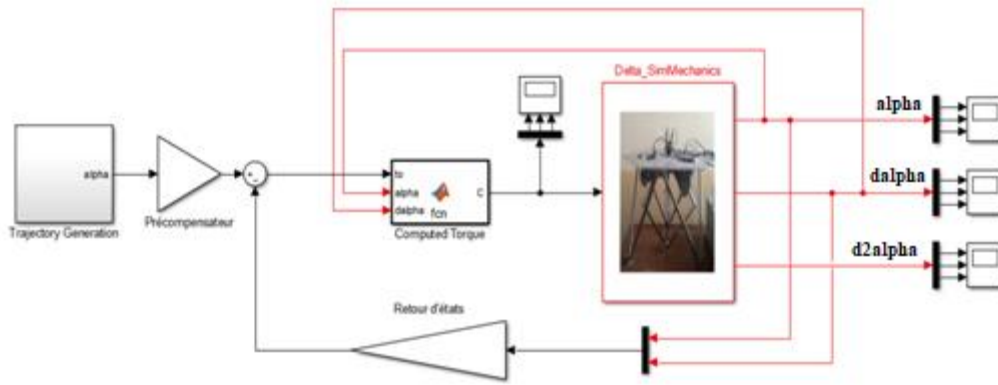


Figure 2.4 Schéma de commande Computed Torque avec Retour d'Etat

Avec $A = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix}$; $B = \begin{pmatrix} 0 \\ I \end{pmatrix}$; $C = (I \ 0)$ et I la matrice identité 3×3

Un bon placement de pôles donne les résultats suivants :

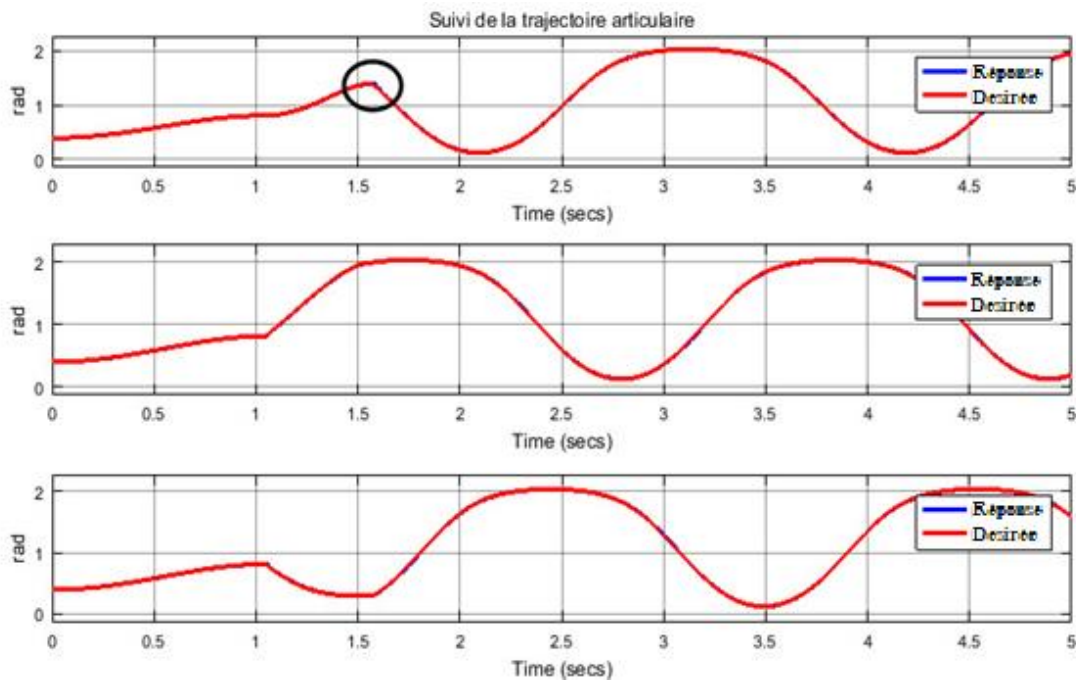


Figure 2.5 Suivi de trajectoire articulaire Computed Torque avec Retour d'Etat

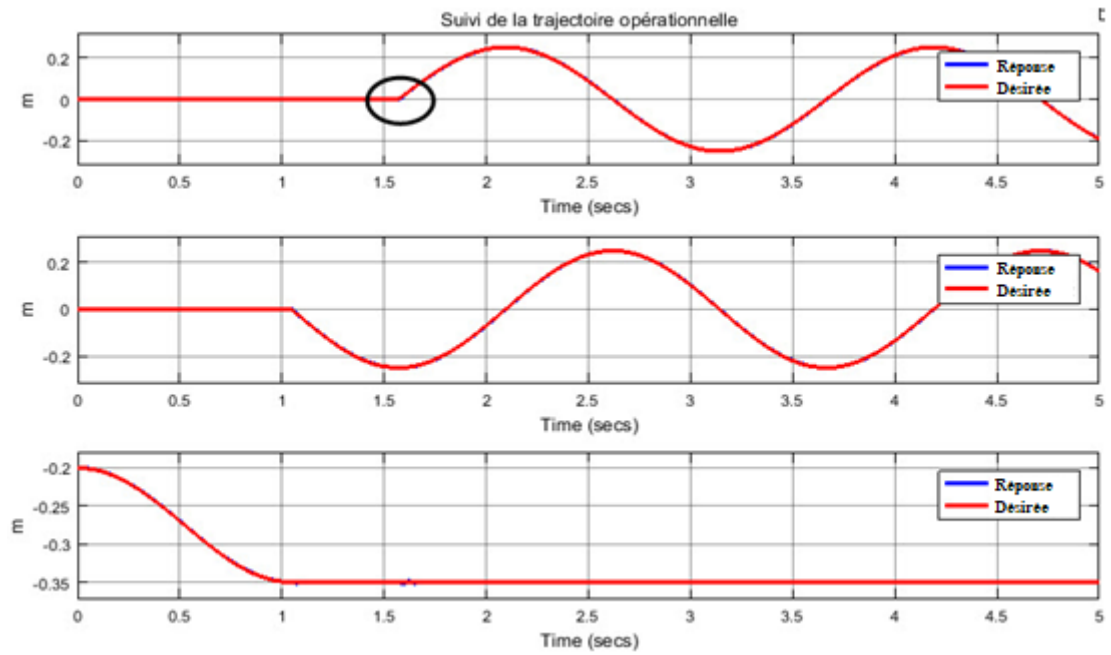


Figure 2.6 Suivi de trajectoire opérationnelle Computed Torque avec Retour d'Etat

Comme nous pouvons le remarquer, le suivi de trajectoire est respecté dans l'espace articulaire et opérationnel. L'erreur entre la position de la Nacelle et celle désirée est :

