

Université de Lille 1
Master 2 SMaRT
Systèmes, Machines autonomes et Réseaux de Terrains



RAPPORT DE PROJET

Contrôle d'un robot parallèle à 6 ddl

CATTIAU Valentin

supervisé par M. Giraud Frédéric

MdC HDR au L2EP

Remerciements

Je tenais à remercier, tout d'abord, les instances du M2 SMaRT de l'Université Lille 1 car grâce à elles nous avons eu l'opportunité de réaliser ce projet lors du troisième semestre du M2 SMaRT.

Je tenais également à remercier Monsieur L. BELKOURA, responsable du M2 SMaRT, pour ce travail d'organisation si bien réalisé.

Puis, je souhaitais remercier l'Institut de Recherche IRCICA pour avoir accepté de me recevoir lors des réunions avec mon tuteur.

Pour finir, je tiens tout naturellement à remercier mon tuteur Monsieur F. GIRAUD pour m'avoir si bien épauler durant ce semestre. Il a su prendre son temps pour m'expliquer des phénomènes qui lui semblaient essentiels pour le bon déroulement du projet. Je remercie aussi Monsieur C. DURIEZ pour m'avoir débloqué dans la partie modélisation de mon projet.

Table des matières

Remerciements	i
Sommaire	ii
Liste des figures	v
Abbréviations	vii
1 Introduction	1
2 Outils pour l'ingénieur	3
2.1 Modélisation du système	3
2.1.1 Matrices de translations et rotations	3
2.1.2 Les angles de Tangage, Roulis et Lacet	4
2.1.3 Changement de variable avec la tangente de l'arc moitié	4
3 Modélisation du robot	5
3.1 Repèrages et paramètres du robot	5
3.1.1 Repèrages du robot	5
3.1.2 Paramètres du robot	5
3.1.3 Entrées et sorties du robot	6
3.2 Cinématique inverse du robot	6
3.2.1 Coordonnées de A_{ij} dans R_A	6
3.2.2 Coordonnées de C_{ij} dans R_C	6
3.2.3 Coordonnées de C_{ij} dans R_A	7
3.2.4 Coordonnées de C_{ij} dans R_{Aij}	7
3.2.5 Coordonnées de B_{ij} dans R_{Aij}	7
3.2.6 Calcul des angles θ_{ij}	7
3.3 Cinématique direct du robot	7
3.3.1 Décomposition des variables	8
3.3.2 Cinématique direct en translation du robot	8
3.4 Dynamique inverse du robot en translation	9
3.5 Dynamique directe du robot en translation	9
3.6 Modélisation sous Matlab simulink	9
3.6.1 Equation dynamique du robot	9
3.6.2 Representation sous Maltab simulink	10
4 Contrôle du robot	11
4.1 Etude en boucle ouverte	11
4.2 Modélisation du contrôle	12
4.3 Design du PID	13
4.3.1 Représentation d'Etat	13
4.3.2 Régulateur PID	14

4.3.3	Placement de pôles	14
4.4	Proposition de contrôle des mouvements de rotation de la plateforme	16
5	Résultats de simulation en boucle fermée	17
5.1	Trajectoire en x et y	17
5.2	Trajectoires selon x, y et z	18
5.3	Couples des moteurs	20
6	Conclusion et perspectives	25
	Bibliographie	26

Table des figures

1.1	Photo du robot Hexapod à 6 DDL	1
2.1	Angles de Tangage, Roulis et Lacet, source : floconcept.fr	4
3.1	Schéma du robot hexapod avec ses repères et ses paramètres	5
3.2	Détermination de la cinématique direct	8
3.3	Schéma de modélisation du robot Hexapod	10
4.1	Schéma de modélisation de la boucle ouverte du robot Hexapod sous Simulink	11
4.2	Etude du robot en boucle ouverte	12
4.3	Schéma de modélisation du robot Hexapod avec la boucle de contrôle	13
4.4	Schéma structurel du régulateur PID	14
4.5	Détermination de la pulsation propre du robot	15
4.6	Schéma de description du contrôle en rotation et translation de la plateforme	16
5.1	Réponse du système de x en fonction de y	17
5.2	$X_{référence}$ et X_{mesure} en fonction du temps	18
5.3	Erreur selon X en fonction du temps	18
5.4	$Y_{référence}$ et Y_{mesure} en fonction du temps	19
5.5	Erreur selon Y en fonction du temps	19
5.6	$Z_{référence}$ et Z_{mesure} en fonction du temps	20
5.7	Couple selon θ_1 en fonction du temps	21
5.8	Couple selon θ_2 en fonction du temps	21
5.9	Couple selon θ_3 en fonction du temps	22
5.10	Datasheet du moteur mdp du robot	22

Abbréviations

DDL	Degrés De Liberté
INRIA	Institut Nationale de Recherche en Informatique et Automatique
IRCICA	Institut de Recherche en Composants logiciels et matériels pour l'Information et la Communication Avancée
L2EP	Laboratoire d'Electrotechnique et d'Electronique de Puissance
PID	Proportionnelle Intégrale Dérivative
HDR	Habilité à Dirigé des Recherches

Chapitre 1

Introduction

Ce projet fût supervisé par M. Giraud, HdR de l'équipe Mint de l'IRCICA. L'IRCICA est une institut qui à pour but de regrouper plusieurs équipes de différents laboratoires. L'équipe Mint regroupe des membres de deux laboratoires : l'INRIA et le L2EP. Ils travaillent tous sur les interactions Homme-machine qui est un thème de recherche au cœur du 21e siècle. En effet, l'essor de la robotique dans la vie de tous les jours a soulevé des questionnements sur l'interaction entre l'Homme et les systèmes automatisés.



FIGURE 1.1 – Photo du robot Hexapod à 6 DDL

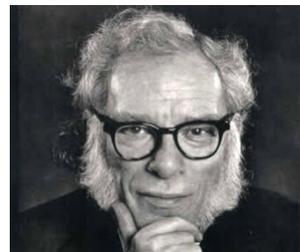
Le projet actuel fait suite à différents travaux antérieures dans le cadre du projet STIMTAC qui vise à développer des stimulateurs tactiles. Tout d'abord, un premier travail a été réalisé par Tao Zheng sur la reproduction de sensations de toucher virtuel [14] c'est-à-dire comment via à un robot, on peut reproduire une forme ou un objet virtuel. Ensuite, un deuxième travail a été effectué par Tanh Hung sur l'étude et la conception du robot Hexapod à 6 DDL [8]. Ce robot a été entièrement construit par le laboratoire. C'est une plateforme reliée à 6 bras qui

sont chacun piloté par un moteur à courant continu. La présence des 6 bras sur ce robot permet à la plateforme de se déplacer aussi bien en translation qu'en rotation semblable à la plateforme de Stewart [7]. C'est pour cela que le robot possède 6 degrés de liberté [2][11][15][6]. Sur ce robot, on peut le commander soit dans l'espace articulaire (avec la position des moteurs) ou dans l'espace cartésien c'est-à-dire avec la position en x,y,z de la plateforme. Le travail de Tanh Hung fut de piloter le robot en boucle ouverte dans l'espace articulaire. Mais on s'est rendu compte que cette méthode est instable à cause du couplage entre les axes. En effet, la mise en mouvement d'un bras provoque un mouvement au niveau de la plateforme mais implique aussi des mouvements des autres bras. Ainsi, il existe donc un couplage entre les différents bras.

Mon travail consiste donc à commander le robot dans l'espace cartésien afin de permettre de découpler les axes. Le but étant de calculer les forces nécessaires au niveau des moteurs pour mettre en mouvement la plateforme. Le projet se décompose donc en plusieurs étapes. La première étape est l'étude bibliographique du projet qui permet de connaître où en ai la recherche dans le domaine. Ensuite, la deuxième étape consiste en l'élaboration des équations du robot au niveau de la cinématique et de la dynamique [3]. Puis, à partir de ces équations, il faut réaliser la simulation sur Matlab Simulink. Ensuite, vient l'étape du contrôle, par la mise en place du contrôleur afin de générer les forces de références au niveau des moteurs.

La différence entre l'être humain et le robot n'est peut-être pas aussi significative que celle qui oppose l'intelligence et la bêtise.

