RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique

Département d'Automatique

Mémoire en vue de l'obtention
du diplôme d'Ingénieur d'État
en Automatique

Thème :

Robot Parallèle Planaire à 4 câbles :

Modélisation, Optimisation, Commande et Réalisation

Étudié par :

SAADI Yakoub

Dirigé par :

Pr. M. TADJINE

Juin 2014

École Nationale Polytechnique, 10, Avenue Hassen Badi, 16200 El Harrach, Alger.

بسم الله الرَّحْمنِ الرَّحِيمِ

(وَإِذْ قَالَ رَبُّكَ لِلْمَلائِكَةِ إِنِّي جَاعِلٌ فِي الأَرْضِ خَلِيفَةً قَالُوا أَتَجْعَلُ فِيهَا مَنْ يُفْسِدُ فِيهَا وَيَسْفِكُ الدِّمَاءَ وَنَحْنُ نُسَبِّحُ بِحَمْدِكَ وَنُقَدِّسُ لَكَ قَالَ إِنِّي أَعْلَمُ مَا لا تَعْلَمُونَ) البقرة (30)

الاهداء

الى من كل البشرية له تدين

الى سيد الخلق محمد النبي

الى من افديهم عمري و كل السنين

الى الغالـيين امي و ابي

الى استاذي ذو الفـكر الرزين

الذي كان عونـي عند الطلبِ

Remerciements

Avant tout, je remercie الله, le tout puissant, pour m'avoir assisté et armé de patience afin d'accomplir ce modeste travail.

Je tiens à exprimer ma profonde gratitude à mon encadreur de mémoire le Professeur **Mohamed TADJINE** pour m'avoir fait l'honneur de diriger ce PFE, que je remercie pour son encadrement, son aide, ses directives, ses conseils précieux et surtout la confiance qu'il ma accordé.

Mes remerciements s'adressent aussi à Monsieur **abdelouahab ZAATRI**, Professeur à l'université de Constantine et Monsieur **Mohammed BOUAZIZ,** Professeur à l'ENP pour leurs encouragements et précieux conseils.

Enfin, je ne pourrais terminer ces remerciements sans une pensée à l'ensemble de mes enseignants, qui sont à l'origine de tout mon savoir.

Dédicaces

Ce mémoire est dédié à :

- *Mes parentes ;*
- *Mes frères et sœurs ;*
- *Toute ma grande famille de près et de loin ;*
- *Mon ami Mohamed Khaldi ;*
- *Mostefa KERMADI pour son soutien durant toutes mes années de Polytech ;*
- *Toute personne ayant contribué à sa réalisation:*

  Abdelkrim,Faical,Mestapha,Zino,Zaki, Karim(Chouchan) ;
- *Tous les enseignants d'Automatique de L'Ecole Nationale Polytechnique.*

ملخص:

تعتبر الروبوتات المتوازية ذات الكابلات , نوعا جديد من الروبوتات المتوازية التي تتميز باستخدام الكابلات كأداة ربط بين قاعدة الربوت والجسم النهائي بدلا من استخدام الروابط الصلبة .

يهدف هذا العمل المتواضع إلى دراسة مختلف تصاميم الربوت المتوازي ذات الأربع كابلات مثل التصميم الهندسي , الحركي, والديناميكي وكذالك يتطرق هذا العمل إلى كيفية التحكم في هذا النوع الجديد من الروبوتات المتوازية .

أتبعت هذه الدراسة النظرية بتطبيق عملي يهدف إلى تصميم نموذج لربوت مستوى متوازي بأربع كابلات بهدف مساعدة ذوي الاحتياجات الخاصة في الكتابة.

الكلمات الافتتاحية: الروبوتات المتوازية ذات الكابلات, الروبوت المتوازي ذو الأربع كابلات , تصميم , تحكم , تطبيق , إعادة تأهيل حركي.

Résumé :

Les robots parallèles à câbles sont de nouveaux types des robots parallèles qui se caractérisent par l'utilisation de câbles au lieu de vérins.

Ce mémoire d'ingéniorat s'articule sur la modélisation, l'optimisation et la commande d'un robot parallèle planaire à 4 câbles. L'étude théorique du robot est suivie d'une réalisation expérimentale d'un prototype à 4 câbles pour la réhabilitation des bras pour but d'aider les gents des besoins spéciales à l'écriture.

Mots clés : robot parallèle à câbles, robot parallèle à quatre câbles, modélisation, optimisation commande, réalisation, réhabilitation.

Abstract :

The cables-based robots are special types of parallel robots**.** They mainly consist of a fixed base, a mobile platform holding the end-effector , cables connecting the mobile platform to the base and a set of motorized pulleys.

This engineering memory hinges on the robot modeling, optimization and control study of 4 planar cable-based robots. The theoretical study of robot is followed by an experimental realization of a prototype for the arm rehabilitation to help disabled people for writing.

Keywords: Cable-based robot, 4 planar cable-based robots, modeling, optimization, control, realization, rehabilitation.

Table des matières

Table des figures

Introduction générale

Les robots intéressent plusieurs secteurs : industriel, social médical... etc. Il existe plusieurs types de robots : robot marcheur, robot manipulateur, robot à roue... etc. On les classe en deux catégories, robots série et robots parallèles. Les robots à câbles font partie de la deuxième catégorie. Chaque câble est actionné par un moteur et est constamment soumis à une tension qui tire l'effecteur. Le mouvement de l'effecteur qui est relié à l'ensemble des câbles du robot s'effectue suivant une trajectoire définie, donc les différents moteurs sont asservis de manière à ce que cette trajectoire soit respectée. Le robot contrôle la position de l'effecteur à l'intérieur de l'espace de travail en diminuant et en augmentant simultanément les longueurs des câbles, tout en les gardant sous tension.

L'objectif de ce travail consiste à étudier et réaliser un robot à 4 câbles. L'étude comporte la modélisation, l'optimisation ainsi la commande. La réalisation comporte un prototype de robot parallèle planaire à 4 câbles pour la réhabilitation des bras pour but d'aider les gents à des besoins spéciaux à l'écriture.

Le mémoire est organisé en deux parties l'une consacrée à l'étude théorique et l'autre à la réalisation expérimentale du robot, et sept chapitres. Le premier est consacré à l'état de l'art concernant les robots à câbles. Quelques applications importantes relatives à ces robots sont présentées.

Dans le deuxième chapitre, nous présentons une description détaillée de la structure étudiée. Nous établissons les différents modèles à savoir le modèle géométrique inverse (MGI), le modèle géométrique direct (MGD), le modèle cinématique inverse (MCI), le modèle cinématique direct (MCD), ainsi que le modèle dynamique (MD) en utilisant la méthode de Lagrange, et de Newton Euler.

Le troisième chapitre traite l'analyse statique des forces et le calcul des couples optimums sur les moteurs suivant la méthode décrite par la référence [9], et par la méthode d'essaim de particules.

Le quatrième chapitre comporte la commande en boucle ouvert du robot, suivi par un cinquième chapitre qui traite de la simulation dynamique de l'effecteur en boucle ouverte. La commande en boucle fermée par modes glissants du robot est traité dans le chapitre six. Le chapitre sept comporte la réalisation pratique du prototype. Le mémoire est terminé par une conclusion générale.

Chapitre 1

État de l'art

1.1 Introduction

Ce premier chapitre est consacré à l'état de l'art concernant les robots à câbles. Nous présentons quelques applications importantes relatives aux robots parallèles à câbles ainsi que les approches de commande présentées dans la littérature.

Il existe plusieurs types de robots : robots à roues, robots marcheurs, robots à Chenille, robots manipulateurs... etc. Les robots industriels sont les premiers à avoir été produits en grand nombre et ils sont encore la plus grande population robotique. Ces robots sont ceux qu'on retrouve particulièrement sur les chaînes de montage, tels que les robots soudeurs, de démolition, de nettoyage, d'emballage ou de surveillance. On trouve aussi en chirurgie médicale des robots qui se présentent sous forme de bras mécaniques couplés à un ordinateur ou par des bras articulés actionnés par le chirurgien et dont les mouvements sont reproduits par le robot en temps réel. Les robots domestiques peuvent faire de multiples tâches ou simplement nous divertir. Les robots militaires sont principalement utilisés pour la surveillance aussi bien dans les airs que dans la mer. Les robots explorateurs remplacent l'homme dans des environnements difficiles, par exemple pour prendre des photos sur les planètes. Lorsqu'on parle de robots, il ne faut pas oublier de mentionner les robots anthropomorphiques (robots ressemblant à l'être humain) qui sont probablement la principale motivation des roboticiens.

1.2 Présentation d'un Robot Parallèle à Câbles

Un bras manipulateur sériel est un robot dont l'organe terminal, appelé effecteur, est relié à la base par une série de bras articulés (chaine ouverte). Un robot parallèle est un mécanisme où l'organe terminal est relié par plus d'un bras à la base (chaine fermée) [4].

Un robot parallèle à câbles est un type particulier de robot parallèle utilisant des liaisons constituées de câbles. Ces robots sont principalement constitués d'une base, d'une plate-

forme mobile (organe terminal), des câbles reliant en parallèle la plate-forme à la base et d'un ensemble de poulies motorisées. La figure (1.1) extraite de la référence [1] montre un exemple de robot à câbles. Généralement, la base est fixe et chaque câble est attaché à une des extrémités de la plate-forme. Sous l'effet des moments moteurs, le câble s'enroule ou se déroule autour de la poulie permettant de contrôler la position et l'orientation de la plateforme mobile.

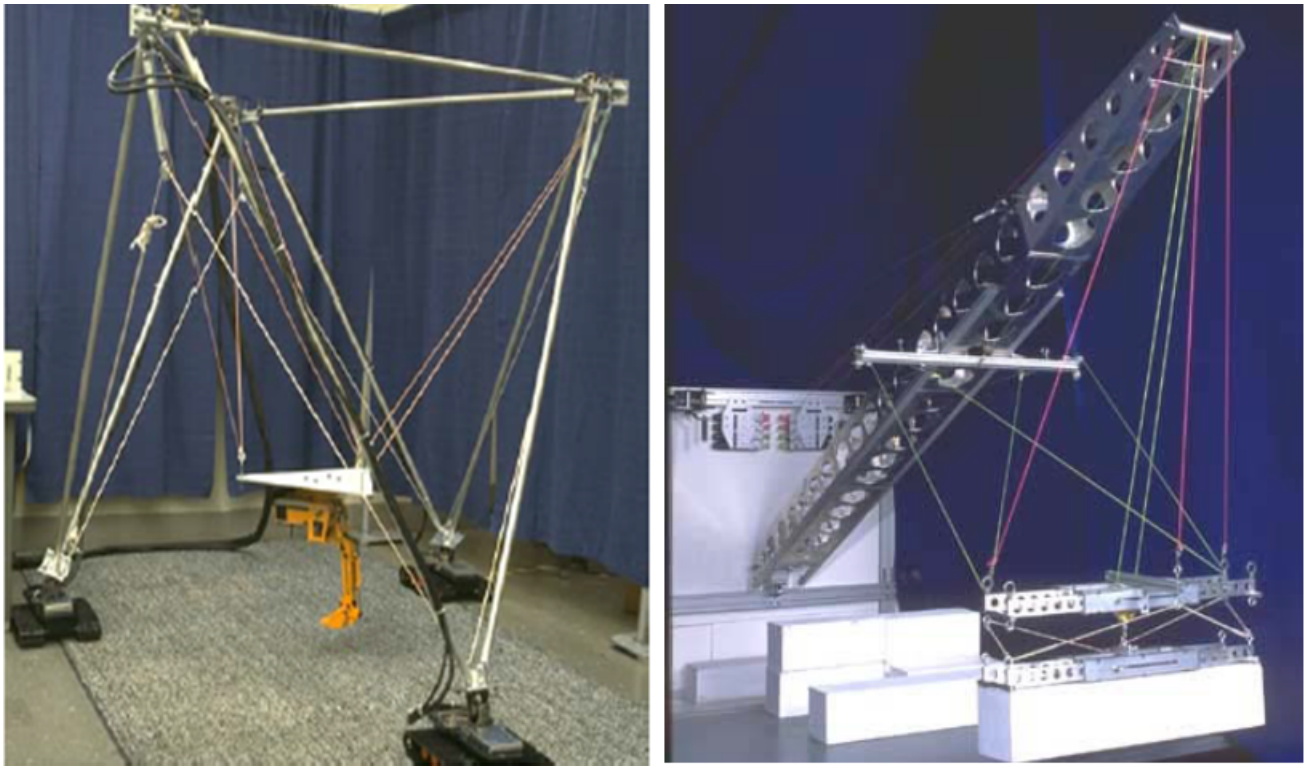


FIGURE 1.1 – Exemple de robot parallèle à câbles (ROBOCRANE)[1]