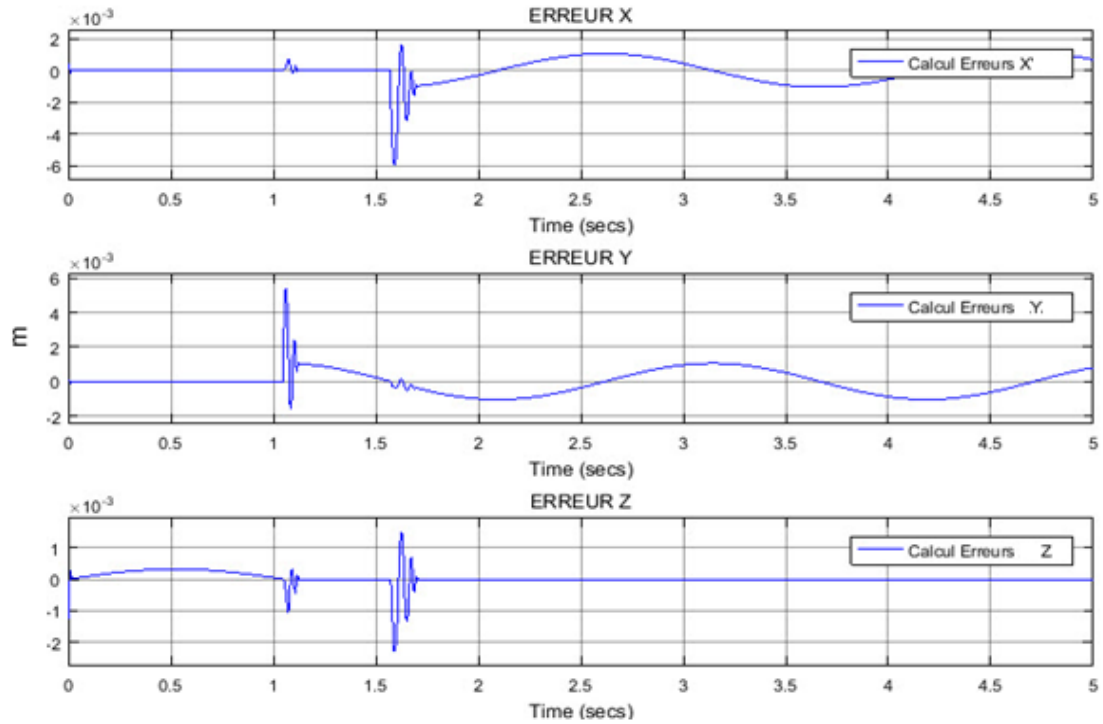


Figure 2.7 Erreurs opérationnelles Computed Torque avec Retour d'Etat

L'erreur en position opérationnelle est de 10^{-3} . Elle est aussi faible dans l'espace articulaire comme le montre la figure suivante :

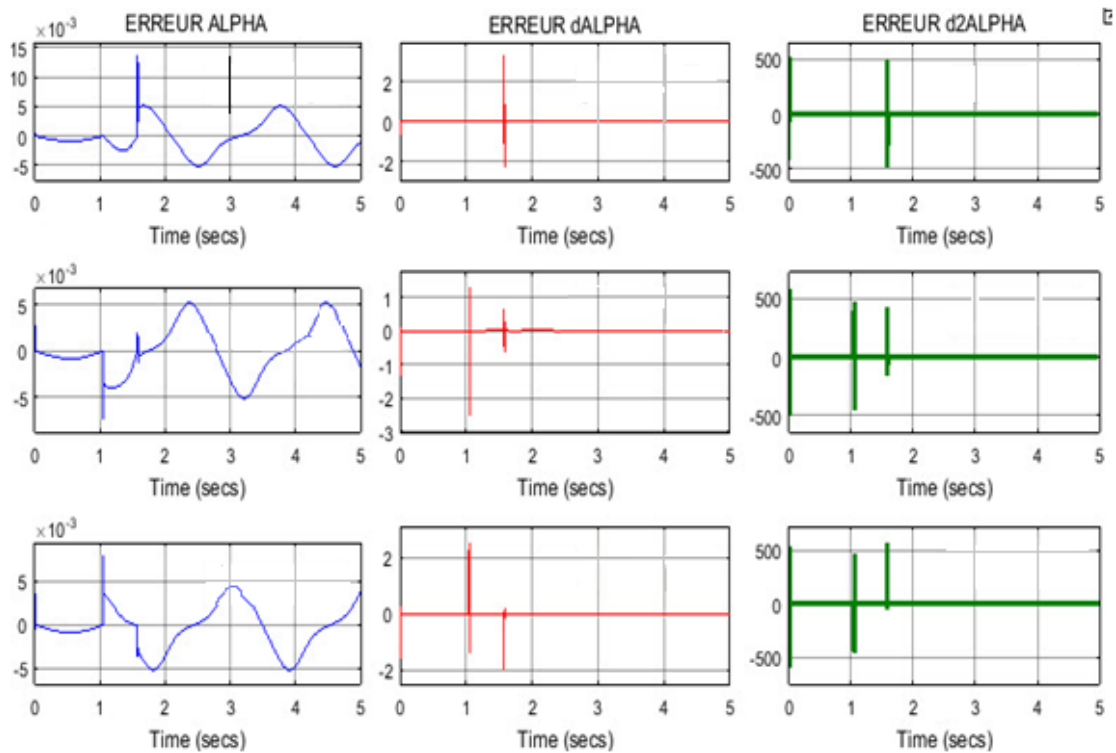


Figure 2.8 Erreurs articulaires Computed Torque avec Retour d'Etat

(Erreur alpha en rad, $\dot{\alpha}$ en rad/s et $\ddot{\alpha}$ en rad/s²)

De plus, ces performances sont atteintes avec une commande admissible, c'est-à-dire supportable par les actionneurs du robot.

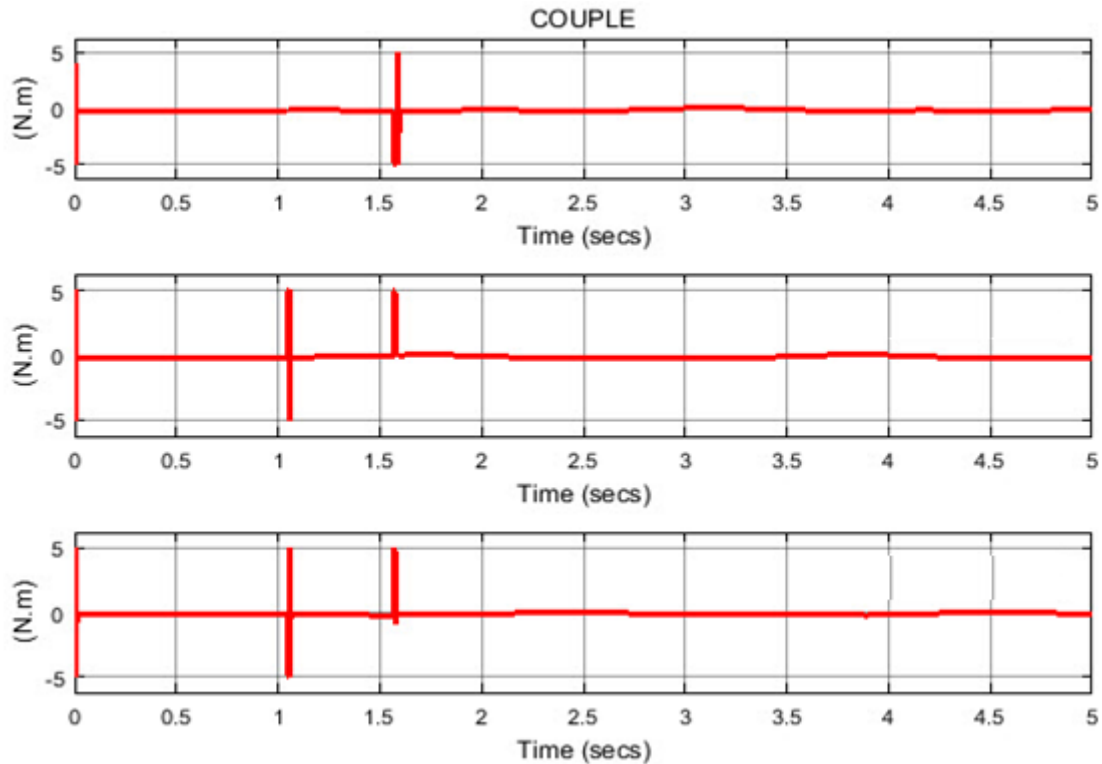


Figure 2.9 Couple calculé avec retour d'état (les trois couples de chaque bras)