Les cavernes d'acier, Isaac Asimov (trad. Jacques Brécart), éd. J'ai Lu, 1975 (ISBN 2-277-12404-4), p. 58



Isaac ASIMOV, 1920-1992, source : <http://magazine.bsbtheatre.com/>

Chapitre 2

Outils pour l'ingénieur

2.1 Modélisation du système

2.1.1 Matrices de translations et rotations

En dimension 3, on peut définir les matrices homogènes T_u qui permet de faire une translation ainsi qu'une rotation avec une seule matrice.

$$H = \begin{pmatrix} R_{33} & T_3 \\ 0 & 1 \end{pmatrix} \quad (2.1)$$

On doit compléter la matrice par des 0 et des 1 afin d'obtenir une matrice 4×4 carrée. Ainsi, on devra aussi compléter les vecteurs par un 1 afin la multiplication matricielle puisse opérer.

Si on remplace la matrice de rotation par la matrice identité, on obtient alors une matrice homogène de translation [9].

$$T_u = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

On peut ensuite définir, en remplaçant la matrice de translation par des 0, la rotation d'angle θ par rapport à l'axe X, Y, et Z d'un repère quelconque noté respectivement R_X , R_Y et R_Z avec les matrices homogènes suivantes :

$$R_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

$$R_Y = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

$$R_Z = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

2.1.2 Les angles de Tangage, Roulis et Lacet

Le plateau mobile du robot peut être orienté. On définit alors trois angles qui permettent de définir cette orientation selon le repère R_C standard.

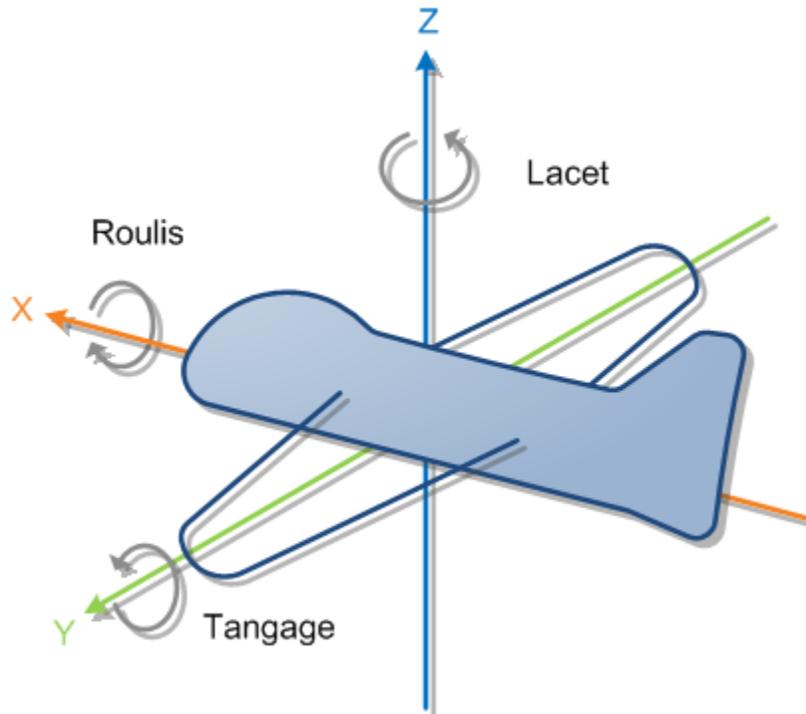


FIGURE 2.1 – Angles de Tangage, Roulis et Lacet, source : floconcept.fr

L'angle Roulis est selon X, Tangage est selon Y et Lacet est selon Z.

2.1.3 Changement de variable avec la tangente de l'arc moitié

Il arrive souvent d'obtenir une équation avec des $\sin(\theta)$ et des $\cos(\theta)$. Il est donc utile d'utiliser ce changement de variable afin d'isoler θ .

On pose :

$$t = \tan\left(\frac{\theta}{2}\right) \quad (2.6)$$

$$\begin{cases} \cos(\theta) = \frac{1-t^2}{1+t^2} \\ \sin(\theta) = \frac{2t}{1+t^2} \end{cases} \quad (2.7)$$

Cela permet donc d'obtenir une équation en t avec une seule variable.

Chapitre 3

Modélisation du robot

3.1 Repèrages et paramètres du robot

Les repèrages du robot permet de poser les "fondations" de la suite du travail. Sans ces repèrages, la cinématique du robot sera incompréhensible par autrui. Il est donc très important de spécifier à chaque fois le repère dans lequel on réalise les calculs.

Voici un schéma du robot associé à ses repères et ses différents paramètres :

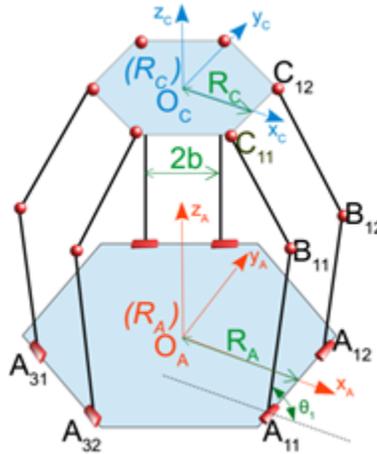


FIGURE 3.1 – Schéma du robot hexapode avec ses repères et ses paramètres

3.1.1 Repèrages du robot

Les différents repères du robot sont :

- Repère du plateau fixe : $(R_A) O_A X_A Y_A Z_A$
- Repère du plateau mobile : $(R_C) O_C X_C Y_C Z_C$
- Repère local fixe de pivot : $(R_{Aij}) O_{Aij} X_{Aij} Y_{Aij} Z_{Aij}$

3.1.2 Paramètres du robot

Les différents paramètres du robot sont :

- Longueur des bras : $A_{ij} B_{ij} = L_A$
- Longueur des bras : $B_{ij} C_{ij} = L_C$
- Rayon : R_A
- Rayon : R_C

3.1.3 Entrées et sorties du robot

Les variables de commandes sont les différents angles θ_{ij} du robot. Et les variables d'entrée sont la position de O_C dans le repère (R_A) et ainsi que les angles de Tangages, Roulis et Lacet (définie plus tard) du plateau par rapport au repère (R_C) .



3.2 Cinématique inverse du robot

Elle permet de déterminer les angles $(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32})$ des moteurs en fonction des coordonnées du plateau $(x, y, z, \alpha, \beta, \gamma)$ du robot.

3.2.1 Coordonnées de A_{ij} dans R_A

D'après le schéma, on peut déduire que :

$$(A_{11})_{R_A} = \begin{pmatrix} R_A \\ -b \\ 0 \end{pmatrix} \quad (3.1)$$

et

$$(A_{12})_{R_A} = \begin{pmatrix} R_A \\ b \\ 0 \end{pmatrix} \quad (3.2)$$

Puis, pour déterminer les coordonnées des autres points, on utilise les relations suivantes :

$$(A_{2j})_{R_A} = R_Z\left(\frac{2\pi}{3}\right)(A_{1j})_{R_A} \quad (3.3)$$

$$(A_{3j})_{R_A} = R_Z\left(\frac{4\pi}{3}\right)(A_{1j})_{R_A} \quad (3.4)$$

3.2.2 Coordonnées de C_{ij} dans R_C

De la même façon, on peut déduire que :

$$(C_{11})_{R_C} = \begin{pmatrix} R_C \\ -b \\ 0 \end{pmatrix} \quad (3.5)$$

et

$$(C_{12})_{R_C} = \begin{pmatrix} R_C \\ b \\ 0 \end{pmatrix} \quad (3.6)$$

Puis, pour déterminer les coordonnées des autres points, on utilise les relations suivantes :

$$(C_{2j})_{R_C} = R_Z\left(\frac{2\pi}{3}\right)(C_{1j})_{R_C} \quad (3.7)$$

$$(C_{3j})_{R_C} = R_Z\left(\frac{4\pi}{3}\right)(C_{1j})_{R_C} \quad (3.8)$$

3.2.3 Coordonnées de C_{ij} dans R_A

Il faut associer les rotations relatifs aux angles tangage α , roulis β et lacet γ . Puis la translation $O_C O_A$.