1.2.1 Avantages et inconvénients des Robots Parallèles à Câbles

Les Robots parallèles à câbles présentent plusieurs avantages [3] :

- Un grand espace de travail en comparaison avec les manipulateurs parallèles classiques ;
- Les robots à câbles ont une structure légère qui présente de bonnes propriétés dynamiques ;
- Un rapport charge utile/masse élevé ;
- Un coût de construction réduit (construction économique) ;
- Les robots parallèles à câbles, de structure facile à monter, à démonter, à stocker, à transporter et donc à utiliser (une bonne portabilité).

Le principal inconvénient des robots à câbles réside dans le fait que les actionneurs ne peuvent que tirer sur câbles et non pas les pousser ; ce qui limite certaines possibilités d'action.

1.2.2 Les Applications des Robots Parallèles à Câbles

Bien que les études sur les robots parallèles soient récentes, différentes applications ont été envisagées, notamment en aviation (simulateurs de vol), dans l'interfaçage haptique, pour le soulèvement des charges, ainsi que pour la réhabilitation des handicapés. Afin d'illustrer certaines utilisations, on présentera quelque exemples d'applications typiques.

1. **Skycam**

   L'application la plus connue est sans doute la Skycam [1], une caméra contrôlée par un mécanisme à câbles Figure (1.2). Elle est utilisée pour la télédiffusion des parties de football professionnel elle procure une meilleur fluidité dans le suivi des rencontres.

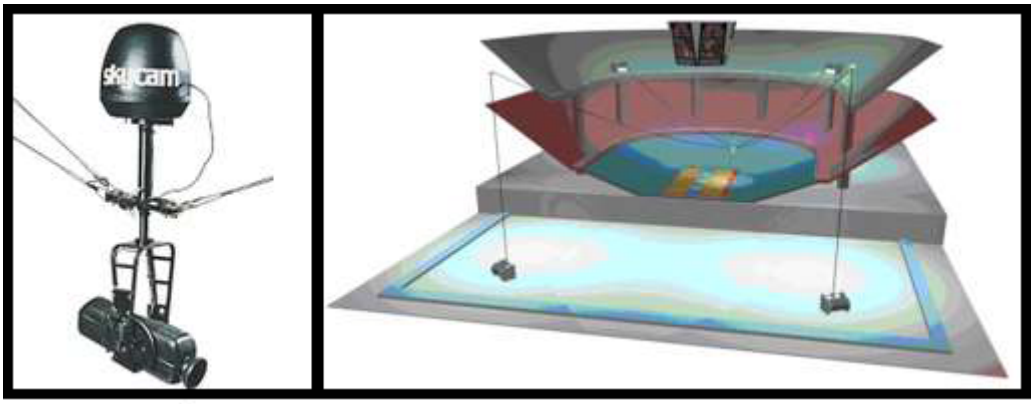


   FIGURE 1.2 – Caméra contrôlée par un mécanisme à câbles avec un Exemple de positionnement dans un stade [1]

2. **IPAnema**

   Depuis 2006, le groupe au Fraunhofer IPA développe une famille de robots nommé IPAnema.Qu'il s'agisse de systèmes économes en énergie ou d'équipements destinés à la fabrication de produits ou d'installation comme les centrales solaires ou les éoliennes[37]. Les composants jusqu'ici grands et/ou lourds sont la plupart du temps manipulés en utilisant les grues conventionnelles [7]. Le robot est un mécanisme parallèle à 6ddl entraîné par des câbles Figure (1.3), un des principaux objectifs des projets de recherche en cours est le développement d'un robot à câble de qualité industrielle avec haute robustesse et précision.

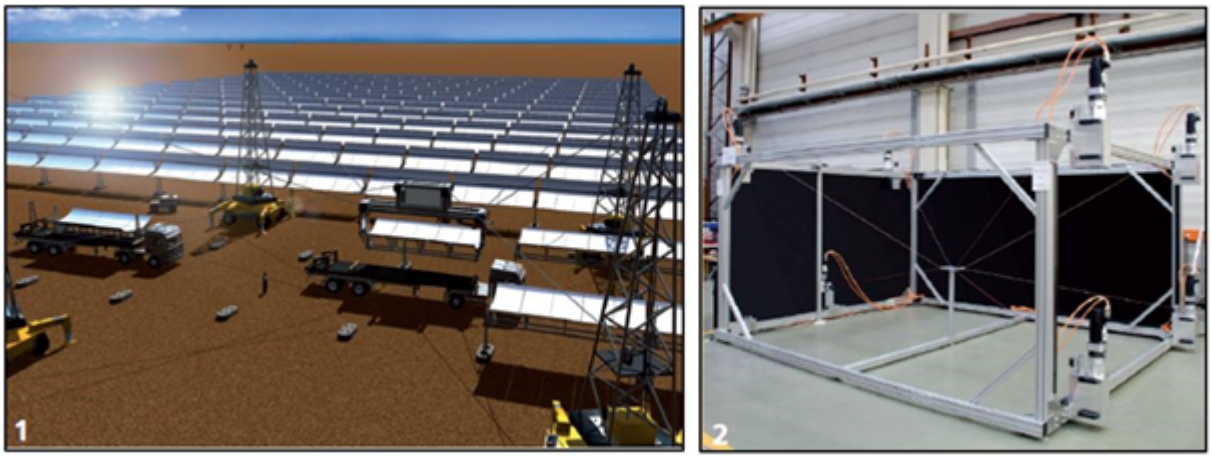


FIGURE 1.3 – IPAnema, mécanisme parallèle à 6ddl entraîné par des câbles [7]

3. **CaTraSys**

   Un autre champ d'intérêt dans les applications biomédicales est le suivi du mouvement. On citera l'exemple de CaTraSys (Cassino Tracking System) a été utilisé pour l'identification des paramètres cinématiques et la mobilité de l'homme [2]. Le CaTraSys est un système de mesure qui a été conçu et construit au LARM (laboratoire de robotique et mécatronique) à Cassino (Italie). Il a été utilisé pour déterminer la position de l'extrémité des membres pendant son mouvement et en plus, il peut mesurer les forces et les couples exercés par le membre comme le montre la figure (1.4).

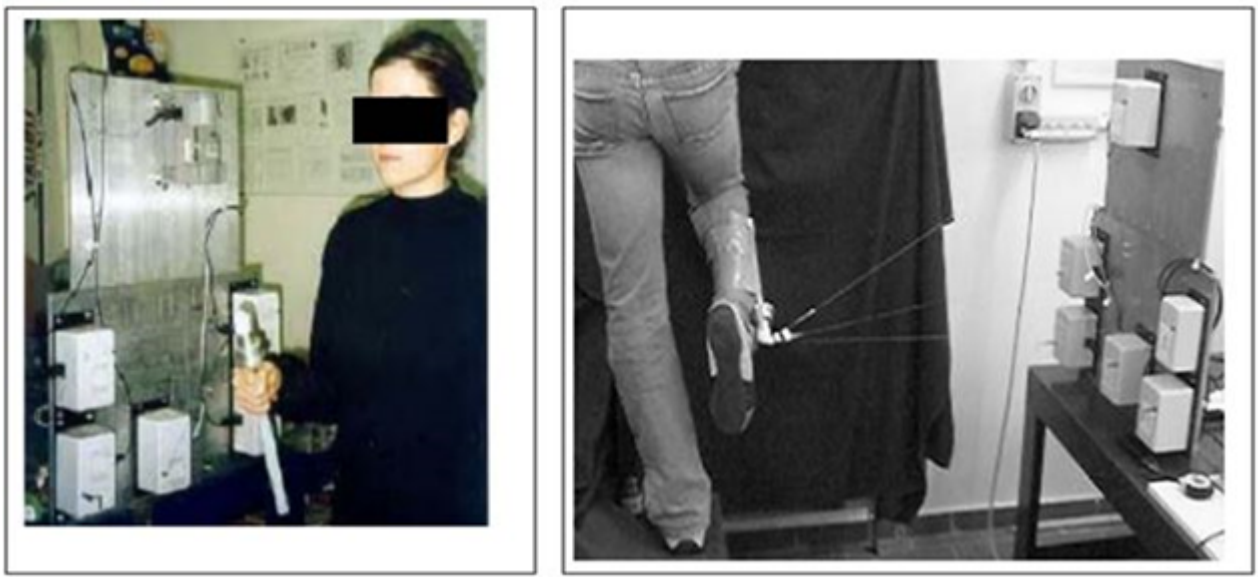


FIGURE 1.4 – Représentation d'un capteur de force pour les robots à câbles (CaTraSys) [2]

4. **STRING-MAN**

   Système à câbles pour la réadaptation de la jambe ,ce système est appelé STRING-MAN. C'est un manipulateur à câbles qui soutient le patient durant la thérapie de

rétablissement [6]. Sa conception a été inspirée par le principe des cordes de marionnette. La structure mécanique, dans la figure (1.5), a une configuration avec dix câbles. Il y a un tapis roulant qui permet le mouvement du patient. Le corps humain constitue l'organe terminal du robot.

FIGURE 1.5 – Le prototype mécanique de STRING-MAN [6]

5. **CSHI**

Une interface haptique sur un robot à huit câbles a été développée et établie à l'Université de l'Ohio. Le but est de créer un dispositif d'entrée/sortie qui fournit des forces et des moments à six degrés de liberté (6 ddl),qui agissent sur l'opérateur humain en réalité virtuel dans les applications à distances [5]. La structure est représentée dans la figure (1.6).