Le robot Delta opérant dans le milieu industriel, les perturbations et les bruits sont très courants. Afin de se mettre dans les conditions réelles du robot, nous injectons la perturbation aléatoire suivante :

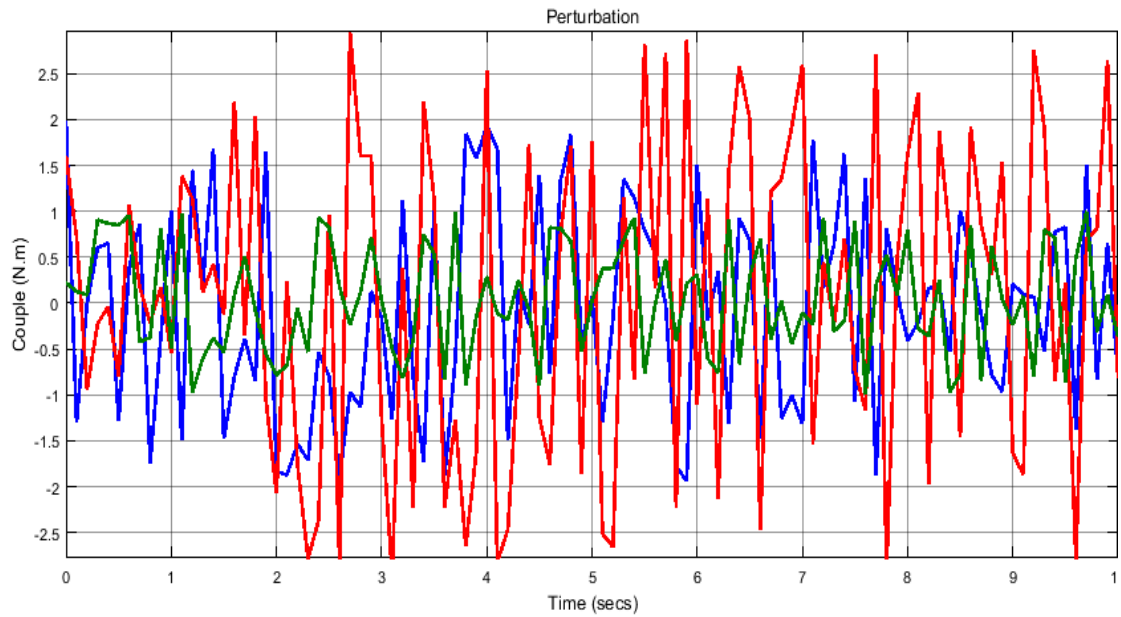


Figure 2.10 Perturbation pour Computed Torque avec Retour d'Etat

Le comportement de notre robot en la présence d'une perturbation est exprimé par l'erreur en position suivante :

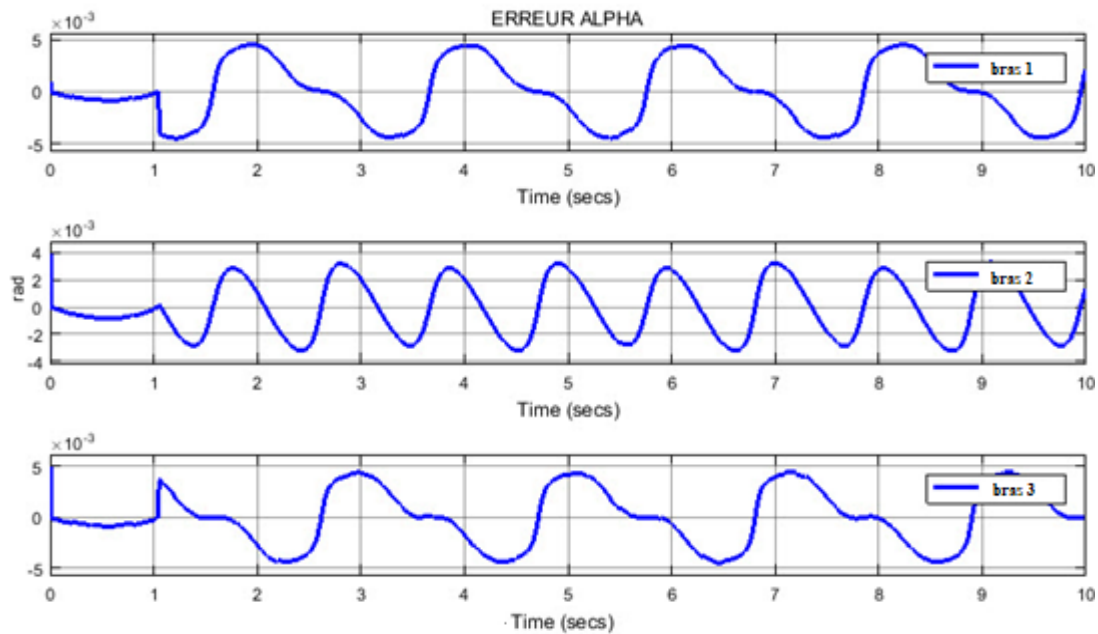


Figure 2.11 Erreur Computed Torque et Retour d'Etat avec perturbation

L'erreur est toujours de l'ordre de 10^{-3} . D'où la validation de notre commande.

2.3.1. Technique du couple calculé avec régulateur PD

Nous continuons avec la méthode du couple calculé, mais cette fois, la commande v est asservie par un régulateur PD. Comme le montre le schéma ci-dessous, la commande v est donnée par :

$$v = -Kd.\dot{e} - Kp.e$$

Dans ce cas, le système linéaire devient :

$$\begin{cases} \dot{X} = \begin{pmatrix} 0 & I \\ -Kp & -Kd \end{pmatrix} X + \begin{pmatrix} 0 \\ I \end{pmatrix} w \end{cases}$$

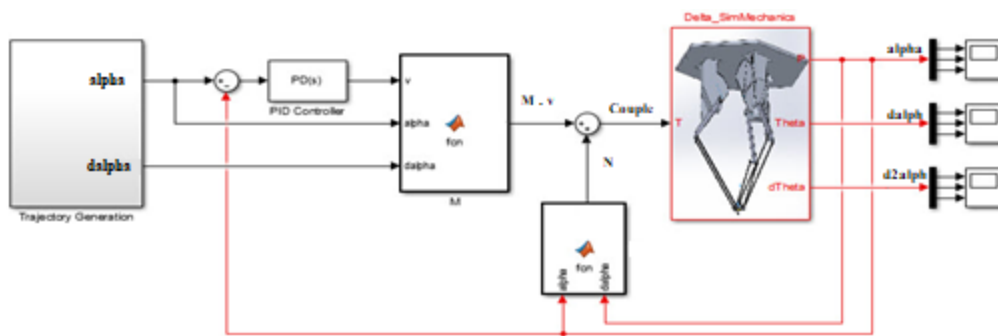


Figure 2.12 Schéma de régulation Computed Torque + PID

La matrice A étant donnée par

$$A = \begin{pmatrix} 0 & I \\ -Kp & -Kd \end{pmatrix}$$

Le polynôme caractéristique (dénominateur du système linéaire) est :

$$\Delta(s) = s^2 + Kd.s + Kp$$

Par identification avec le dénominateur d'un système de second ordre :

$$\Delta(s) = s^2 + 2\xi\omega_n.s + \omega_n^2 \quad \text{avec } \xi \text{ le coefficient d'amortissement et } \omega_n \text{ la pulsation}$$