On utilise la relation suivante :

$$(C_{ij})_{R_A} = R_X(\alpha)R_Y(\beta)R_Z(\gamma)T_{O_C O_A}(C_{ij})_{R_C} \quad (3.9)$$

3.2.4 Coordonnées de C_{ij} dans R_{Aij}

On utilise la relation suivante :

$$(C_{ij})_{R_{Aij}} = T_{O_A A_{ij}}(C_{ij})_{R_A} \quad (3.10)$$

3.2.5 Coordonnées de B_{ij} dans R_{Aij}

D'après le schéma, on a :

$$(B_{ij})_{R_{Aij}} = \begin{pmatrix} L_A \cos \theta_{ij} \\ 0 \\ L_A \sin \theta_{ij} \end{pmatrix} \quad (3.11)$$

En sachant qu'on ne connaît pas les valeurs de θ_{ij} qu'on va déterminer juste après.

3.2.6 Calcul des angles θ_{ij}

Tous les calculs sont réalisés dans le repère R_{Aij} .

D'après la mécanique du robot, on a :

$$B_{ij}C_{ij} = L_C \quad (3.12)$$

On obtient donc une équation de ce type :

$$E_{ij} \cos(\theta_{ij}) + F_{ij} \sin(\theta_{ij}) + G_{ij} = 0 \quad (3.13) \quad \text{🗨️}$$

Le problème étant la présence des $\cos(\theta)$ et des $\sin(\theta)$ dans la même équation. On va donc utiliser le changement de variable avec la tangente de l'arc (cf equation 2.6 et 2.7).

Ce qui donne :

$$E_{ij} \left(\frac{1 - t_{ij}^2}{1 + t_{ij}^2} \right) + F_{ij} \left(\frac{2t_{ij}}{1 + t_{ij}^2} \right) + G_{ij} = 0 \quad (3.14)$$

On réordonne :

$$(G_{ij} - E_{ij})t_{ij}^2 + (2F_{ij})t_{ij} + (G_{ij} + E_{ij}) = 0 \quad (3.15)$$

Par résolution, on a donc :

$$t_{ij} = \frac{-F_{ij} \pm \sqrt{(E_{ij}^2 + F_{ij}^2 - G_{ij}^2)}}{G_{ij} - E_{ij}} \quad (3.16)$$

La présence du \pm caractérise la coude vers le haut ou vers le bas, pour une même position du plateau [1].

3.3 Cinématique direct du robot

Elle permet de déterminer les coordonnées $(x, y, z, \alpha, \beta, \gamma)$ du plateau en fonction des angles $(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32})$ des moteurs.

3.3.1 Décomposition des variables

NB : Attention, cette sous partie est une étape essentielle dans la suite du projet.

La cinématique direct du robot est beaucoup plus complexe à déterminer sur un robot parallèle.

En posant,

$$\begin{cases} \theta_i &= \frac{\theta_{ij} + \theta_{i(j+1)}}{2} \\ d\theta_i &= \frac{\theta_{ij} - \theta_{i(j+1)}}{2} \end{cases} \quad (3.17)$$

Cela permet de travailler séparément la translation et la rotation du plateau. Les θ_i impliquent des translations du plateau et les $d\theta_i$ entraînent des rotations au niveau du plateau.

Or, la cinématique direct en translation est plus facile à obtenir que la cinématique direct en rotation. C'est la même méthode que la robot delta (robot intégré dans les imprimantes 3D) [1]. Ce qui m'a permis de poursuivre le travail de modélisation.

On s'intéressera donc uniquement aux mouvements de translation pour la suite du projet.

3.3.2 Cinématique direct en translation du robot

On détermine les coordonnées de $(B_i)_{RA}$. Puis, on applique la translation $T_{C_i R_C}$. Ensuite, on détermine l'intersection des 3 sphères de rayon L_C . Elles se couperont en un point, le point O_C .

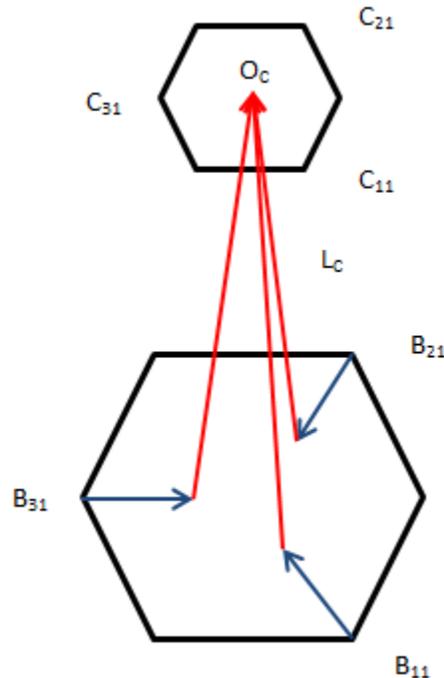


FIGURE 3.2 – Détermination de la cinématique direct

Pour cela, j'ai utilisé un algorithme sous matlab qui calcule le point d'intersection de trois sphères.



3.4 Dynamique inverse du robot en translation

Grâce à la cinématique inverse, on a actuellement le modèle inverse du robot sous cette forme :

$$\begin{cases} \theta_{11} = f_{11}(x_{O_C}, y_{O_C}, z_{O_C}) \\ \theta_{12} = f_{12}(x_{O_C}, y_{O_C}, z_{O_C}) \\ \theta_{21} = f_{21}(x_{O_C}, y_{O_C}, z_{O_C}) \\ \theta_{22} = f_{22}(x_{O_C}, y_{O_C}, z_{O_C}) \\ \theta_{31} = f_{31}(x_{O_C}, y_{O_C}, z_{O_C}) \\ \theta_{32} = f_{32}(x_{O_C}, y_{O_C}, z_{O_C}) \end{cases} \quad (3.18)$$

On a donc :

$$\theta_{ij} = f_{ij}(X) \quad (3.19)$$

Pour déterminer la dynamique du système, il suffit donc de faire des dérivés partiels de f_{ij} [8].

On aura donc :

$$\theta \dot{i} j = Jv \quad (3.20)$$

Avec

- J : la jacobienne du système
- v : la vitesse du plateau mobile

3.5 Dynamique directe du robot en translation

Grâce à la jacobienne du système, on peut déterminer la jacobienne inverse du système afin de pouvoir disposer de la dynamique direct du robot en translation.

On aura donc :

$$\theta \dot{i} j = J^{-1}v \quad (3.21)$$

Avec

- J^{-1} : la jacobienne inverse du système
- v : la vitesse du plateau mobile

3.6 Modélisation sous Matlab simulink

3.6.1 Equation dynamique du robot

Le robot 6 DDL étudié est une plateforme en mouvement raccordée par des tiges qui sont pilotées par des moteurs. On peut donc modéliser le robot par une masse en mouvement. De plus, nous voulons étudier le problème au niveau des forces que l'on peut appliquer sur cette masse.

On peut obtenir l'équation dynamique du robot par application du Principe fondamental de la dynamique :

$$\vec{F} + \vec{P} = Masse \times \frac{dv}{dt} \quad (3.22)$$

Avec

- : La force appliquée à l'effecteur
- : Le Poids de l'effecteur
- v : vitesse de l'effecteur
- Masse : Masse de l'effecteur

En faisant une projection selon les axes x , y , z , on obtient 3 équations de la dynamique du robot [5][16]. (On choisira l'axe des z vers le haut)

$$\begin{cases} F_x & = & Masse \times \frac{dv_x}{dt} \\ F_y & = & Masse \times \frac{dv_y}{dt} \\ F_z - Masse \times g & = & Masse \times \frac{dv_z}{dt} \end{cases} \quad (3.23)$$

3.6.2 Représentation sous Matlab simulink

Le schéma suivant représente le principe de la modélisation sur Matlab Simulink du robot. Pour modéliser la cinématique inverse ou directe ainsi que la dynamique inverse ou direct, on choisit alors d'utiliser des blocs fonctions sur Matlab Simulink.