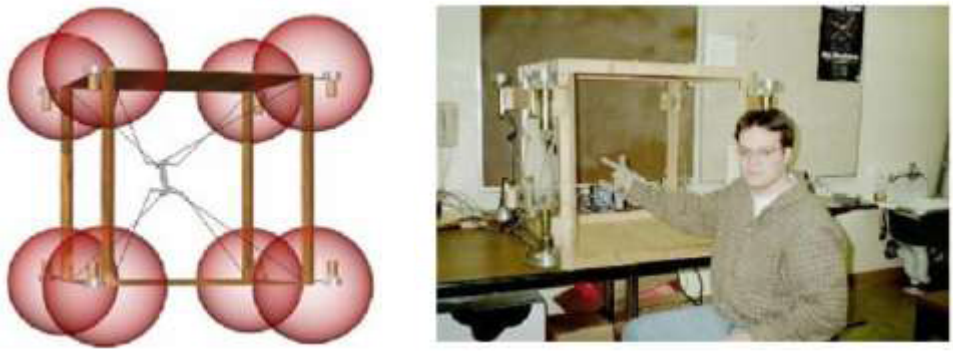
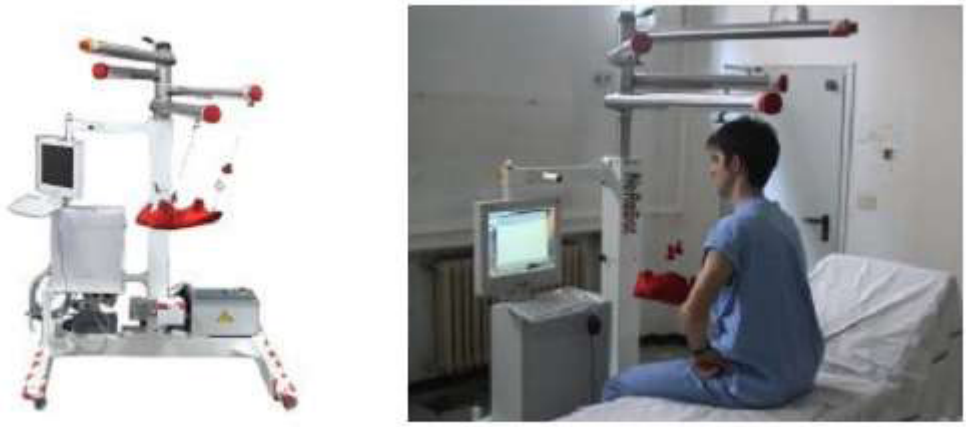


FIGURE 1.6 – Interface haptique de 6 degrés de liberté [5]

6. **NeReBot**

NeReBot (Neuro Rehabilitation roBot) et un robot à câbles développé par le laboratoire de robotique du département de l'innovation mécanique et management à l'université de (PADUA) Italie. Il a trois degrés de liberté et a été conçu pour le traitement et la réhabilitation pour post-choc [1].



(a) (b)

FIGURE 1.7 – NeReBot est un robot à câble basé pour la réadaptation des membres supérieurs avec 3 ddl [1] : a) la structure mécanique b) le système pendant les essais cliniques

La structure mécanique du manipulateur consiste en une colonne centrale et quatre tubes de liaisons en aluminium, comme montre la figure (1.7),les trois câbles en plastique liés avec une attelle dons laquelle le membre du patient est positionné. Chaque câble est conduit par un moteur débroché positionné sur la base de la structure, le robot réalise une trajectoire spatiale de l'attelle en tirant les câbles.

La structure mécanique est ajustée d'une façon manuelle par le thérapeute par apport au besoin spécifique du patient et de la thérapie. C'est une tache compliquée qui exige une expérience et une compétence technique. La position angulaire de chaque liaison et la position linéaire des points d'entrée des câbles à travers les liens doivent être établies.

La thérapie consiste en des mouvements répétitifs passifs performants des membres supérieurs du patient, il y a une phase d'apprentissage durant laquelle le thérapeute déplace les membres à travers une trajectoire et en sauvegarde les positions angulaire de chaque moteur.

Le contrôle est obtenu en utilisant le software qui implémente l'ensemble du contrôle du mouvement (une action " PID proportionnel intégral dérivé " pour commander séparément chaque moteur) et une interface graphique ("interaction Homme Machine")pour l'utilisateur.

1.3 Conclusion

La robotique en général et les manipulateurs parallèles en particuliers ont connue un grand essor des ces dernières années,notamment dans l'industrie ou leur utilisation s'est

largement démocratisée. De plus en plus de tâches sont dévolues aux robots parallèle à câbles qui sont plus en plus complexes. les robot manipulateurs à câbles sont utilisés dans de multiples fonctions qui vont du soudage,à la manutention en passant par l'usinage ainsi dans le demain biomédicale.Il est évident que chaque tâche requiert un type particulier de manipulateur suivant les contraintes à respecter.

Dans ce chapitre on a parlé sur les robots parallèles à câbles, ainsi que les avantages et inconvénients,s'en est suivi des quelques applications de ces robots.

Chapitre 2

Modélisation du robot à 4 câbles

2.1 Introduction

Dans ce chapitre on va présenter le modèle géométrique direct et inverse, cinématique et dynamique du robot à quatre câbles permettant de réaliser un mouvement plan (2 ddl), dont l'espace de travail est de forme carré comme illustré dans la figure (2.1).

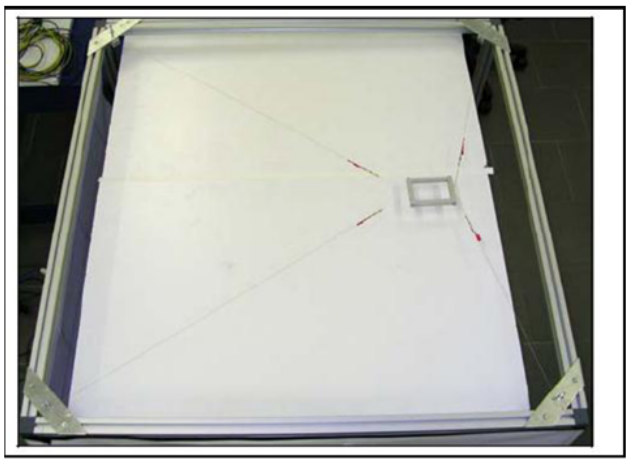


FIGURE 2.1 – Robot parallèle à quatre câbles [8]

La structure mécanique de ce robot est constituée de :

* une base (cadre) fixe de forme carré.
* une plateforme mobile, qui porte l'organe terminal.
* La base et la plateforme mobile sont reliées par quatre câbles.
* chaque câble est lié à la base par un actionneur (moteurs).

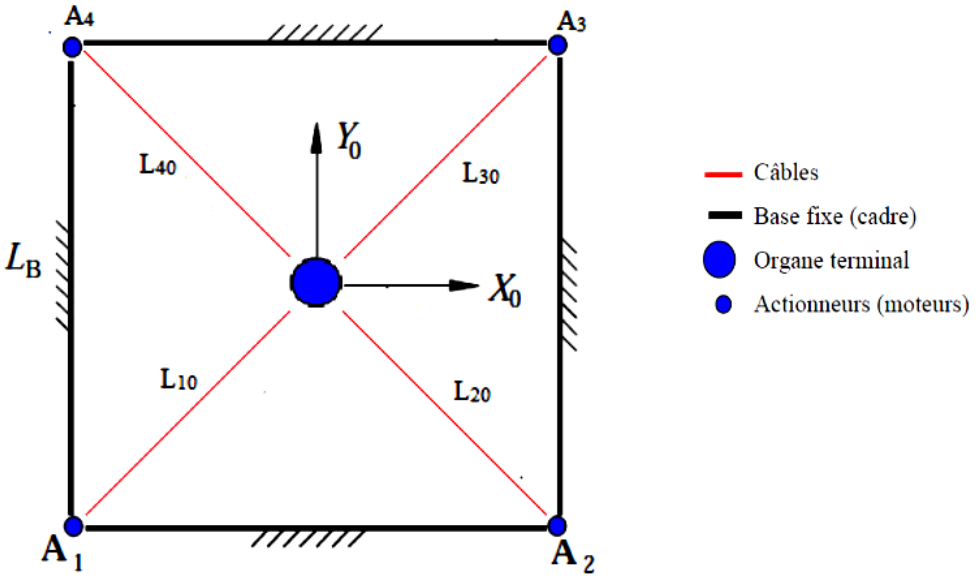


FIGURE 2.2 – Schématisation du robot parallèle à quatre câbles

Avant de pouvoir appliquer des lois de commande sur le Robot, il est à présent primordial de le modéliser afin d'avoir les outils nécessaires à sa description et afin de formaliser les relations mathématiques qui existent entre l'entrée du robot (variable articulaire) et la sortie de celui-ci (variable opérationnelle)(Figure(2.2)).

Plusieurs niveaux de modélisation sont possibles, dépendant des spécifications et de la nature de l'application envisagée : modèle géométrique, cinématique et dynamique.

Cette modélisation représente les transformations (ou les relations) entre l'espace opérationnel (dans lequel on définit la situation de l'organe terminal) et l'espace articulaire (dans lequel on définie la configuration des articulations du robot). On distingue :

- Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
- Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.
- Les modèles dynamiques définissant les équations du mouvement du robot qui permettent d'établir les relations entre les couples ou les forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations.

2.2 Modélisation géométrique

La modélisation géométrique d'une structure robotisée consiste simplement en la mise sous forme d'équations qui lie la position de l'organe terminal (effectuant la tâche) aux différentes coordonnées articulaires du manipulateur, Figure (2.3).

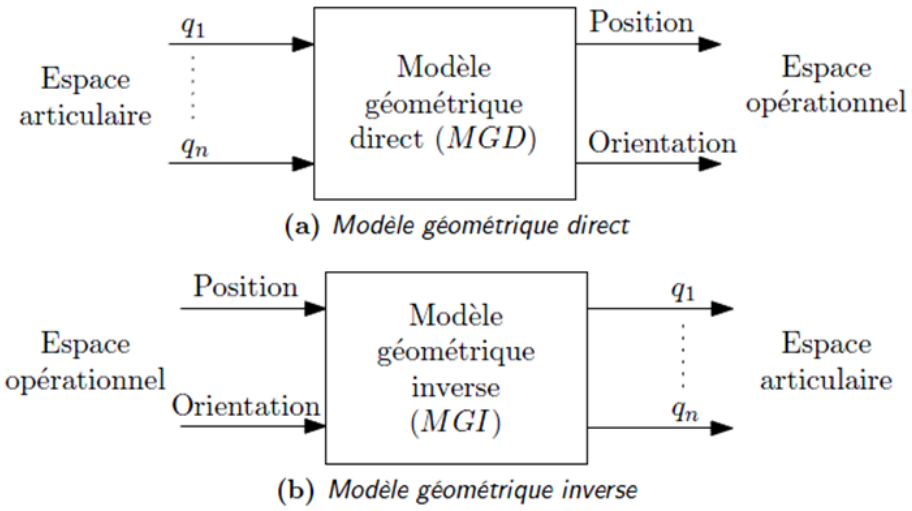


FIGURE 2.3 – Modèle géométrique

On établira dans ce qui suit le modèle géométrique direct et inverse du robot parallèle à quatre câbles la figure (2.4) nous montre l'organe terminal dans la position (x,y) et les différentes longueurs des câbles (L_i) ainsi que les angles que font ces dernier avec l'axe des x.