Il en résulte :

$$Kp = \omega_n^2 \quad ; \quad Kd = 2\xi\omega_n$$

Chapitre 2 : Commande du Robot Delta ISIR88

En automatique, on impose souvent $\zeta = \frac{1}{\sqrt{2}}$. Quant à ω_n , on la choisit pour atteindre l'objectif de commande tout en ayant une commande admissible (tolérée par les actionneurs). La valeur 60 rad/s semble raisonnable.

En appliquant ces valeurs, la simulation donne :

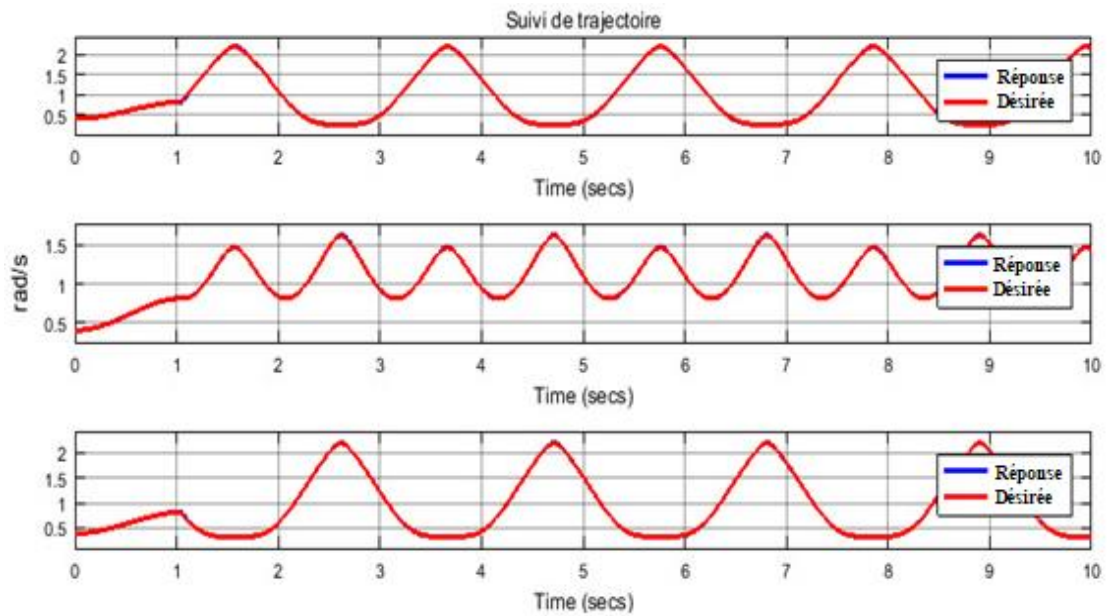


Figure 2.13 Suivi de trajectoire Computed Torque + PID

Et l'erreur de positionnement est donnée par :

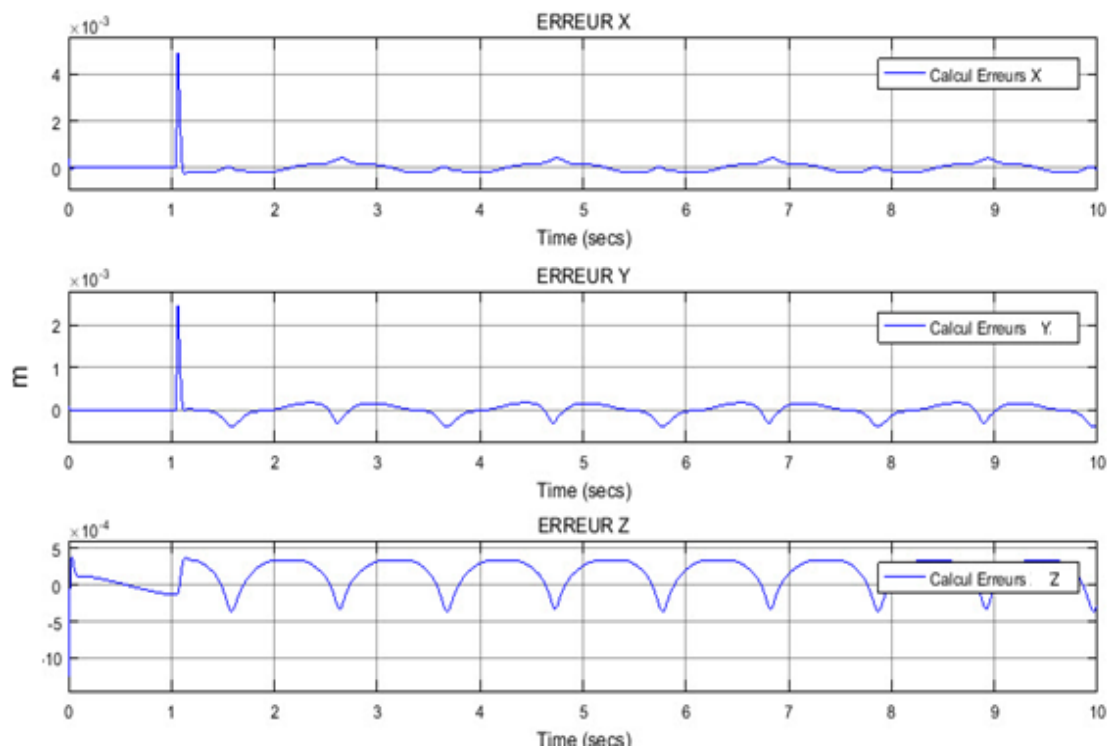


Figure 2.14 Erreur de positionnement de la nacelle Computed Torque + PID

Nous remarquons que l'erreur de la position de la nacelle par rapport à celle désirée est de l'ordre de 10^{-3} . Observons maintenant les erreurs de l'espace articulaire :