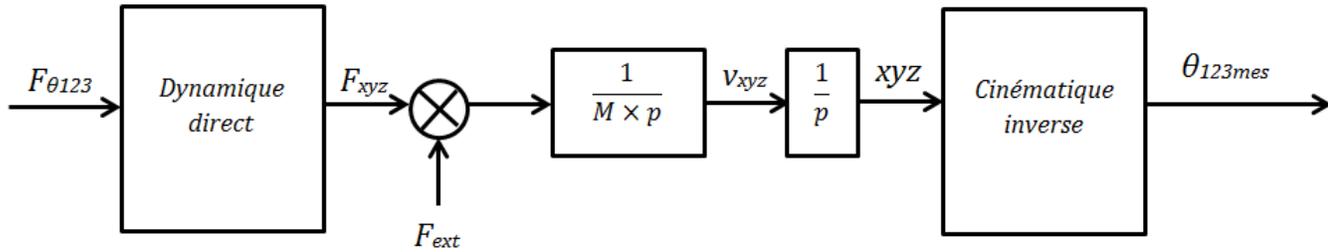


FIGURE 3.3 – Schéma de modélisation du robot Hexapod

Les entrées du modèle sont les forces selon θ au niveau des moteurs. Mais en réalité, on commandera les moteurs en couple. Puis, ces forces selon θ sont transformées en forces selon xyz par la dynamique direct. La masse est ainsi déplacée par le principe de la dynamique.

On récupère la vitesse de la masse qu'on intègre afin d'obtenir la position de la masse en xyz . La présence de l'intégrale nous permet de modéliser les conditions initiales du mouvement. Ensuite, cette position selon xyz est transformée en position selon θ par la cinématique inverse du robot. On récupère donc en sortie la position selon θ [12].

Chapitre 4

Contrôle du robot

4.1 Etude en boucle ouverte

Le schéma suivant représente le robot en boucle ouverte. Afin de contrôler le robot, il est important de bien étudier le système en boucle ouverte. Cette étude nous permettra de trouver les gains du correcteur PID.

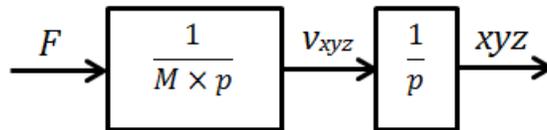


FIGURE 4.1 – Schéma de modélisation de la boucle ouverte du robot Hexapod sous Simulink

Pour étudier le comportement du robot en boucle ouverte, nous allons insérer un step en entrée au niveau du moteur de θ_1 et observer ce qui se passe au niveau des autres moteurs.

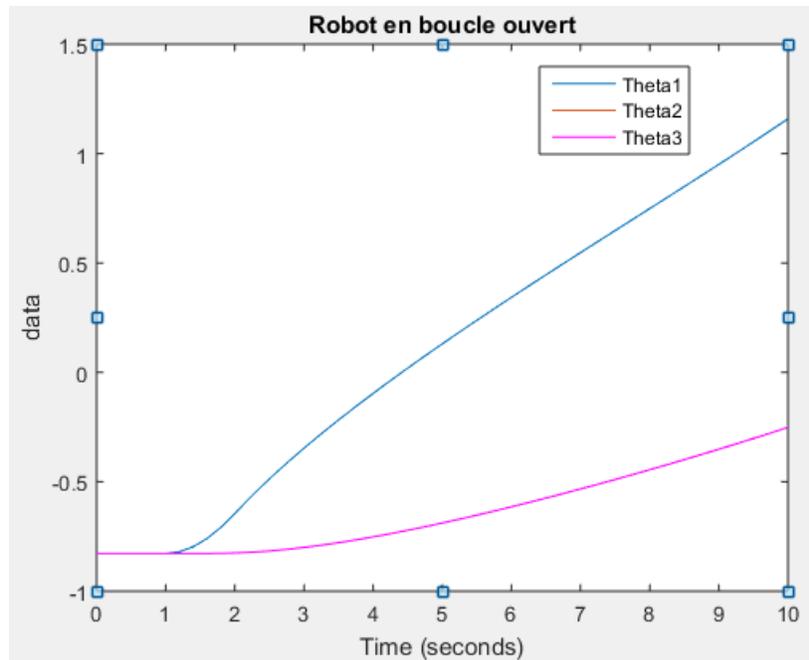


FIGURE 4.2 – Etude du robot en boucle ouverte

Ce graphique nous montre bien la présence d'un couplage entre les différents bras du robot. En effet, en insérant uniquement une force selon θ_1 , on observe que tous les autres bras du robot bouge aussi car les moteurs de θ_2 et θ_3 sont en mouvement.

Il est donc nécessaire de découpler les axes du robot. C'est donc pour cela que nous allons travailler dans l'espace cartésien afin que tous les moteurs fonctionnent ensemble et non indépendamment [10][13].

4.2 Modélisation du contrôle

L'objectif du projet est de contrôler le robot dans l'espace cartésien et non l'espace articulaire. Ainsi, il est donc important de repasser dans l'espace cartésien avant d'insérer le contrôleur PID.

Voici le schéma de contrôle du robot :

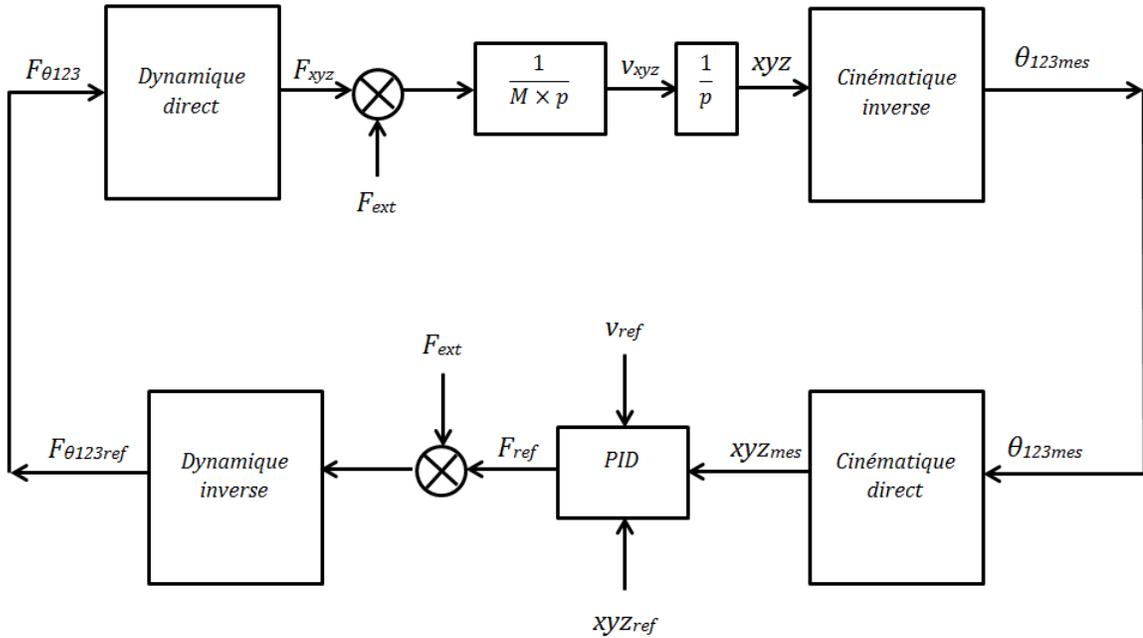


FIGURE 4.3 – Schéma de modélisation du robot Hexapod avec la boucle de contrôle

En entrée, on récupère la sortie capteur du robot, θ à-dire les positions θ des moteurs. Puis, on repasse dans l'espace cartésien par la cinématique direct. De là, on réalise le PID avec les positions de références et les vitesses de références.

Puis, on obtient ainsi les forces selon xyz de référence. On repasse dans le domaine articulaire par le biais de la dynamique inverse. Les forces selon θ sont donc les sorties du contrôle.

4.3 Design du PID

4.3.1 Représentation d'Etat

La représentation d'état du système est un outil de modélisation en Automatique qui permet ensuite de contrôler le système assez facilement. Ici, elle nous permettra uniquement de déterminer les valeurs du PID.

En rapport avec la structure du PID, nous choisirons comme vecteur d'état :

$$x = \begin{pmatrix} \dot{x} \\ x \\ \int x \end{pmatrix} \quad (4.1)$$

On peut donc écrire la représentation d'état sous cette forme :

$$\{ \dot{x} = A_d x + B_d u \quad (4.2)$$

On ne s'occupe pas des sorties du système car on ne va pas contrôler le système avec cette représentation. Elle nous sert juste à déterminer les valeurs du PID.