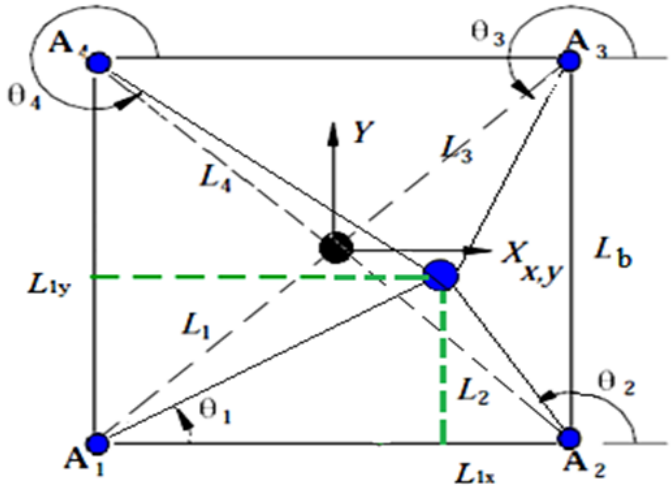


FIGURE 2.4 – Modèle géométrique du robot à quatre câbles

2.2.1 Modèle géométrique Inverse

On cherche à trouver les longueurs des câbles L_i et les angles θ_i en fonction des coordonnées de l'effecteur (x, y), exprimés par les équations suivantes [8] :

$$L_i = \sqrt{L_{ix}^2 + L_{iy}^2} \qquad i = 1, 4 \tag{2.1}$$

avec :

$$L_{ix} = x - A_{ix} \qquad i = 1, 4 \tag{2.2}$$

$$L_{iy} = y - A_{iy} \qquad i = 1, 4 \tag{2.3}$$

En remplaçant (2.2) et (2.3) dans (2.1) on trouve :

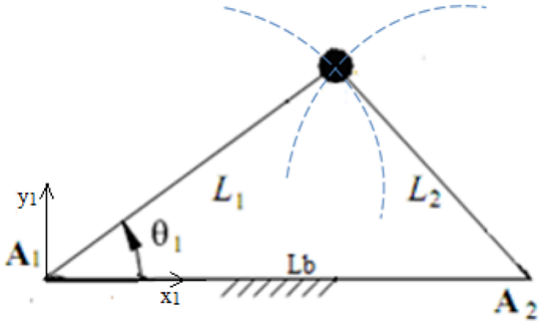
$$L_i = \sqrt{(x - A_{ix})^2 + (y - A_{iy})^2} \qquad i = 1, 4 \tag{2.4}$$

$$\theta_i = \arctan(\frac{y - A_{iy}}{x - A_{ix}}) \qquad i = 1, 4 \tag{2.5}$$

2.2.2 Modèle géométrique Direct

Le MGD exprime la position de l'effecteur M(x,y) en fonction des longueurs des câbles L_i. Pour les manipulateurs parallèles, le modèle géométrique directe est difficile à résoudre à cause de sa structure fermée (les angles (θ_i) sont liés avec les longueurs des câbles L_i). La relation entre la position X=(x, y) et les coordonnées généralisées est non linéaire.

Ce problème peut être simplifié en déplaçant le repère R (O , X , Y) au point A_1 ce qui nous donne de nouvelles coordonnées des points A_1 = (0 , 0) et A_2 = (Lb , 0),comme montre la figure(2.2.2). Alors la solution du modèle géométrique direct est l'intersection de deux cercles, un de centre A_1 avec un rayon L_1, et l'autre de centre A_2 de rayon L_2 :



Le résultat est [8] :

On a

$$x_1^2 + y_1^2 = L_1^2$$

$$(Lb - x_1)^2 + y_1^2 = L_2^2$$

Ce qui donne :

$$(Lb - x_1)^2 - x_1^2 = L_2^2 - L_1^2$$

$$x_1^2 - (Lb^2 + x_1^2 - 2 * x_1 * Lb) = L_1^2 - L_2^2$$

$$2 * x_1 * Lb - Lb^2 = L_1^2 - L_2^2$$

D'où :

$$\begin{cases} x &= \frac{Lb^2+L_1^2-L_2^2}{2Lb^2} \\ y &= \pm\sqrt{L_2^2 - (Lb - x_1)^2} \quad = \pm\sqrt{L_1^2 - x_1^2} \end{cases}$$

La solution de modèle géométrique direct exige le choix d'une valeur positive de y.

2.3 Modélisation Cinématique

La modélisation cinématique d'un robot en général permet d'établir les relations entre les vitesses articulaires (généralisées) et les vitesses opérationnelles de l'organe terminal. Dans cette partie nous présentons la modélisation cinématique inverse et directe du robot plan à 4 câbles.

2.3.1 Modèle Cinématique Inverse

Pour calculer le modèle cinématique inverse, nous considérons le i$^{\text{ème}}$ vecteur obtenu des équations (2.2) et (2.3) :

$$x = A_{ix} + L_i \cos(\theta_i) \qquad i = 1, 4$$

Et :

$$y = A_{iy} + L_i \sin(\theta_i) \qquad i = 1, 4$$

$$\begin{pmatrix} x \\ \\ y \end{pmatrix} = \begin{pmatrix} A_{ix} + L_i \cos(\theta_i) \\ \\ A_{iy} + L_i \sin(\theta_i) \end{pmatrix} \tag{2.6}$$

Si on dérive $\begin{pmatrix} x \\ \\ y \end{pmatrix}$ par rapport au temps, on obtient :

$$\begin{pmatrix} \dot{x} \\ \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) & -L_i \sin(\theta_i) \\ \\ \sin(\theta_i) & L_i \cos(\theta_i) \end{pmatrix} \begin{pmatrix} \dot{L}_i \\ \\ \dot{\theta}_i \end{pmatrix} \qquad i = 1, 4 \tag{2.7}$$

En inversant l'équation (2.7) on obtient :

$$\begin{pmatrix} \dot{L}_i \\ \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) & \sin(\theta_i) \\ \\ \frac{-\sin(\theta_i)}{L_i} & \frac{-\cos(\theta_i)}{L_i} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \\ \dot{y} \end{pmatrix} \qquad i = 1, 4 \tag{2.8}$$

Comme nous nous intéressons à la vitesse des câbles en fonction de la vitesse de l'effecteur nous pouvons extraire la première ligne de (2.8) pour obtenir :

$$\begin{pmatrix} \dot{L}_1 \\ \\ \dot{L}_2 \\ \\ \dot{L}_3 \\ \\ \dot{L}_4 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \\ \cos(\theta_2) & \sin(\theta_2) \\ \\ \cos(\theta_3) & \sin(\theta_3) \\ \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \\ \dot{y} \end{pmatrix} \qquad i = 1, 4 \tag{2.9}$$

On pose :

$$M = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \\ \cos(\theta_2) & \sin(\theta_2) \\ \\ \cos(\theta_3) & \sin(\theta_3) \\ \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \quad M \ : \ \textit{Jacobienne inverse}$$

2.3.2 Modèle Cinématique Direct

Pour obtenir le modèle cinématique direct, on doit inverser l'équation (2.9) qui nous donne : $\dot{X} = M^{-1}\dot{L}$. donc la solution exige le calcul de la matrice Jacobienne inverse. En raison de redondance d'actionnement, M n'est pas carré mais de dimension (n * 2) ce qui nous crées un problème de calcule de M^{-1}. Alors pour résoudre ce problème, nous avons utilisés la pseudo inverse de Moore-Penrose [8] :

$$\dot{X} = M^{+}\dot{L} \quad ou : \quad M^{+} = (M^{T}M)^{-1}M^{T}.$$

On peut écrire l'équation (2.9) sous la forme : $\dot{X} = M^{-1}\dot{L}$

Où :

$\dot{L}$: est le vecteur de vitesse du quatre câbles.
M^{-1} : est la matrice Jacobienne inverse.
M^{+} : pseudo-inverse de la matrice M.
$\dot{X} = (\dot{x}, \dot{y})$: est le vecteur de vitesse de l'organe terminal.

2.4 Modèle Dynamique

L'étude dynamique du système peut se faire par différentes méthodes, On peut citer Newton-Euler, puissances virtuelles et la méthode de Lagrange. On va utiliser 2 méthodes pour établir les équations de mouvement de l'effecteur.

2.4.1 Méthode de Lagrange

Pour commencer notre étude on doit tout d'abord connaitre les différentes forces qui engendrent le mouvement. Notre système comporte quatre moteurs sur lesquels on fixe des poulies reliées à un effecteur par l'intermédiaire de câbles. La figure (2.5) ci-dessous montre les différentes forces et moments qui s'appliquent sur la i$^{\text{ème}}$ poulie[20].

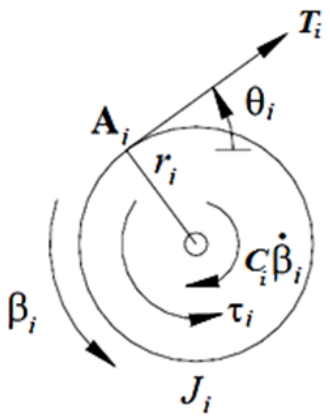


FIGURE 2.5 – Diagramme du corps libre de la i$^{\text{ème}}$ poulie [9]

2.4.1.1 Équation de Lagrange

Afin d'établir les équations de mouvement par la méthode de Lagrange, nous devons au préalable calculer les énergies cinétique (T), potentielle (U) et de dissipation (D) du système. L'équation (2.10) représente l'équation de Lagrange en coordonnées généralisées q.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = Q_d \tag{2.10}$$

Où L est le lagrangien du système défini par la différence entre l'énergie cinétique et potentielle (T-U), et Q_d est le vecteur des forces données lié aux couples des moteurs par des relations qu'on retrouvera par la suite.

2.4.1.2 A . Calcul des vitesses et accélérations angulaires des poulies :

Pour calculer les énergies cinétiques et de dissipations du système on doit d'abord connaitre les vitesses et accélérations angulaire des poulies. L'équation (2.11) donne les angles de ro-

tation β_i des poulies :

$$\beta = \begin{pmatrix} \beta_1(X) \\ \beta_2(X) \\ \beta_3(X) \\ \beta_4(X) \end{pmatrix} = \frac{1}{r} \begin{pmatrix} L_{10} - L_1 \\ L_{20} - L_2 \\ L_{30} - L_3 \\ L_{40} - L_4 \end{pmatrix} \tag{2.11}$$

En dérivant par rapport au temps l'équation (2.11) et de l'équation (2.9) on obtient les vitesses angulaires des poulies :

$$\dot{\beta} = -\frac{1}{r} \begin{pmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \end{pmatrix} = -\frac{1}{r} \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{2.12}$$

Et en dérivant par rapport au temps l'équation (2.12) on obtient l'accélération angulaire des poulies :

$$\ddot{beta} = \frac{1}{r} \left[\begin{pmatrix} \dot{\theta}_1 \sin(\theta_1) & -\dot{\theta}_1 \cos(\theta_1) \\ \dot{\theta}_2 \sin(\theta_2) & -\dot{\theta}_2 \cos(\theta_2) \\ \dot{\theta}_3 \sin(\theta_3) & -\dot{\theta}_3 \cos(\theta_3) \\ \dot{\theta}_4 \sin(\theta_4) & -\dot{\theta}_4 \cos(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} - \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \begin{pmatrix} \dot{\dot{x}} \\ \dot{\dot{y}} \end{pmatrix} \right] \tag{2.13}$$

2.4.1.3 B . L'énergie cinétique du système :

Ce sont les énergies cinétiques de l'effecteur et celles des poulies qui s'expriment comme suit :

$$T = \frac{1}{2} m V_e^2 + \frac{1}{2} J_1 \dot{\beta}_1^2 + \frac{1}{2} J_2 \dot{\beta}_2^2 + \frac{1}{2} J_3 \dot{\beta}_3^2 + \frac{1}{2} J_4 \dot{\beta}_4^2 \tag{2.14}$$

Où V_e représente la vitesse linéaire de l'effecteur qui s'exprime par :

$$V_e^2 = \dot{x}^2 + \dot{y}^2.$$

En remplaçant les $\dot{\beta}_i$ par leurs expressions de l'équation (2.12) on obtient :

$$T = \frac{1}{2}(\dot{x}^2+\dot{y}^2)+\frac{1}{2r^2}J_1(\cos(\theta_1)\dot{x}+\sin(\theta_1)\dot{y})^2+\frac{1}{2r^2}J_2(\cos(\theta_2)\dot{x}+\sin(\theta_2)\dot{y})^2+\frac{1}{2r^2}J_3(\cos(\theta_3)\dot{x}+\sin(\theta_3)\dot{y})^2$$
$$+\frac{1}{2r^2}J_4(\cos(\theta_4)\dot{x}+\sin(\theta_4)\dot{y})^2 \tag{2.15}$$

2.4.1.4 C . L'énergie potentielle du système

Vu que l'effecteur se déplace sur un plan horizontal et que les centres de masse des poulies ne se déplacent pas, et vu aussi que les fils sont considérés de masse négligeable et inextensible, alors l'énergie potentielle du system est nulle :

$$U = 0 \tag{2.16}$$

Le lagrangien du system s'écrire donc : L = T .