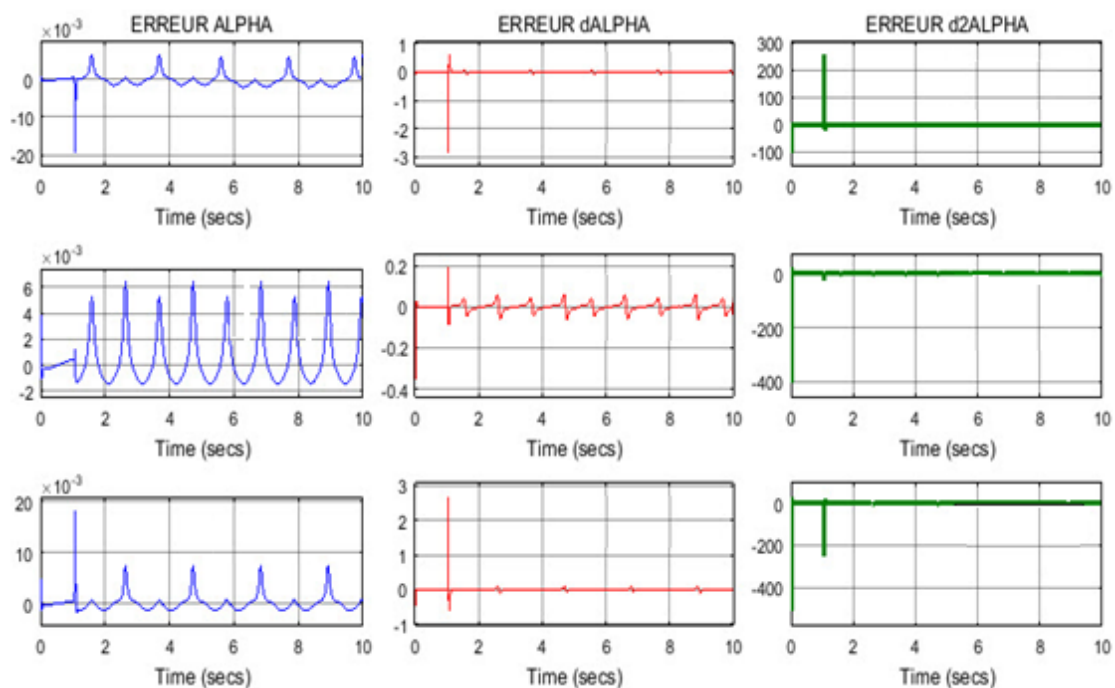


Figure 2.15 Erreurs articulaires Computed Torque + PID

(Erreur alpha en rad, $\dot{\alpha}$ en rad/s et $\ddot{\alpha}$ en rad/s²)

Cette méthode de commande assure ainsi un suivi parfait de la trajectoire, et ceci, en appliquant un couple admissible au robot (voir figure suivante).

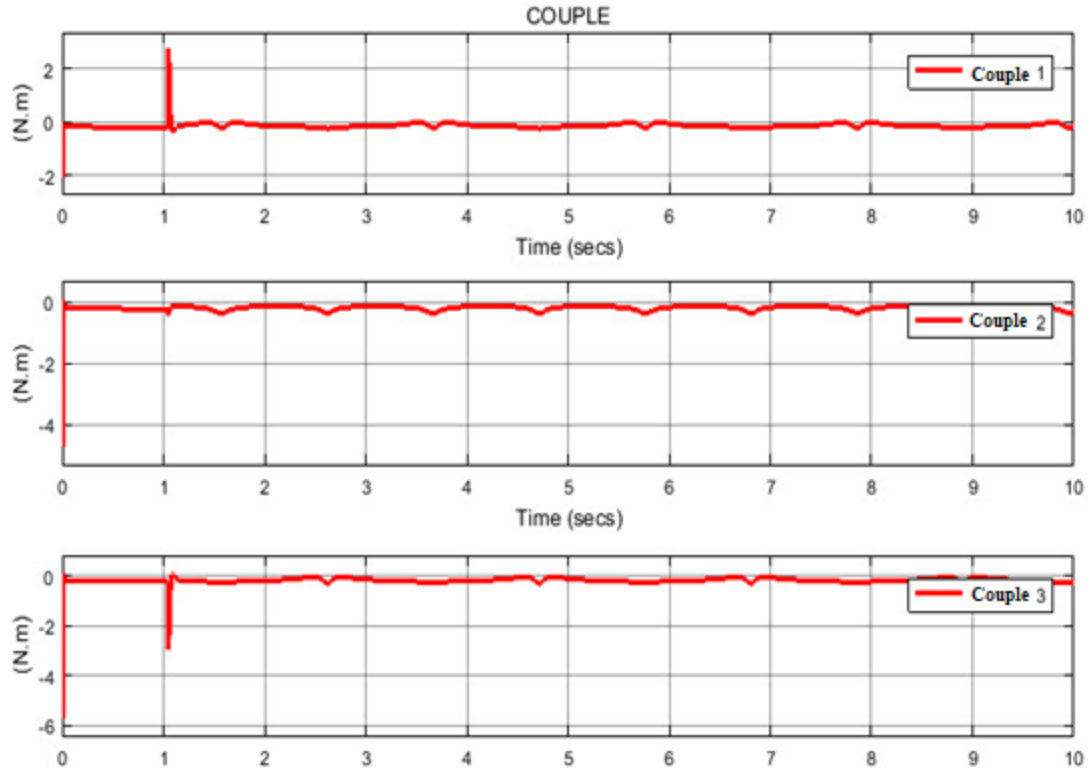


Figure 2.16 Valeurs des couples Computed Torque + PID

2.4. Commande par Gain Pré programmé

Cette Technique de commande consiste à calculer une commande linéaire par rapport au modèle linéaire du système autour du point de fonctionnement, cette commande sera ensuite ajoutée au couple qui ramène le système vers ce point de fonctionnement.

$$U = \Delta U + U_0$$

2.4.1. Linéarisation :

Dans cette partie nous tirons les modèles linéaires autour du point de fonctionnement de notre système non linéaire.

Nous avons notre système :

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

En écrivant le modèle comme suite :

$$\dot{X} = F(X, U)$$

La linéarisation du système se fait autour du point de fonctionnement $\begin{pmatrix} X_f \\ U_f \end{pmatrix}$

Et nous avons donc :

$$X = \Delta X + X_f$$

$$U = \Delta U + U_f$$

En écrivant le développement de Taylor du modèle non linéaire au premier ordre:

$$\dot{X} = \Delta \dot{X} = \frac{\partial F}{\partial X} \partial X + \frac{\partial F}{\partial U} \partial U$$

Le Modèle linéaire au point de fonctionnement est le suivant :

$$\Delta \dot{X} = \frac{\partial F}{\partial X}(X_f, U_f) \Delta X + \frac{\partial F}{\partial U}(X_f, U_f) \Delta U$$

$$\Delta Y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \cdot X$$

Pour calculer ces matrices Jacobiennes du modèle nous avons utilisé la dérivation numérique, nous avons donc obtenu notre modèle linéaire :

$$\Delta \dot{X} = A \cdot \Delta X + B \cdot \Delta U$$

$$\Delta Y = C \cdot X$$

Avec :

$$A = \frac{\partial F}{\partial X}(X_f, U_f)$$

$$B = \frac{\partial F}{\partial U}(X_f, U_f)$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

2.4.1. Commande par Retour d'état :

Pour appliquer cette commande nous prenons le modèle linéaire du système autour du point de la trajectoire désiré à chaque instant :

Donc la commande U_f est calculée avec le Modèle dynamique inverse car il représente le couple nécessaire pour avoir notre système à un point de la trajectoire désirée.

$$U_f = J\ddot{\alpha}_f + f_1(\alpha_f, \dot{\alpha}_f, \ddot{\alpha}_f) - f_2(\alpha_f)$$

Une fois notre modèle linéarisé, il nous suffit de calculer la commande linéaire de ce modèle.

Nous posons donc :

$$\Delta U = -K_f \cdot \Delta X$$

Avec :

K : matrice (3x6) obtenu grâce à un placement de pole.