Avec :

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

$$B_d = \begin{pmatrix} \frac{1}{\text{Masse}} \\ 0 \\ 0 \end{pmatrix}$$

On a donc bien une représentation d'état du système.

4.3.2 Régulateur PID

Le régulateur PID offre de bonnes performances tout en ayant une facilité de mise en place.

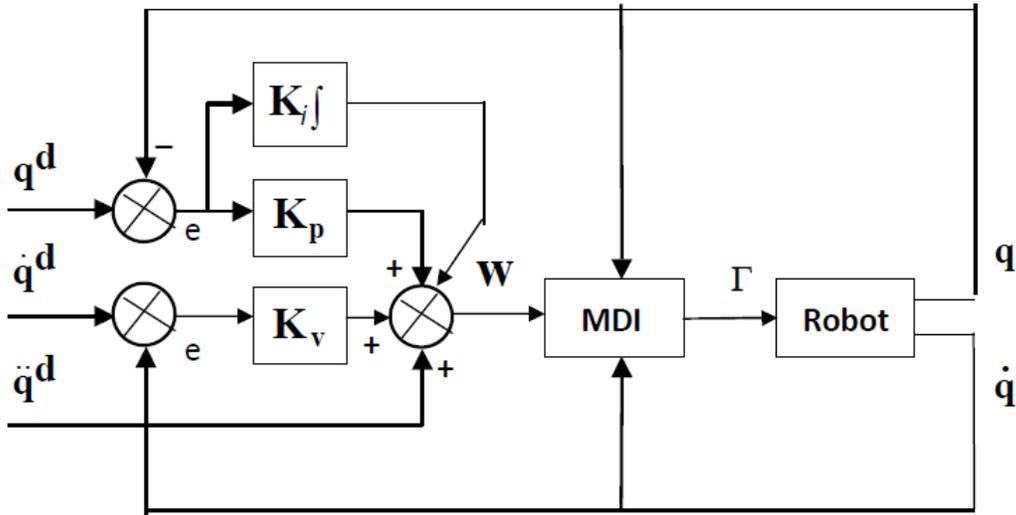


FIGURE 4.4 – Schéma structurel du régulateur PID

On introduit donc l'erreur entre la position désirée et la position mesurée, ainsi que l'erreur entre la vitesse désirée et la vitesse mesurée. Le K_p est le gain pour l'erreur de position, K_i est le gain pour l'intégral de l'erreur de position et le K_d est le gain pour l'erreur de vitesse.

Il est donc important de réaliser le PID à la main et non pas d'utiliser le block PID de Matlab.

La commande sera donc de la forme suivante :

$$u = K_p(x_{des} - x_{mes}) + K_i(x_{des} - x_{mes}) + K_d(v_{des} - v_{mes}) \quad (4.3)$$

Cette structure permet à la fois de commander en position et en vitesse le robot.

4.3.3 Placement de pôles

Pour le design du PID, on choisira la technique de placement de pôles. La bibliographie nous indique qu'il faut choisir des pôles de mêmes valeurs et inférieur à la pulsation propre mécanique du robot. Cette pulsation propre peut être mesurée.

On aura donc comme polynôme caractéristique :

$$(s + \omega)^3 \quad (4.4)$$

avec $\omega < \omega_p$

Détermination de la pulsation propre du robot

Dans la bibliographie, on a vu qu'il faut choisir comme pôles du système : $\omega < \omega_p$ [15][4]. Un ingénieur du laboratoire a déterminé le diagramme de bode du robot.

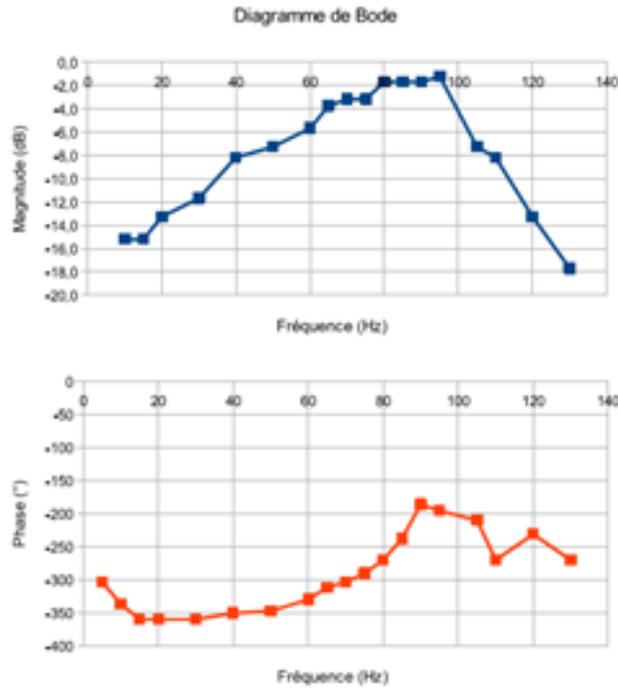


FIGURE 4.5 – Détermination de la pulsation propre du robot

Ce diagramme nous renseigne sur la nature du filtre du robot. C'est un filtre passe bande. La fréquence qui laisse passer le plus d'onde est la fréquence propre du robot. On peut la lire au alentour de 90 Hz. De plus, le robot n'est pas totalement rigide par la présence du jeu au niveau des articulations du robot. Il se comporte donc comme un système du second ordre. En effet, le jeu induit des petites oscillations caractéristiques d'un système du second ordre.

Choix des pôles

Pour obtenir une erreur de l'ordre du millimètre, on doit donc choisir des gains du PID pouvant aller jusqu'à 10^3 afin d'avoir une force de l'ordre du Newton. Ainsi, j'ai donc choisi comme pôle du système : $-2\pi i \times 3.5$.

Avec la commande de Maltlab : `Place(A,B,pôle)`, on obtient ainsi les gains PID du contrôleur.

$$K = (32.98 \quad 724.92 \quad 5.3 \cdot 10^{-3})$$

$$x = \begin{pmatrix} \dot{x} \\ x \\ \int x \end{pmatrix} \quad (4.5)$$

Or d'après la forme du vecteur d'état x , on en déduit que la première colonne du vecteur K est le gain K_d , la deuxième colonne le gain K_p et la troisième colonne le gain K_i . On a donc réglé notre contrôleur PID.

4.4 Proposition de contrôle des mouvements de rotation de la plateforme

Jusqu'à maintenant, le robot peut se déplacer avec des mouvements de translations. Afin de pouvoir reproduire des formes ou objets avec la plateforme, il est donc nécessaire de pouvoir réaliser des mouvements de rotation. Comme il est très difficile de connaître le modèle direct en rotation du robot, l'ajout d'un capteur gyroscopique qu'on placerait en dessous de la plateforme afin de ne pas gêner le capteur tactile, nous permettrait de connaître les angles instantanés de la plateforme.