2.4.1.5 D . L'énergie de dissipation du système

L'énergie de dissipation du système est due principalement aux moteurs. Elle est défini comme suit :

$$D = \frac{1}{2}C_1\dot{\beta}_1^2 + \frac{1}{2}C_2\dot{\beta}_2^2 + \frac{1}{2}C_3\dot{\beta}_3^2 + \frac{1}{2}C_4\dot{\beta}_4^2 \tag{2.17}$$

En remplaçant les $\dot{\beta}_i$ par leurs expressions on obtient :

$$D = \frac{1}{2}C_1(\cos(\theta_1)\dot{x} + \sin(\theta_1)\dot{y})^2 + \frac{1}{2}C_2(\cos(\theta_2)\dot{x} + \sin(\theta_2)\dot{y})^2 + \frac{1}{2}C_3(\cos(\theta_3)\dot{x} + \sin(\theta_3)\dot{y})^2$$
$$+\frac{1}{2}C_4(\cos(\theta_4)\dot{x} + \sin(\theta_4)\dot{y})^2 \tag{2.18}$$

2.4.1.6 E . Calcul des dérivées

On dérive l'équation de l'énergie cinétique (2.15) par rapport à $\dot{x}$ on trouve :

$$\frac{\partial T}{\partial \dot{x}} = m\dot{x} + \frac{1}{2r^2}J_1\cos(\theta_1)(\cos(\theta_1)\dot{x} + \sin(\theta_1)\dot{y}) + \frac{1}{2r^2}J_2\cos(\theta_2)(\cos(\theta_2)\dot{x} + \sin(\theta_2)\dot{y})$$
$$+\frac{1}{2r^2}J_3\cos(\theta_3)(\cos(\theta_3)\dot{x} + \sin(\theta_3)\dot{y}) + \frac{1}{2r^2}J_4\cos(\theta_4)(\cos(\theta_4)\dot{x} + \sin(\theta_4)\dot{y}) \tag{2.19}$$

On dérive l'équation (2.19) par rapport au temps on obtient :

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}}\right) = m\ddot{x} + \frac{J_1}{r^2}[\cos^2(\theta_1)\ddot{x} + \cos(\theta_1)\sin(\theta_1)\ddot{y} - 2\dot{\theta}_1\cos(\theta_1)\sin(\theta_1)\dot{x} + \dot{\theta}_1(2\cos^2(\theta_1) - 1)\dot{y}]$$
$$+\frac{J_2}{r^2}[\cos^2(\theta_2)\ddot{x} + \cos(\theta_2)\sin(\theta_2)\ddot{y} - 2\dot{\theta}_2\cos(\theta_2)\sin(\theta_2)\dot{x} + \dot{\theta}_2(2\cos^2(\theta_2) - 1)\dot{y}]$$

$$+\frac{J_3}{r^2}[\cos^2(\theta_3)\ddot{x}+\cos(\theta_3)\sin(\theta_3)\ddot{y}-2\dot{\theta}_3\cos(\theta_3)\sin(\theta_3)\dot{x}+\dot{\theta}_3(2\cos^2(\theta_3)-1)\dot{y}]$$

$$+\frac{J_4}{r^2}[\cos^2(\theta_4)\ddot{x}+\cos(\theta_4)\sin(\theta_4)\ddot{y}-2\dot{\theta}_4\cos(\theta_4)\sin(\theta_4)\dot{x}+\dot{\theta}_4(2\cos^2(\theta_4)-1)\dot{y}] \tag{2.20}$$

On dérive l'équation (2.18) par rapport à $\dot{x}$:

$$\frac{\partial D}{\partial \dot{x}}=\frac{1}{r^2}C_1\cos(\theta_1)(\cos(\theta_1)\dot{x}+\sin(\theta_1)\dot{y})+\frac{1}{r^2}C_2\cos(\theta_2)(\cos(\theta_2)\dot{x}+\sin(\theta_2)\dot{y})+$$

$$\frac{1}{r^2}C_3\cos(\theta_3)(\cos(\theta_3)\dot{x}+\sin(\theta_3)\dot{y})+\frac{1}{r^2}C_4\cos(\theta_4)(\cos(\theta_4)\dot{x}+\sin(\theta_4)\dot{y}) \tag{2.21}$$

On dérive l'équation de l'énergie cinétique (2.15) par rapport à $\dot{y}$ on trouve :

$$\frac{\partial T}{\partial \dot{y}}=m\dot{y}+\frac{1}{2r^2}J_1\sin(\theta_1)(\cos(\theta_1)\dot{x}+\sin(\theta_1)\dot{y})+\frac{1}{2r^2}J_2\sin(\theta_2)(\cos(\theta_2)\dot{x}+\sin(\theta_2)\dot{y})+$$

$$\frac{1}{2r^2}J_3\sin(\theta_3)(\cos(\theta_3)\dot{x}+\sin(\theta_3)\dot{y})+\frac{1}{2r^2}J_4\sin(\theta_4)(\cos(\theta_4)\dot{x}+\sin(\theta_4)\dot{y}) \tag{2.22}$$

En dérivant l'équation (2.22) par rapport au temps :

$$\frac{d}{dt}(\frac{\partial T}{\partial \dot{y}})=m\ddot{y}+\frac{J_1}{r^2}[\sin^2(\theta_1)\ddot{y}+\cos(\theta_1)\sin(\theta_1)\ddot{x}+2\dot{\theta}_1\cos(\theta_1)\sin(\theta_1)\dot{y}+\dot{\theta}_1\cos^2(\theta_1)-\sin^2(\theta_1)\dot{x}]$$

$$+\frac{J_2}{r^2}[\sin^2(\theta_2)\ddot{y}+\cos(\theta_2)\sin(\theta_2)\ddot{x}+2\dot{\theta}_2\cos(\theta_2)\sin(\theta_2)\dot{y}+\dot{\theta}_2\cos^2(\theta_2)-\sin^2(\theta_2)\dot{x}]$$

$$+\frac{J_3}{r^2}[\sin^2(\theta_3)\ddot{y}+\cos(\theta_3)\sin(\theta_3)\ddot{x}+2\dot{\theta}_3\cos(\theta_3)\sin(\theta_3)\dot{y}+\dot{\theta}_3\cos^2(\theta_3)-\sin^2(\theta_3)\dot{x}]$$

$$+\frac{J_4}{r^2}[\sin^2(\theta_4)\ddot{y}+\cos(\theta_4)\sin(\theta_4)\ddot{x}+2\dot{\theta}_4\cos(\theta_4)\sin(\theta_4)\dot{y}+\dot{\theta}_4\cos^2(\theta_4)-\sin^2(\theta_4)\dot{x}] \tag{2.23}$$

On dérive l'équation (2.18) par rapport à $\dot{y}$:

$$\frac{\partial D}{\partial \dot{y}}=\frac{1}{r^2}C_1\sin(\theta_1)(\cos(\theta_1)\dot{x}+\sin(\theta_1)\dot{y})+\frac{1}{r^2}C_2\sin(\theta_2)(\cos(\theta_2)\dot{x}+\sin(\theta_2)\dot{y})+$$

$$\frac{1}{r^2}C_3\sin(\theta_3)(\cos(\theta_3)\dot{x}+\sin(\theta_3)\dot{y})+\frac{1}{r^2}C_4\sin(\theta_4)(\cos(\theta_4)\dot{x}+\sin(\theta_4)\dot{y}) \tag{2.24}$$

Les forces données sont calculées à partir de la puissance virtuelle des couples des quatre moteurs qui est donnée par [8] :

$$P^*=\tau_1\dot{\beta}_1^*+\tau_2\dot{\beta}_2^*+\tau_3\dot{\beta}_3^*+\tau_4\dot{\beta}_4^* \tag{2.25}$$

Tel que :

$$\dot{\beta}^*=-\frac{1}{r}\begin{pmatrix}\cos(\theta_1) & \sin(\theta_1)\\ \cos(\theta_2) & \sin(\theta_2)\\ \cos(\theta_3) & \sin(\theta_3)\\ \cos(\theta_4) & \sin(\theta_4)\end{pmatrix}\begin{pmatrix}\dot{x}^*\\ \dot{y}^*\end{pmatrix} \tag{2.26}$$

En remplaçant (2.26) dans (2.25) on obtient :

$$P^* = -\frac{1}{r}\begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}\begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix}\begin{pmatrix} \dot{x}^* \\ \dot{y}^* \end{pmatrix} \tag{2.27}$$

De l'équation (2.27) on tire les expressions des forces données suivant x et y :

$$Q_{dx} = -\frac{1}{r}(\tau_1\cos(\theta_1) + \tau_2\cos(\theta_2) + \tau_3\cos(\theta_3) + \tau_4\cos(\theta_4)) \tag{2.28}$$

$$Q_{dy} = -\frac{1}{r}(\tau_1\sin(\theta_1) + \tau_2\sin(\theta_2) + \tau_3\sin(\theta_3) + \tau_4\sin(\theta_4)) \tag{2.29}$$

$$Q_d = -\frac{1}{r}S(X)\tau$$

Tel que :

$$S(X) = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \tag{2.30}$$

Et :

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} \tag{2.31}$$

2.4.1.7 F . Équations de Lagrange sous forme matricielle

Après manipulation des équations obtenues précédemment on les réécrit sous forme la matricielle suivante :

$$M(X)\ddot{X} + N(X,\dot{X}) = S(X)\tau \tag{2.32}$$

Avec :

$$Q = rQ_d = S(X)\tau$$

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \tag{2.33}$$

Et :

$$N = \begin{pmatrix} N_{11} & N_{12} \\ \\ N_{21} & N_{22} \end{pmatrix} \tag{2.34}$$