Enfin la commande finale par cette technique :

$$U = U_f + \Delta U$$

$$U = J\ddot{\alpha}_f + f_1(\alpha_f, \dot{\alpha}_f, \ddot{\alpha}_f) - f_2(\alpha_f) - K_f \cdot \Delta X$$

Le schéma de la structure de commande est le suivant :

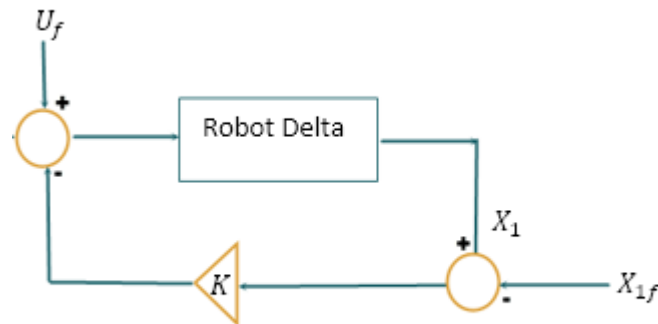


Figure 2.17 Schéma de la structure de Commande par Gain Préprogrammé

Résultat de simulation de la commande :

Nous avons appliqué la loi commande synthétisé sur le bloque SimMecanics représentant notre Robot Delta ISIR88 sous Simulink, nous avons eu les résultats suivants :