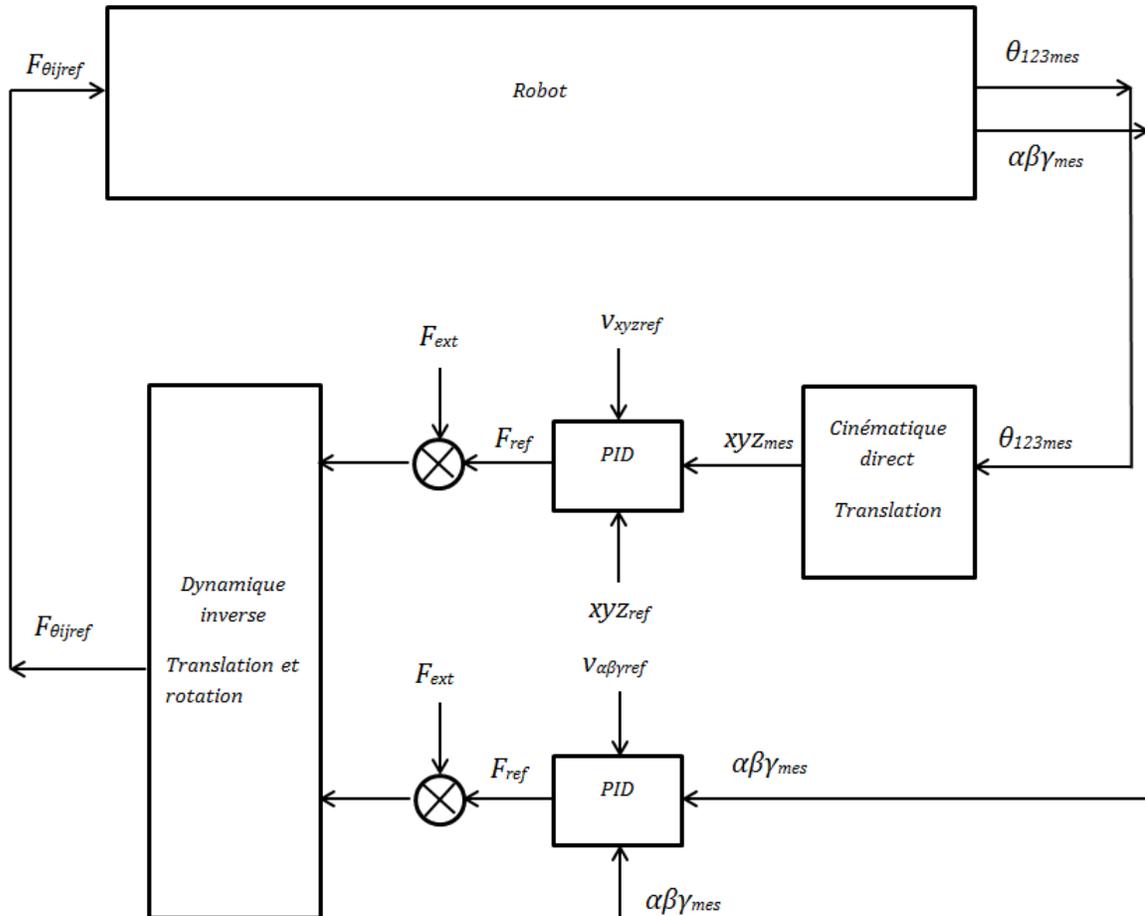


FIGURE 4.6 – Schéma de description du contrôle en rotation et translation de la plateforme

On pourrait alors effectuer un PID, puis utiliser le modèle inverse en rotation et translation afin de générer les forces de références au niveau des moteurs.

Chapitre 5

Résultats de simulation en boucle fermée

L'objectif de la simulation, est de réaliser un cercle avec la plateforme pour un z donné. Les signaux de références pour x et y sont donc des sinusoïdes.

5.1 Trajectoire en x et y

Après avoir simulé le modèle, voici la réponse de x en fonction de y .

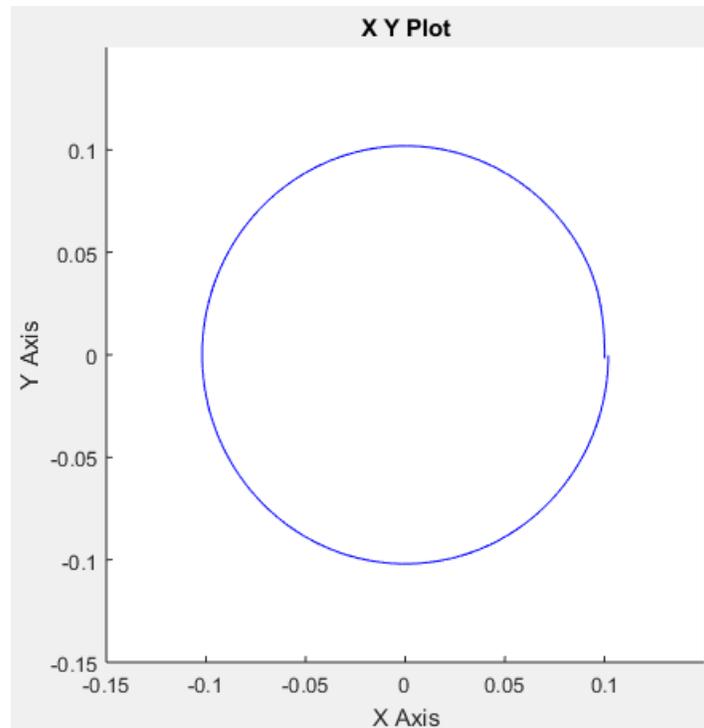


FIGURE 5.1 – Réponse du système de x en fonction de y

Il y a un léger décalage qui est dû à la mise en mouvement de la masse au départ.

5.2 Trajectoires selon x, y et z

Voici la réponse du système selon X :

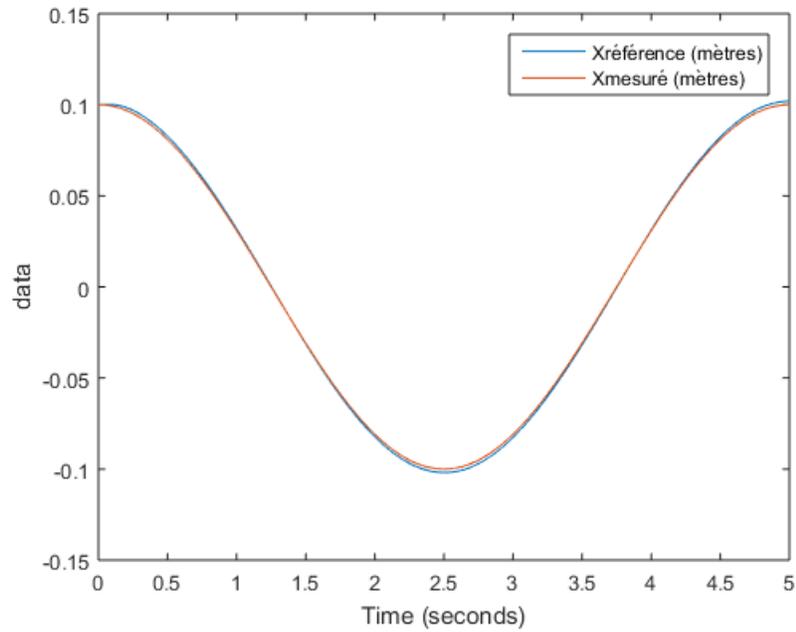


FIGURE 5.2 – $X_{référence}$ et X_{mesure} en fonction du temps

Pour mieux évaluer les performances du contrôle, le graphique représentant l'erreur entre le signal mesuré et le signal de référence nous donne plus d'informations.

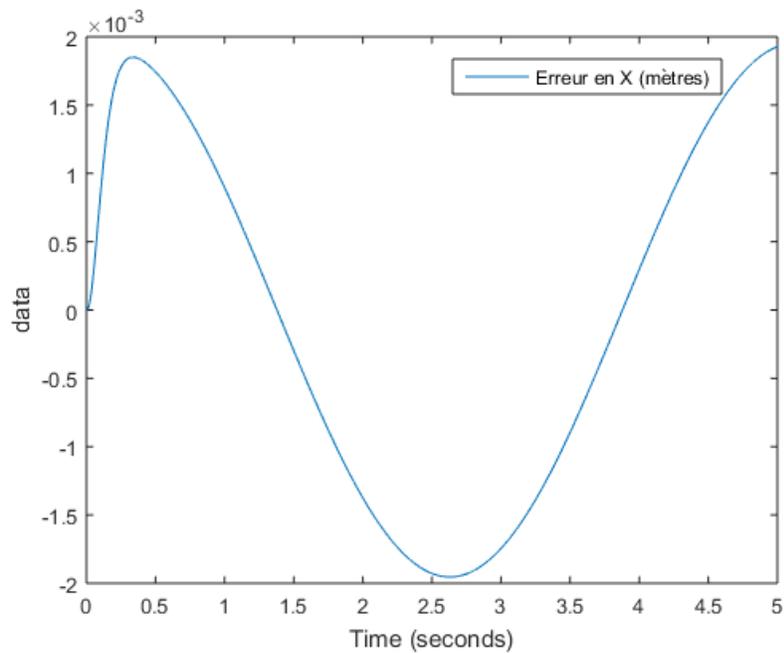


FIGURE 5.3 – Erreur selon X en fonction du temps

Voici la réponse du système selon Y :