Avec :

$$M_{11} = r * m + \frac{1}{r}\left(J_1 \cos^2(\theta_1) + J_2 \cos^2(\theta_2) + J_3 \cos^2(\theta_3) + J_4 \cos^2(\theta_4)\right) \tag{2.35}$$

$$M_{12} = \frac{1}{r}\left(J_1 \cos(\theta_1)\sin(\theta_1) + J_2 \cos(\theta_2)\sin(\theta_2) + J_3 \cos(\theta_3)\sin(\theta_3) + J_4 \cos(\theta_4)\sin(\theta_4)\right) \tag{2.36}$$

$$M_{12} = M_{21} \tag{2.37}$$

$$M_{22} = r * m + \frac{1}{r}\left(J_1 \sin^2(\theta_1) + J_2 \sin^2(\theta_2) + J_3 \sin^2(\theta_3) + J_4 \sin^2(\theta_4)\right) \tag{2.38}$$

$$N_{11} = \frac{1}{r}(\cos(\theta_1)\left(C_1\cos(\theta_1) - 2J_1\dot{\theta}_1\sin(\theta_1)\right) + \cos(\theta_2)\left(C_2\cos(\theta_2) - 2J_2\dot{\theta}_2\sin(\theta_2)\right)$$
$$+ \cos(\theta_3)\left(C_3\cos(\theta_3) - 2J_3\dot{\theta}_3\sin(\theta_3)\right) + \cos(\theta_4)\left(C_4\cos(\theta_4) - 2J_4\dot{\theta}_4\sin(\theta_4)\right)) \tag{2.39}$$

$$N_{22} = \frac{1}{r}(\sin(\theta_1)(C_1\cos(\theta_1) - 2J_1\dot{\theta}_1\cos(\theta_1)) + \sin(\theta_2)(C_2\sin(\theta_2) - 2J_2\dot{\theta}_2\cos(\theta_2))$$
$$+ \sin(\theta_3)(C_3\sin(\theta_3) - 2J_3\dot{\theta}_3\cos(\theta_3)) + \sin(\theta_4)(C_4\sin(\theta_4) - 2J_4\dot{\theta}_4\cos(\theta_4))) \tag{2.40}$$

$$N_{12} = \frac{1}{r}(C_1\sin(\theta_1)\cos(\theta_1) + J_1\dot{\theta}_1(\cos^2(\theta_1) - \sin^2(\theta_1))$$
$$+C_2\sin(\theta_2)\cos(\theta_2) + J_2\dot{\theta}_2(\cos^2(\theta_2) - \sin^2(\theta_2)) + C_3\sin(\theta_3)\cos(\theta_3) + J_3\dot{\theta}_3(\cos^2(\theta_3) - \sin^2(\theta_3))$$
$$+C_4\sin(\theta_4)\cos(\theta_4) + J_4\dot{\theta}_4(\cos^2(\theta_4) - \sin^2(\theta_4))) \tag{2.41}$$
$$N_{12} = N_{21}$$

2.4.2 Méthode de Newton-Euler

2.4.2.1 A . Modèle dynamique de l'effecteur

Le modèle dynamique de l'effecteur est exprimé par la relation suivante :

$$m\ddot{X} = F_R \tag{2.42}$$

$$\begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} \ddot{X} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} F_{Rx} \\ F_{Ry} \end{pmatrix} \tag{2.43}$$

Où :

m : est la matrice de masse.

$\ddot{X}$: Est le vecteur d'accélération de l'organe terminal.

$F_R = \begin{pmatrix} F_{Rx} & F_{Ry} \end{pmatrix}^T$: est la force résultante de toutes les tensions des câbles appliquées sur l'organe terminal .

2.4.2.2 B . Structure mécanique des moteurs

Le comportement dynamique des moteurs est exprimé par l'équation(voir figure (2.6)) :

$$J\ddot{\beta} + C\dot{\beta} = \tau - rT \tag{2.44}$$

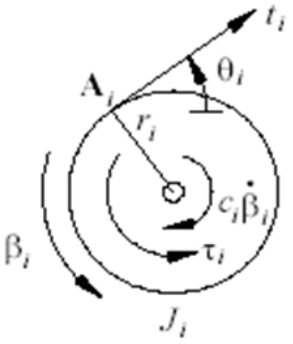


FIGURE 2.6 – Diagramme du corps libre de la i$^{\text{ème}}$ poulie.

avec :

$$J = \begin{pmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & J_3 & 0 \\ 0 & 0 & 0 & J_4 \end{pmatrix} \quad et \quad C = \begin{pmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{pmatrix}$$

Ce sont des matrices diagonales qui représentent les inerties et les coefficients d'amortissement visqueux de chaque moteur. Nous considérons que tous les rayons des poulies sont identiques (r_i =1.2.3.4). $\tau = (\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4)^T$ est le vecteur des couples appliqués par les moteurs, $T = (t_1 \quad t_2 \quad t_3 \quad t_4)^T$ est le vecteur des tensions des câbles. β est l'angle de rota-

tion de la poulie.
Donc :

$$T = \frac{1}{r}(\tau - J\dot{\beta} - C\dot{\beta}) \tag{2.45}$$

2.4.2.3 C . Modèle dynamique du système

Le modèle dynamique global du système est obtenu par combinaison entre le modèle dynamique de l'organe terminal et le modèle dynamique des moteurs [9]. Si on considère que les angles des poulies sont nuls quand la position de l'organe terminal est au centre du carré X = (0 , 0)^T, on a aussi :

$$L_{10} = L_{20} = L_{30} = L_{40} = \frac{Lb\sqrt{2}}{2} \tag{2.46}$$

La relation entre les angles des rotations (β_i) des poulies de rayon r et les variations des longueurs des câbles (ΔL_i) est :

$$\beta_i * r = -\Delta L_i \quad Avec \quad \Delta L_i = L_i - L_{i0} \tag{2.47}$$

Où :

$$L_i = \sqrt{(x - L_{i0x})^2 + y - L_{i0y})^2} \quad i = 1, 4 \tag{2.48}$$

Et L_{i0} sont les longueurs initiales des câbles :

$$L_{i0} = \sqrt{L_{i0x}^2 + L_{i0y}^2} \tag{2.49}$$

Donc :

$$\beta = \begin{pmatrix} \beta_1(X) \\ \beta_2(X) \\ \beta_3(X) \\ \beta_4(X) \end{pmatrix} = \frac{1}{r} \begin{pmatrix} L_{10} - L_1 \\ L_{20} - L_2 \\ L_{30} - L_3 \\ L_{40} - L_4 \end{pmatrix} \tag{2.50}$$

En dérivant β successivement par rapport au temps, on obtient :

$$\dot{\beta} = \begin{pmatrix} \dot{\beta}_1(X) \\ \dot{\beta}_2(X) \\ \dot{\beta}_3(X) \\ \dot{\beta}_4(X) \end{pmatrix} = -\frac{1}{r} \begin{pmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \end{pmatrix} \tag{2.51}$$

De l'équation :

$$\dot{\beta} = -\frac{1}{r}\begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{2.52}$$

D'où :

$$\ddot{\beta} = -\frac{1}{r}\left[\begin{pmatrix} \dot{\theta}_1\sin(\theta_1) & -\dot{\theta}_1\cos(\theta_1) \\ \dot{\theta}_2\sin(\theta_2) & -\dot{\theta}_2\cos(\theta_2) \\ \dot{\theta}_3\sin(\theta_3) & -\dot{\theta}_3\cos(\theta_3) \\ \dot{\theta}_4\sin(\theta_4) & -\dot{\theta}_4\cos(\theta_4) \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} - \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix}\begin{pmatrix} \dot{\dot{x}} \\ \dot{\dot{y}} \end{pmatrix}\right] \tag{2.53}$$

D'où :

$$T = \frac{1}{r}\left(\tau - J\left(\frac{d}{dt}\left(\frac{\partial\beta}{\partial X}\right)\dot{X} + \frac{\partial\beta}{\partial X}\dot{\dot{X}}\right) - C\frac{\partial\beta}{\partial X}\dot{X}\right) \tag{2.54}$$

Avec :

$$\dot{\beta} = \frac{\partial\beta}{\partial X}\dot{X} \tag{2.55}$$

$$\ddot{\beta} = \frac{d}{dt}\left(\frac{\partial\beta}{\partial X}\right)\dot{X} + \frac{\partial\beta}{\partial X}\ddot{X} \tag{2.56}$$

$$\frac{\partial\beta}{\partial X} = -\frac{1}{r}\begin{pmatrix} \frac{x-L_{10x}}{L_1} & \frac{y-L_{10y}}{L_1} \\ \frac{x-L_{20x}}{L_2} & \frac{y-L_{20y}}{L_2} \\ \frac{x-L_{30x}}{L_3} & \frac{y-L_{30y}}{L_3} \\ \frac{x-L_{40x}}{L_4} & \frac{y-L_{40y}}{L_4} \end{pmatrix} \tag{2.57}$$

Du modèle cinématique inverse :

$$x = L_{10x} + L_i\cos(\theta_i) \qquad y = L_{10y} + L_i\sin(\theta_i) \tag{2.58}$$

D'où :

$$\frac{\partial\beta}{\partial X} = -\frac{1}{r}\begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} \tag{2.59}$$

$$F_R = ST \tag{2.60}$$

avec F c'est le vecteur résultant des forces appliquées sur l'effecteur suivant les deux axes. où :

$$S = \left(-\overrightarrow{L_1} \ -\overrightarrow{L_2} \ -\overrightarrow{L_3} \ -\overrightarrow{L_4}\right)^T \tag{2.61}$$

est une matrice de dimension (2×4).

$$S(X) = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \tag{2.62}$$

Enfin, en combinant les résultats précédents (2.43) (2.54) et (2.60), l'ensemble des équations du modèle dynamique peut être exprimé dans une forme standard pour les systèmes robotiques :

$$M(X)\ddot{X} + N(X, \dot{X}) = S(X)\tau \tag{2.63}$$

où :

$$M = r * m + S(X)J\frac{\partial \beta}{\partial X}$$

Et :

$$N(X, \dot{X}) = S(X)(J\frac{d}{dt}\frac{\partial \beta}{\partial X} + C\frac{\partial \beta}{\partial X})\dot{X}$$

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \qquad Et \qquad N(X, \dot{X}) = \begin{pmatrix} N_1(X, \dot{X}) \\ N_2(X, \dot{X}) \end{pmatrix}$$