Suivi de trajectoire

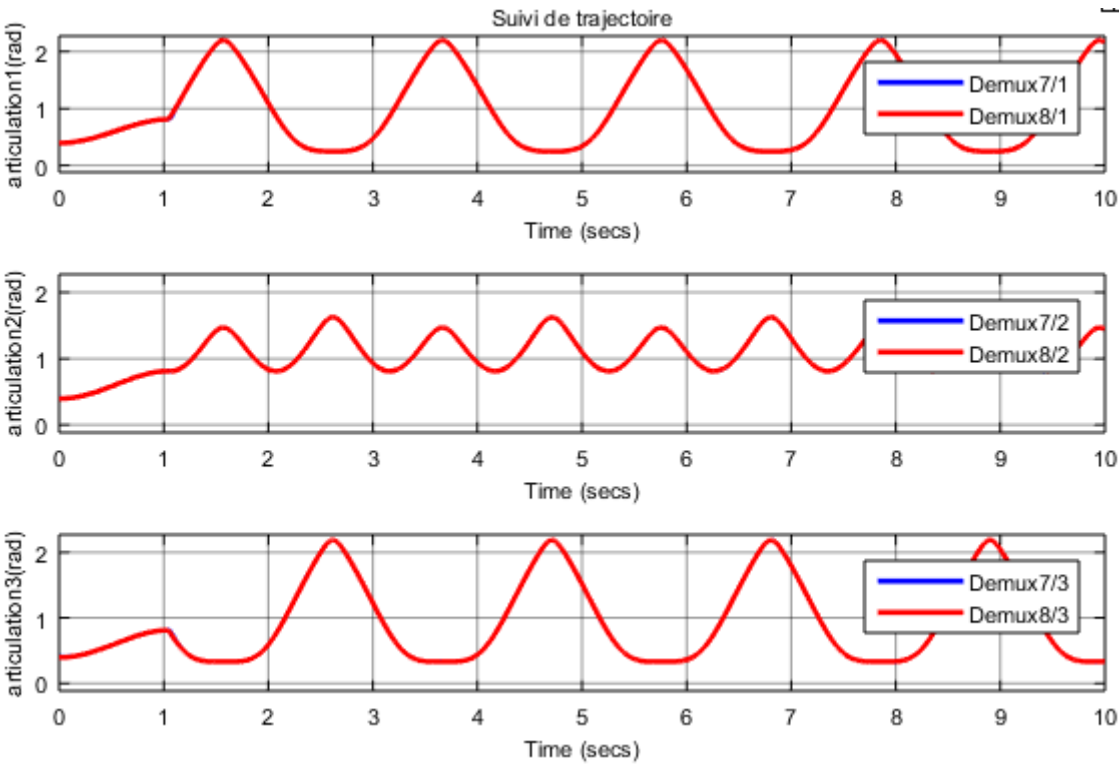


Figure 2.18 Suivi de Trajectoire Articulaire

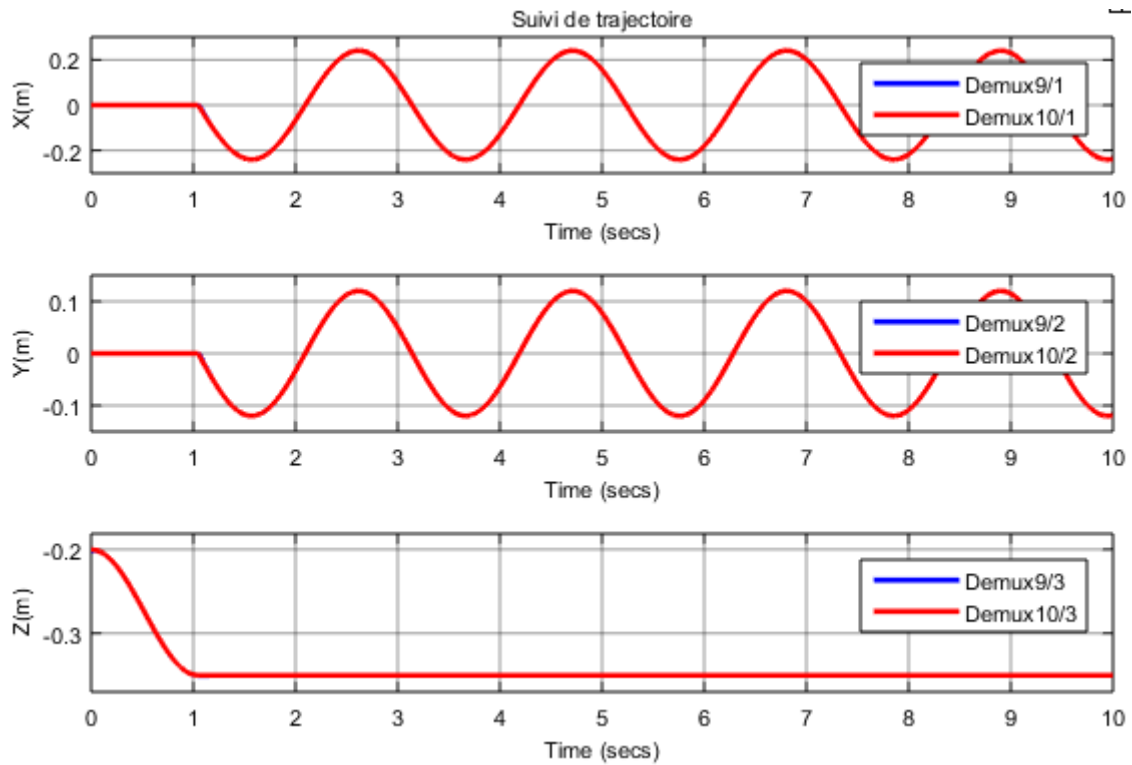


Figure 2.19 Suivi de Trajectoire Operationnel

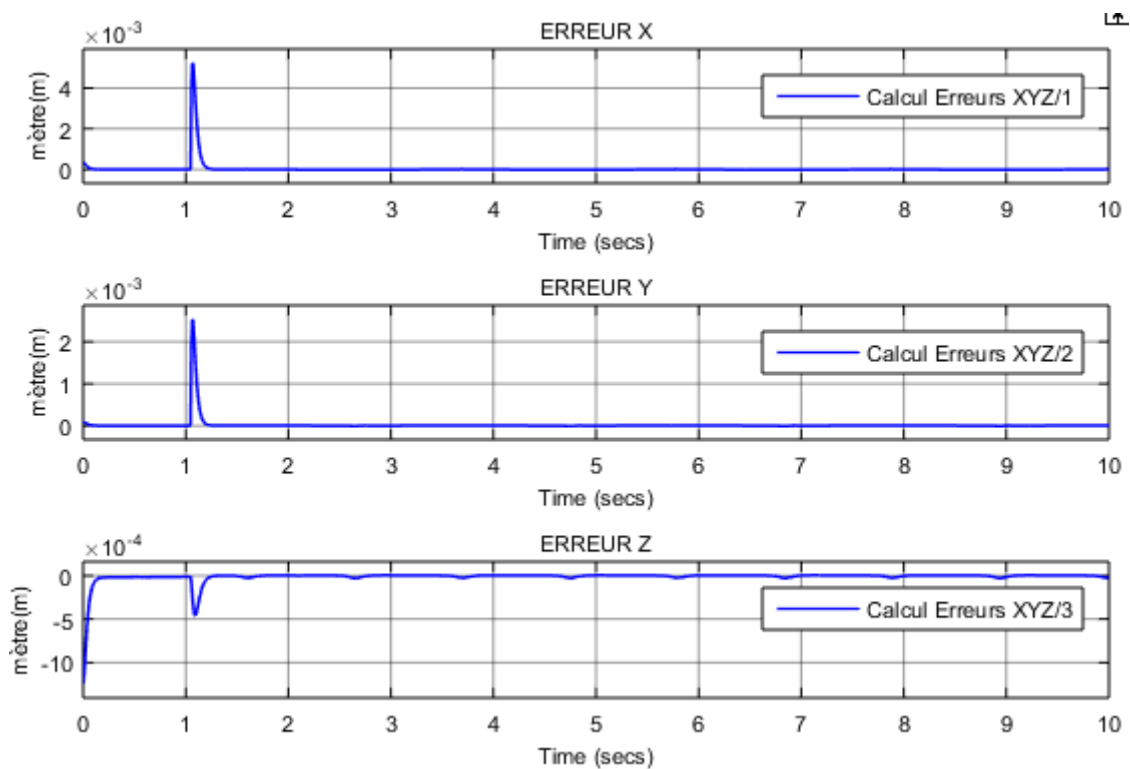


Figure 2.20 Erreur sur la Trajectoire operationnel

Commentaire :

On obtient avec cette commande une erreur relatif pour le suivi de la trajectoire opérationnel de l'ordre de 10^{-5}

Donc on en conclue que le suivi de la trajectoire opérationnel est très bon.

Suivi de la Dynamique

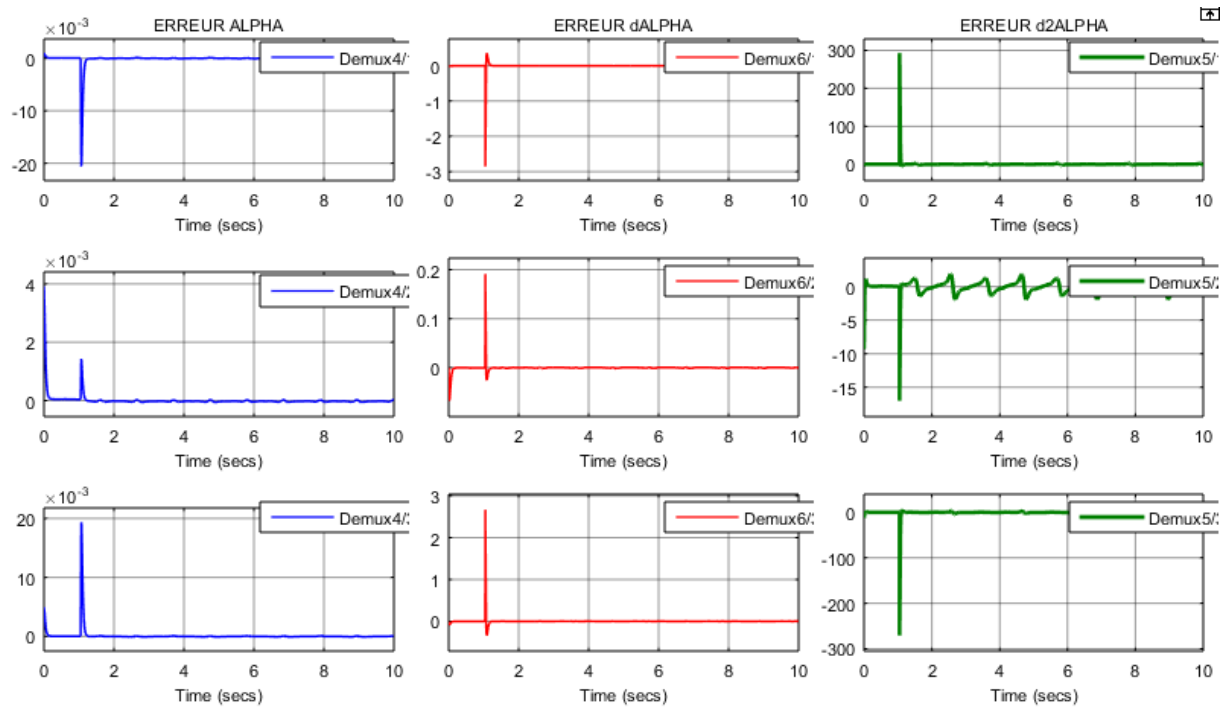


Figure 2.21 Erreurs sur la dynamique des articulations

Commentaire :

La Figure montre les erreurs en position, vitesse et accélération pour les trois articulations.

Nous avons pour les vitesses des erreurs relatives de l'ordre de 10^{-4} , donc on a un bon suivi en vitesse.

Mais la performance en accélération est moins bonne, en effet l'erreur relatif en accélération atteint les 10%, cette erreur est acceptable vue l'exigence de l'utilisation du robot dans une tache de prise dépose.