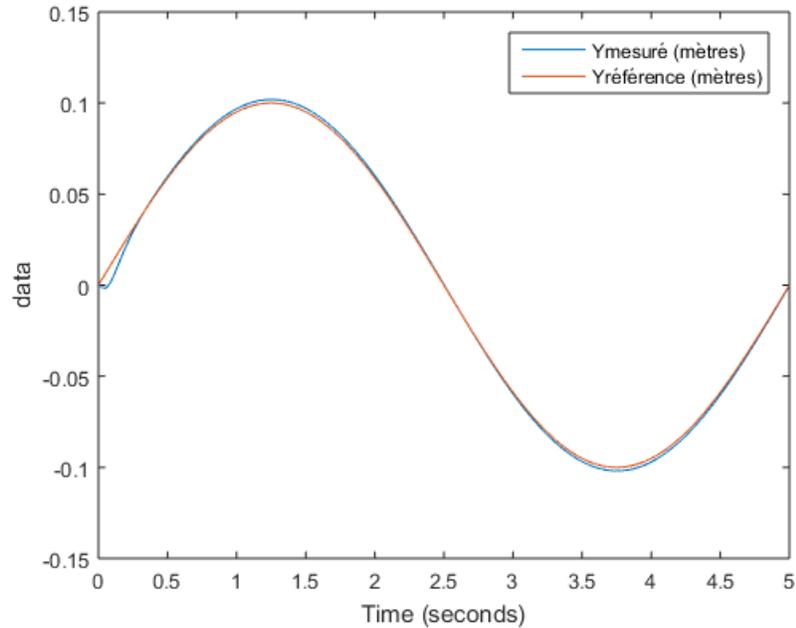


FIGURE 5.4 – $Y_{référence}$ et Y_{mesure} en fonction du temps

Pour mieux évaluer les performances du contrôle, j'ai tracé l'erreur entre le signal mesuré et le signal de référence.

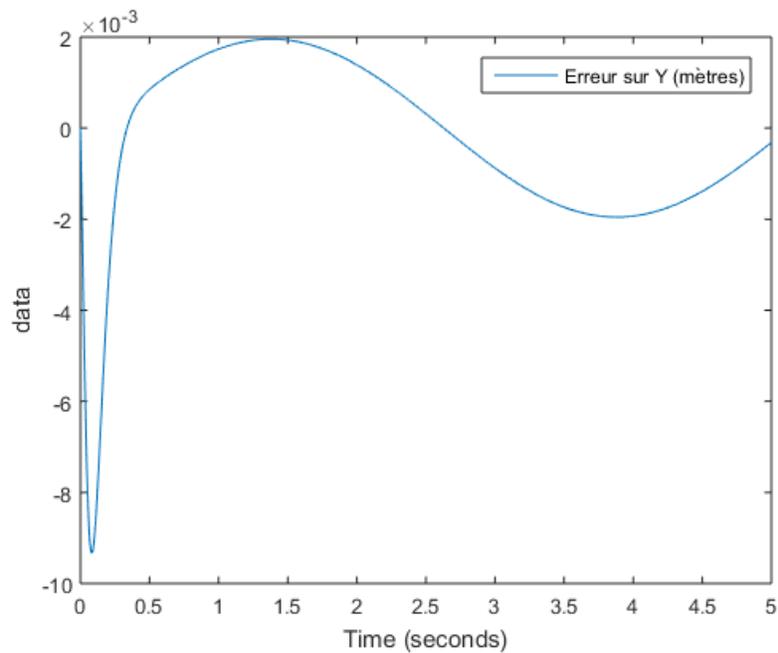


FIGURE 5.5 – Erreur selon Y en fonction du temps

La position de référence selon Z est fixe.

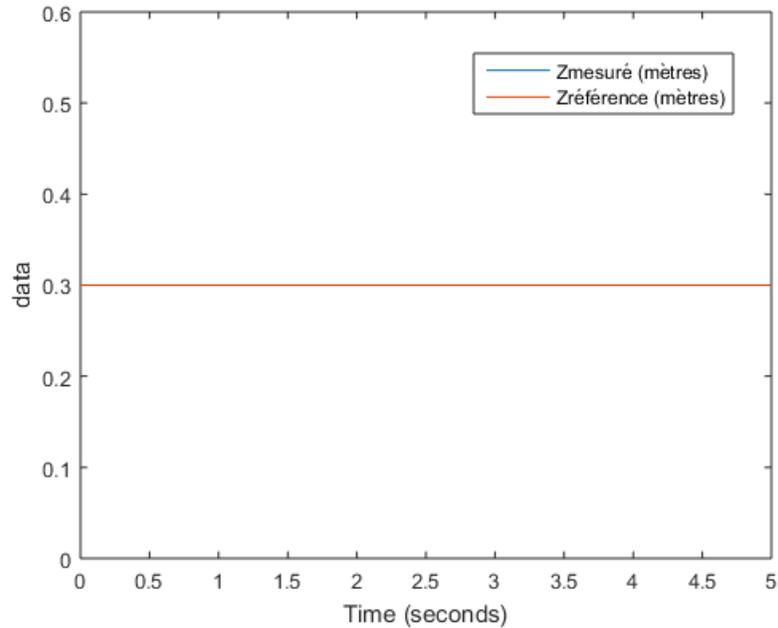


FIGURE 5.6 – $Z_{référence}$ et Z_{mesure} en fonction du temps

5.3 Couples des moteurs

A partir des forces de références à la sortie des PID, on peut obtenir les couples moteurs de références par la relation suivante :

$$C = F \times R_{moteur} \quad (5.1)$$

Avec

- C : Couple au niveau du moteur
- F : Force exercé par le moteur
- R_{moteur} : Rayon du moteur

Voici le couple de référence selon les trois angles :

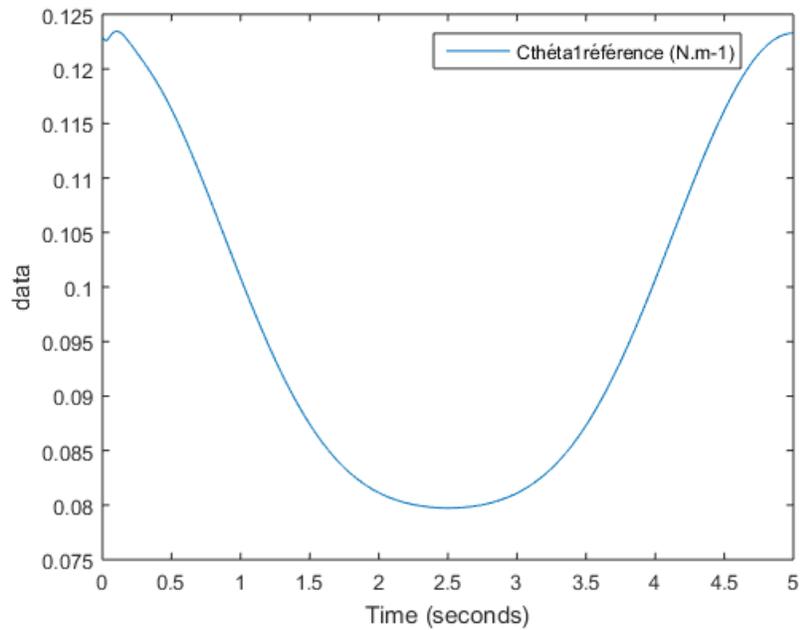


FIGURE 5.7 – Couple selon θ_1 en fonction du temps

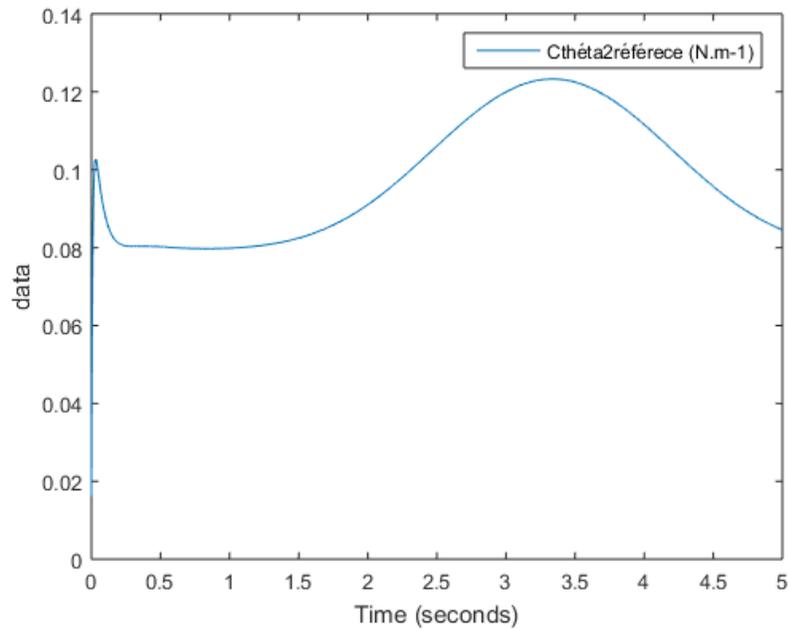


FIGURE 5.8 – Couple selon θ_2 en fonction du temps

Les couples sont exprimés pour une association de deux moteurs. En effet, on a auparavant moyenné les positions des moteurs afin de séparer les mouvements de translations et de rotations. Il faut donc diviser par deux la valeurs des couples moteurs.