Avec :

$$M_{11} = r * m + \frac{1}{r}\left(J_1 \cos^2(\theta_1) + J_2 \cos^2(\theta_2) + J_3 \cos^2(\theta_3) + J_4 \cos^2(\theta_4)\right)$$

$$M_{12} = \frac{1}{r}\left(J_1 \cos(\theta_1)\sin(\theta_1) + J_2 \cos(\theta_2)\sin(\theta_2) + J_3 \cos(\theta_3)\sin(\theta_3) + J_4 \cos(\theta_4)\sin(\theta_4)\right)$$

$$M_{12} = M_{21}$$

$$M_{22} = r * m + \frac{1}{r}\left(J_1 \sin^2(\theta_1) + J_2 \sin^2(\theta_2) + J_3 \sin^2(\theta_3) + J_4 \sin^2(\theta_4)\right)$$

$$N_{11} = \frac{1}{r}(\cos(\theta_1)(C_1 \cos(\theta_1) - 2J_1\dot{\theta}_1 \sin(\theta_1)) + \cos(\theta_2)(C_2 \cos(\theta_2) - 2J_2\dot{\theta}_2 \sin(\theta_2)) +$$

$$\cos(\theta_3)(C_3 \cos(\theta_3) - 2J_3\dot{\theta}_3 \sin(\theta_3)) + \cos(\theta_4)(C_4 \cos(\theta_4) - 2J_4\dot{\theta}_4 \sin(\theta_4)))$$

$$N_{22} = \frac{1}{r}(\sin(\theta_1)(C_1\cos(\theta_1) - 2J_1\dot{\theta}_1\cos(\theta_1)) + \sin(\theta_2)(C_2\sin(\theta_2) - 2J_2\dot{\theta}_2\cos(\theta_2)) +$$

$$\sin(\theta_3)(C_3\sin(\theta_3) - 2J_3\dot{\theta}_3\cos(\theta_3)) + \sin(\theta_4)(C_4\sin(\theta_4) - 2J_4\dot{\theta}_4\cos(\theta_4)))$$

$$N_{12} = \frac{1}{r}(C_1\sin(\theta_1)\cos(\theta_1) + J_1\dot{\theta}_1(\cos^2(\theta_1) - \sin^2(\theta_1)) + C_2\sin(\theta_2)\cos(\theta_2) + J_2\dot{\theta}_2(\cos^2(\theta_2) - \sin^2(\theta_2)) +$$

$$C_3\sin(\theta_3)\cos(\theta_3) + J_3\dot{\theta}_3(\cos^2(\theta_3) - \sin^2(\theta_3)) + C_4\sin(\theta_4)\cos(\theta_4) + J_4\dot{\theta}_4(\cos^2(\theta_4) - \sin^2(\theta_4)))$$

$$N_{12} = N_{21}$$

2.5 Validation et Simulation du Modèle Géométrique du Robot

Dans cette section on a développé des programmes sous Matlab pour la commande en boucle ouverte en position de l'organe terminal d'un robot à 4 câbles, basé sur le modèle géométrique inverse. Les résultats des calculs sont représentés sous forme de graphes correspondant aux variations de longueurs et des angles des câbles. Il permet également la visualisation graphique du système.

On a aussi simulé les tensions nécessaires pour que l'effecteur poursuive une trajectoire circulaire tout en imposant une force constante le long du parcours.

2.5.1 Simulation du Modèle Géométrique Inverse

On donne les valeurs de (x, y) et on détermine les longueurs du câbles et les angles pour chaque déplacement de l'organe terminal.

La Figure (2.7) illustre l'organigramme du programme calculant les différentes longueurs des 4 câbles du robot (L_1, L_2, L_3, L_4), ainsi que les angles (θ_1, θ_2, θ_3, θ_4) que font ces câbles avec l'axe des abscisses.

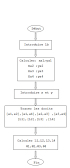


FIGURE 2.7 – Organigramme des longueurs et des angles des 4 câbles de robot

La Figure (2.8) représente le robot à 4 câbles avec l'organe terminal en position de repos (x = 0 , y = 0). L'organe terminal est repère par un signe plus (+).

$x_p(cm)$	$y_p(cm)$	$l_a(cm)$	$l_b(cm)$	$l_c(cm)$	$l_d(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
0	0	28.28	28.28	28.28	28.28	45	135	255	315

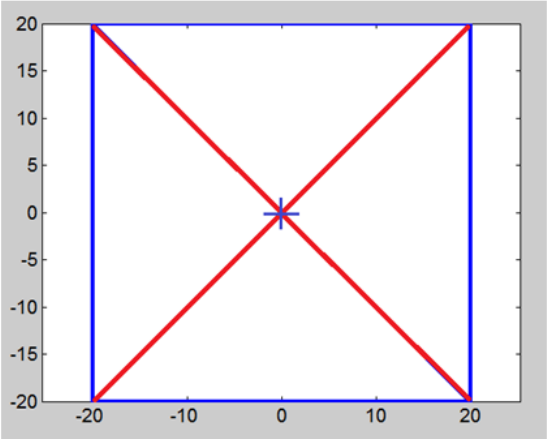


FIGURE 2.8 – Organe terminal en position initial

Les Figure (2.9) et Figure (2.9) montrent quelques positions de l'organe terminal obtenues par le programme.

Exemple 1 : x_p = 14 cm, y_p = 7 cm.

$x_p(cm)$	$y_p(cm)$	$l_a(cm)$	$l_b(cm)$	$l_c(cm)$	$l_d(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
14	7	43.41	27.65	14.31	36.40	38.45	102.52	245.22	339.07

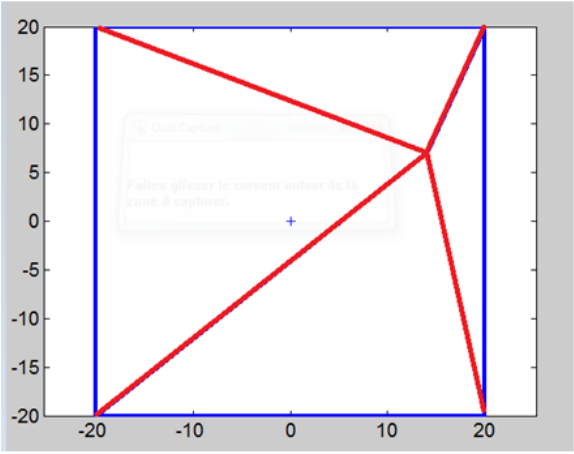


FIGURE 2.9 – Organe terminal en position (14,7)

Exemple 1 : x_p = 10 cm, y_p = -17 cm.

$x_p(cm)$	$y_p(cm)$	$l_a(cm)$	$l_b(cm)$	$l_c(cm)$	$l_d(cm)$	$\theta_1(deg)$	$\theta_2(deg)$	$\theta_3(deg)$	$\theta_4(deg)$
10	-17	30.14	10.44	38.32	47.63	5.71	163.30	254.87	309.03

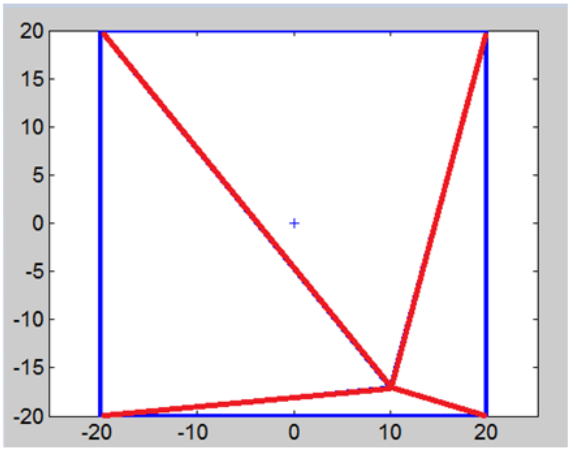


FIGURE 2.10 – Organe terminal en position (10,-17)

2.5.2 Simulation de la trajectoire de l'organe terminal

Dans cette partie, nous essayons de simuler les variations des longueurs des câbles, ainsi que les angles que font ceux-ci avec l'axe des abscisses, en fonction de l'angle polaire que fait l'organe terminal.

On propose une trajectoire circulaire d'un rayon R et dont le centre est l'origine O.

Les équations paramétriques de cette trajectoire sont :

$$x = R\cos(\theta)$$

$$y = R\sin(\theta)$$

où :

$$0 \leq \theta \leq 2\pi \quad et \quad 0 \leq R \leq R_{max}$$

où : $\mathrm{R}_{max} = \frac{L}{2}$ est la distance entre le centre du carré O et l'un des quatre côtés.

La Figure (2.11) montre l'organigramme du programme traçant la géométrie à 4 câbles, avec la trajectoire circulaire parcourue par l'organe terminal.

FIGURE 2.11 – Organigramme traçant la géométrie à 4 câbles avec la trajectoire circulaire

Pour une trajectoire circulaire de rayon $\mathrm{R} = \frac{R_{max}}{2}$ la géométrie de problème ressemble à la Figure (2.12).

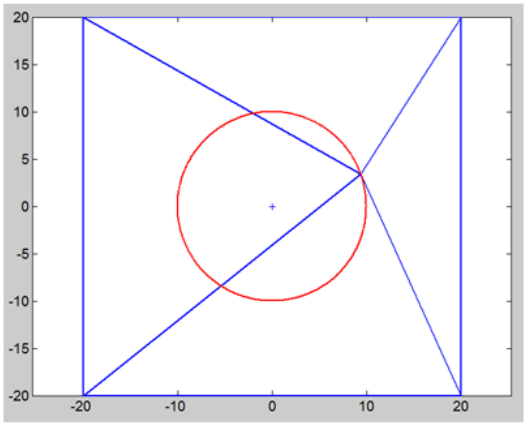


FIGURE 2.12 – trajectoire circulaire de rayon $R = \frac{R_{max}}{2}$

La Figure (2.13) illustre l'organigramme du programme traçant les variations des longueurs, des câbles et les angles de ceux-ci par rapport à l'axe des abscisses, en fonction de l'angle polaire que fait l'organe terminal lorsqu'il rebrousse la trajectoire circulaire.

FIGURE 2.13 – Organigramme traçant les variations des longueurs et des angles des câbles

2.5.2.1 A . Variation des longueur des câbles

Pour une trajectoire circulaire de rayon R = R_{max}, les variations des langueurs des 4 câbles sont représentées par la Figure (2.14).

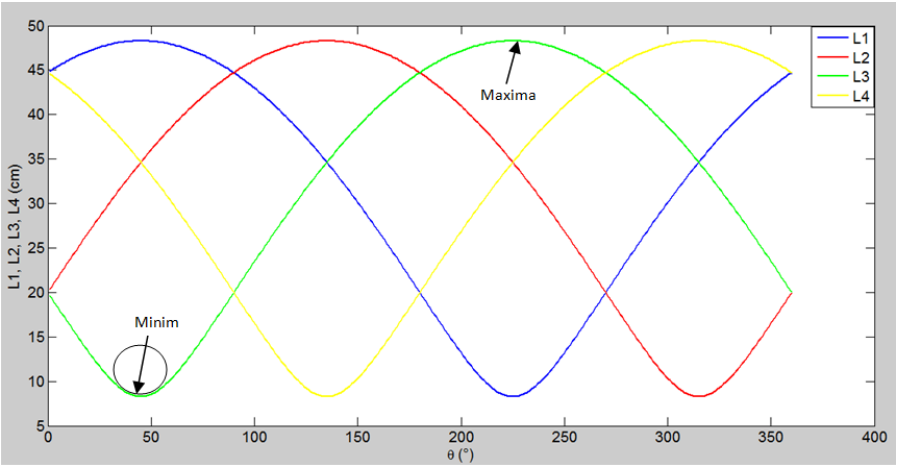


FIGURE 2.14 – Variations des langueurs des 4 câbles du robot pour R = R_{max}

Commentaires :

1. On remarque que les variations des longueurs des câbles sont 2π-périodiques.
2. Les variations des longueurs des 4 câbles sont similaires, et se reproduisent alternativement chaque 90°.
3. Remarquons aussi, que les allures de ces courbes sont non-sinusoïdales,ce qui traduit que les rayons des courbures aux minimas sont inférieurs aux rayons des courbures aux maximas. Cette différence devient plus visible au fur et à mesure que le rayon de la trajectoire circulaire augmente.

La Figure (2.15) permet de visualiser les variations des longueurs des câbles en fonction de l'angle polaire que fait l'organe terminal, présentées sous une forme polaire.