Couples des articulation

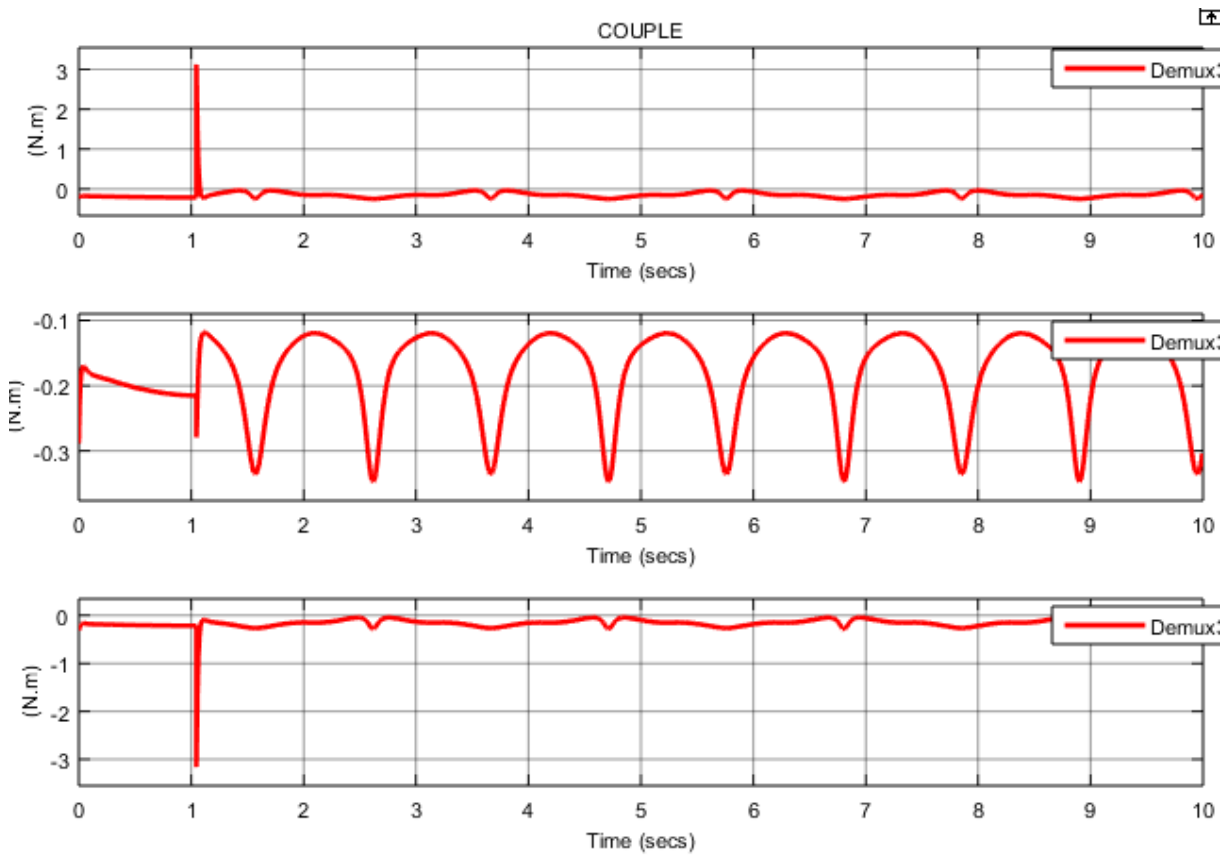


Figure 2.22 Couples de la commande par gain préprogrammé

Commentaire :

Les couples des articulations sont admissibles, et nos actionneurs peuvent générer ces couples grâce aux réducteurs mécaniques.

2.5. Etude Comparatif

Pour ce Projet de Fin d'étude, nous avons pu synthétiser deux types de commande par bouclage linéarisant :

Commande par gain Pré Programmé.

Et la technique du couple calculé.

En comparant les figures 2.8 et 2.21, on voit que les erreurs articulaires dans le cas de la commande par Gain Préprogrammé sont meilleurs, et on a aussi des couples plus petits, cela est démontré en comparant les figures 2.9 et 2.22, donc la commande par Gain Préprogrammé génère des couple plus admissibles pour les actionneurs .

2.6. Conclusion

Dans ce chapitre Nous avons établi deux lois de commandes, la Commande par gain préprogrammé, et la commande par couple calculée.

Puis nous avons eu les résultats de simulation sur le modèle SimMechanics des commandes synthétisées, et les résultats de ces lois de commande sont satisfaisants.

Après comparaison des deux commandes synthétisées, nous avons trouvé que la commande par gain Préprogrammé donne de meilleurs résultats.

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, l'utilisation du calcul symbolique nous donne explicitement la matrice Jacobienne. 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 par bouclage linéarisant, nous avons conçu deux Commandes de ce type, la commande par gain Préprogrammé et la technique du couple calculé.

Une fois la trajectoire de consigne générée et les schémas de régulations choisies, nous avons comparé nos deux commandes, Selon nos simulations, la commande

par gain Préprogrammé donnait les meilleurs résultats, cette dernière est plus performante en termes de précision dans le suivi de trajectoire et en admissibilité de la commande.

Ce travail constitue donc une base solide pour les prochains projets du LCP touchant à la robotique parallèle. Les perspectives à moyen terme pour compléter ce travail concernent en premier lieu l'implémentation pratique des lois de commandes synthétisées dans le but d'améliorer les performances du robot par rapport aux commandes déjà implémenté.

Bibliographie

- [1] N. I. Mokhtari et H. Nadour, Contribution à la modélisation et à la commande du robot delta ISIR88, ENP Alger, 2016.
- [2] R. Clavel, Conception d'un robot parallèle rapide à 4 degrés de liberté, EPFL Lausanne, 1991.
- [3] R. Clavel et A. Codourey, Contribution à la commande des robots rapides et précis: application au robot delta à entraînement direct, IEEE, 1991.
- [4] R. Clavel, DELTA, a fast robot with parallel geometry, 18th International Symposium on Industrial Robots, p.91-100, IFS Publications, 1988.
- [5] 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.
- [6] H. Walid, Modélisation et commande du Robot Delta ISIR88 par la technique du Couple Calculé, 2017.
- [7] A. Olsson, Modeling and control of a Delta-3 robot, Lund University: Department of Automatic Control, 2009.
- [8] O. Hamdoun, F. Z. Baghli et L. Elbakkali, Inverse Kinematic Modeling of 3RRR Parallel Robot, Tetouan, Morocco: Faculty of Sciences.

- [9] S.-D. Stan, M. Manic, C. Szep et R. Balan, Performance analysis of 3 DOF Delta parallel, University of Idaho, 2011.

- [10] K. Rosquist, Modelling and Control of a parallel kinematic robot, Lund University: Department of Automatic Control 2013.

- [11] F. Paccot, N. Andreff et P. Martinet, A Review on the dynamic control of parallel kinematics: theory and experiments, Clermont-Ferrant, 2016.

- [12] K. Oualid et A. Larbi, Modélisaiton et commande d'un robot parallèle de type delta, ENP Alger, 2010.

- [13] 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.

- [14] R. Meryem, Modélisation et Commande Robuste de Robots Parallèles: Application au Robot Delta, ENP Alger, 2016.

- [15] A. B. Lintott, Geometric Modelling and Accuracy enhancement of parallel manipulators, University of Canterbury: Mechanical Engineering, 2000.

- [16] N. R. CREȚESCU, KINEMATIC AND DYNAMIC SIMULATION, University of Braşov: Bulletin of the Transilvania, 2015.

- [17] M. Bouri, Getting started with FlexCom, 2014.

- [18] M. Bouri, FlexCom A windows PC based robot controller An open architecture, Moveit Automation.

- [19] M. Bouri, Commande des robots, 2014.

- [20] J. L. Boimond, Cours de robotique, Université Angers: ISTIA.
- [21] S. Belhabib, Contribution à l'étude du robot Delta, ENP Alger, 2005.
- [22] 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.
- [23] H. Alashqar, Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator, The Islamic University of Gaza, 2007.
- [24] «Site Web du Laboratoire de Systèmes Robotiques,» [En ligne]. Available: <http://lsro.epfl.ch/>.
- [25] R. W. II, The Delta Parallel Robot: Kinematics Solutions, Ohio University: Mechanical Engineering, 2016.
- [26] 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.
- [27] X. Lu et M. Liu, «Optimal Design and Tuning,» *International Journal of Advanced Robotic Systems*, 2016.