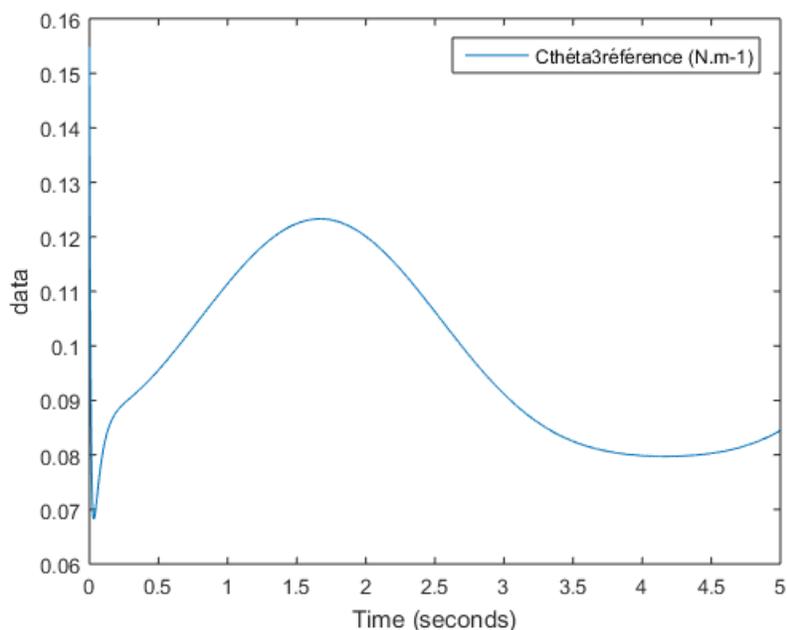


FIGURE 5.9 – Couple selon θ_3 en fonction du temps

Les couples sont-ils admissibles par les moteurs ?

maxon motor

Tension d'alimentation (Ua)	V	12	24
Vitesse au courant In	tr/mn	5565	3493
Couple au courant In	mNm	77.70	113.00
Courant max permanent (In)	mA	4000	2150
Vitesse à vide à Ua à +/- 10%	tr/mn	6222	4303
Courant à vide à +/- 50%	mA	245	93
Couple de démarrage à Ua	mNm	698.00	611.00
Courant de démarrage à Ua	mA	35920	11600
Constante de couple	mNm/A	19.40	52.50
Constante de vitesse	tr/mn/V	491	182
Pente vitesse/couple	tr/mn/mN	8	7
Vitesse limite	tr/mn	8200	8200
Puissance utile max. à Ua	W	114	68.9
Rendement maximum	%	81	86
Constante de temps	ms	6	5
Inertie	gcm ²	65.5	69.6
Résistance aux bornes	Ohm	0.334	2.07
Inductivité	mH	0.09	0.62
Résistance thermique	K/W	6.2	6.2
Résistance thermique Rotor/Boîtier	K/W	2	2

FIGURE 5.10 – Datasheet du moteur mdp du robot

Le couple maximal ne dépasse pas $0.14N.m^{-1}$. Ce qui donne en réalité $0.07N.m^{-1}$. Or, d'après la datasheet, cette valeur est bien en dessous de la valeur maximale du couple moteur au courant In ($0.07N.m^{-1}$).

Chapitre 6

Conclusion et perspectives

L'objectif de ce projet était de réaliser la modélisation et le contrôle en simulation du robot Hexapod. On connaît donc dorénavant les différences entre les coordonnées articulaires et cartésiennes mais aussi les vitesses articulaires et cartésiennes. De plus, on peut aussi mettre en mouvement le robot dans l'espace d'état via le contrôleur PID qui a été mis en place.

Ce projet m'a beaucoup apporté car il m'a permis de développer mes compétences sur les robots parallèles qui est un domaine complexe en robot. De plus, la présence d'un projet dans le Master 2 Smart, nous prépare au métier de chercheur. En effet, j'ai pu découvrir qu'il faut persévérer même si les résultats ne sont pas concluants en approfondissant la bibliographie. J'ai pu aussi m'apercevoir que la rédaction joue un rôle important dans le métier de chercheur. Car, en effet, cela permet de transmettre des connaissances mais aussi de montrer la qualité du travail accompli. Il est donc important d'en tenir compte.

Il reste donc à tester le modèle de contrôle sur le robot par l'implémentation en temps réel sur des DSP. Ce test nous permettra de savoir si le modèle de cinématique ou dynamique du robot est fiable. Mais aussi, si le contrôle dans l'espace cartésien conduit à des résultats plus stables que dans l'espace articulaire. Puis, on ajoutera une interface tactile au dessus de la plateforme. Le but étant d'interagir avec la plateforme afin de créer une sensation de toucher d'objet ou de forme virtuelle.

Deux routes ont divergé dans un bois, et moi, j'ai pris le moins emprunté, et cela a fait toute la différence.



Robert FROST, 1874-1963

Bibliographie

- [1] The Delta Parallel Robot : Kinematics Solutions.
- [2] Les robots parallèles - Merlet - Hermes science publications - Livres.
- [3] *Modélisation cinématique et dynamique des systèmes poly-articulés à chaînes ouvertes ou fermées. Cas des robots parallèles.* PhD thesis.
- [4] Modélisation, identification et commande des robots - W. Khalil, E.... - Librairie Eyrolles.
- [5] Saliha Bellakehal, Nicolas Andreff, Youcef Mezouar, and Mohamed Tadjine. Commande Vision/Force de robots parallèles. *Journal Européen des Systèmes Automatisés*, 44(8) :887–911, 2010.
- [6] Ilian A. Bonev. Les robots parallèles : de la recherche vers les applications. *Journées Nationales de la Recherche en Robotique, Obernai, France, 2007.*
- [7] Roger Boudreau and Clément M. Gosselin. *La synthèse d'une plate-forme de Gough-Stewart pour un espace atteignable prescrit*, volume 36. March 2001.
- [8] Tanh Hung. Compte rendu Post-doc sur la conception, modélisation et contrôle d'un robot parallèle à 6ddl, June 2015.
- [9] Wisama Khalil and Etienne Dombre. *Modeling Identification and Control of Robots.* CRC Press, 2002.
- [10] Oussama KHATIB. The operational Space Famework. 1993.
- [11] Samir Lahouar. *Contribution à la modélisation et à la planification de trajectoires de robots parallèles évitant les singularités et les collisions.* PhD thesis, Poitiers, 2008.
- [12] Philippe Lelias. *Modélisation, identification et commande dynamique d'un robot d'architecture parallèle.* PhD thesis.
- [13] Flavien Paccot. *Contribution à la commande dynamique référencée capteurs de robots parallèles.* PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2009.
- [14] ZHENG Tao. *Conception et Contrôle d'un périphérique dédié à la simulation couplée kinesthésique et tactile.* PhD thesis, 2012.
- [15] Oscar Andrés Vivas. *Contribution à l'identification et à la commande des robots parallèles.* phdthesis, Université Montpellier II - Sciences et Techniques du Languedoc, November 2004.
- [16] Nathaniel Zoso. *Modélisation, simulation et commande d'un robot parallèle plan à câbles sous-actionné.* PhD thesis, Université Laval, 2011.