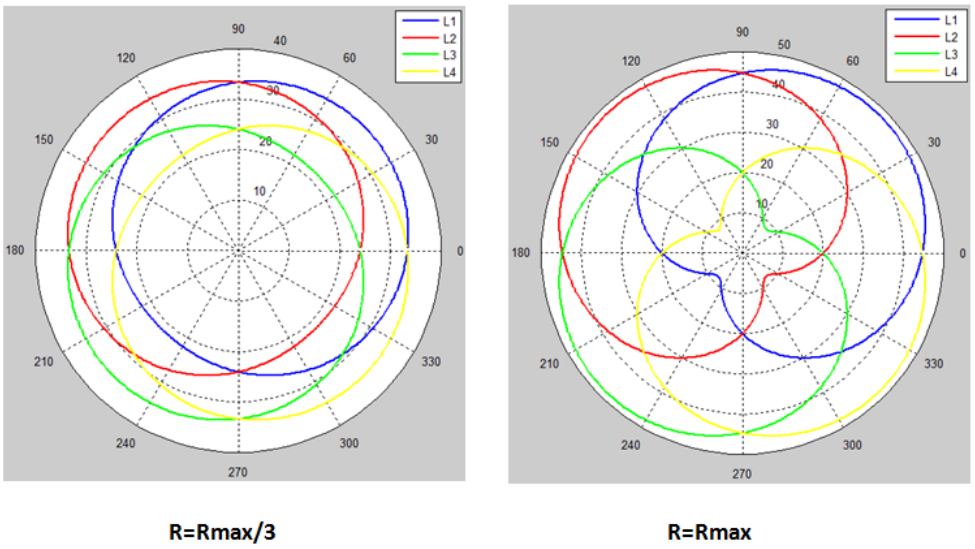


FIGURE 2.15 – variations des longueurs des câbles sous forme polaire

2.5.2.2 B . Variation des angles des câbles par rapport à l'axe des abscisses

La Figure (2.16) montre les variations des angles (θ_1, θ_2, θ_3, θ_4) que font les câbles par rapport l'axe des abscisses en fonction de l'angle polaire que fait l'organe terminal.

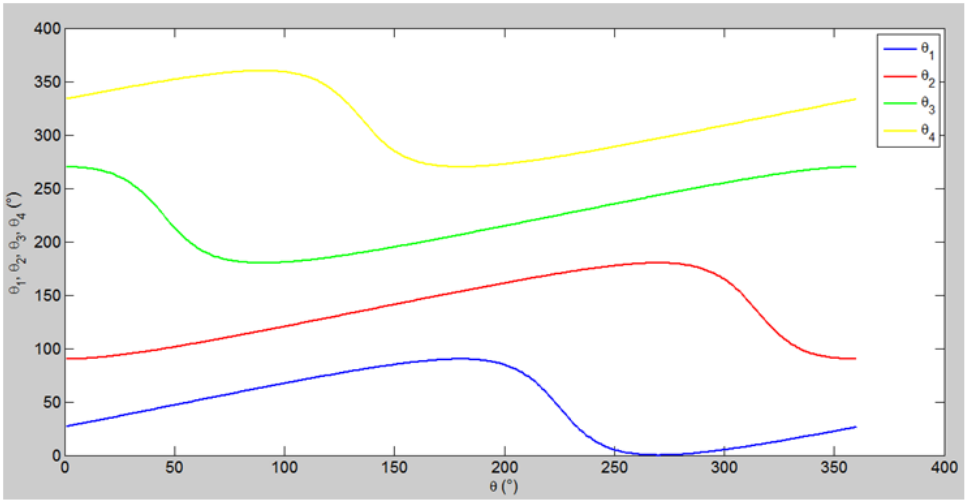


FIGURE 2.16 – Variations des angles des câbles

2.6 Conclusion

Ce chapitre récapitule les calculs de modélisation géométrique, cinématique et dynamique pour un robot à quatre câbles (mouvement suivant deux directions). La modélisation de ce robot est suivie d'une simulation (avec le logiciel Matlab) des déplacements de l'organe terminal, des variations des longueurs et des angles.

Chapitre 3

Analyse statique des forces et calcul des couples optimums

Dans ce chapitre on va étudier la structure des robots parallèles à quatre câbles et présenter l'analyse statique des forces afin d'entamer le calcul des couples optimums appliquées sur les actionneurs de ces robots.

3.1 Analyse statique des force

Dans l'équilibre statique, la somme des forces extérieures exercées sur l'effecteur par les câbles devraient être égaux à la force résultante FR externe exercée sur l'environnement[9]. La Figure (3.1) montre la statique du corps libre pour 4 câbles du CDDR plan.

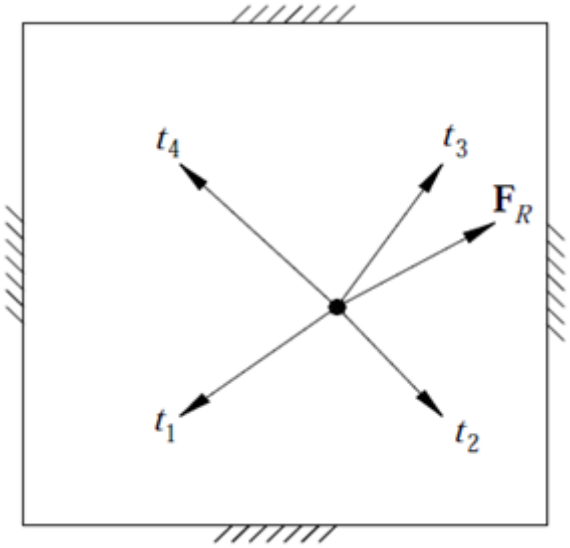


FIGURE 3.1 – Force statique 4 câbles plan CDDRs [9]

L'équation suivante exprime cette relation :

$$\sum_{i=1}^{4} t_i = -\sum_{i=1}^{4} t_i \vec{L_i} = F_R \tag{3.1}$$

ou :

$$L_i = \{cos(\theta_i) \quad sin(\theta_i)\}^T$$

Dans ce cas, la gravité est ignorée parce qu'elle est supposée perpendiculaire au plan du CDDRs. L'équation (3.1) peut être exprimée comme :

$$S * T = F_R \tag{3.2}$$

Ou : $S = \{-\vec{L}_1 \quad -\vec{L}_2 \quad -\vec{L}_3 \quad -\vec{L}_4\}$ est une matrice de dimension (2 * 4)

A partir de l'équation (3.1) on peut l'exprimer par [8] :

$$\begin{pmatrix} -cos\theta_1 & -cos\theta_2 & -cos\theta_3 & -cos\theta_4 \\ -sin\theta_1 & -sin\theta_2 & -sin\theta_3 & -sin\theta_4 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \tag{3.3}$$

L'équation (3.3) est sous-contrainte, ce qui signifie qu'il y a une infini des solutions au vecteur des tensions T des câbles pour exercer la force F_R. Pour inverser cette équation (pour exprimer les tensions des câbles T en fonction de F_R), nous utilisons les notions de solution particulière et homogène :

$$T = S^+ F_R + (I_n - S^+ S)Z \tag{3.4}$$

ou :

I_n : est la matrice d'identité de dimension (4*4) ;

Z : est un vecteur arbitraire de dimension 4 ;

S^+ : est le pseudo-inverse de S par la méthode de Moore-Penrose de dimension (4*2).

Le premier terme de (3.1) est la solution particulière, et le deuxième terme est la solution homogène. Pour la redondance d'actionnement du deuxième degré, une expression équivalente à l'équation (3.4) est :

$$T = \begin{pmatrix} t_{p1} \\ t_{p2} \\ t_{p3} \\ t_{p4} \end{pmatrix} + a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} \tag{3.5}$$

La solution particulière est le premier terme dans (3.5) et la solution homogène est exprimée comme le noyau du vecteur N (n_1, n_2, n_3, n_4) multiplié par un scalaire arbitraire a

plus le vecteur P (P_1, P_2, P_3, P_4) multiplié par un scalaire arbitraire b. Afin de déterminer si un point donné se trouve dans l'espace de travail statique pour un simple CDDRs, si :

$$a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} > \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.6}$$

Divisant l'espace de travail en quatre zones comme illustrer dans la Figure (3.2). pour construire quatre bases possibles,chaque base est formée par deux noyaux(Kernel) N et P.

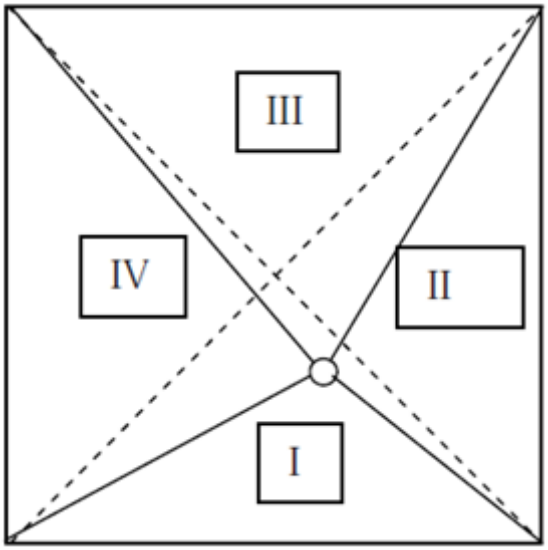


FIGURE 3.2 – Robot à 4 câbles à 4 zones d'espace de travail [9]

Cas I : Supposons que l'organe terminal est dans la première zone. Une base possible pour les noyaux sont :

$$N = \begin{pmatrix} \dfrac{sin(\theta_4 - \theta_2)}{sin(\theta_2 - \theta_1)} \\ \dfrac{sin(\theta_1 - \theta_4)}{sin(\theta_2 - \theta_1)} \\ 0 \\ 1 \end{pmatrix} \qquad P = \begin{pmatrix} \dfrac{sin(\theta_3 - \theta_2)}{sin(\theta_2 - \theta_1)} \\ \dfrac{sin(\theta_1 - \theta_3)}{sin(\theta_2 - \theta_1)} \\ 1 \\ 0 \end{pmatrix} \tag{3.7}$$

Si l'organe terminal se trouve à l'intérieur de la zone I. Les intervalles des angles des câbles sont :

$$0 \leq \theta_1 \leq 45^0 \quad , \quad 135^0 \leq \theta_2 \leq 180^0 \quad , \quad 225^0 \leq \theta_3 \leq 270^0 \quad , \quad 270^0 \leq \theta_4 \leq 315^0$$

Avec :

$$90^0 \leq \theta_4 - \theta_2 \leq 135^0$$

$$90^0 \leq \theta_2 - \theta_1 \leq 180^0$$

$$-270^0 \leq \theta_1 - \theta_3 \leq -180^0$$

$$-315^0 \leq \theta_1 - \theta_4 \leq -225^0$$

Donc toutes les fonctions sinus en (3.7) sont positif ou nul et et n'importe quelle combinaison de N et P (avec a et b positif) comporte toujours des composantes positives comme l'exige (3.6)

En conclusion, la première zone appartient à l'espace de travail statique.

Cas II : Supposons que l'organe terminal est dans la deuxième zone. On peut choisir les noyaux suivants :

$$N = \begin{pmatrix} 0 \\ \frac{sin(\theta_4 - \theta_3)}{sin(\theta_3 - \theta_2)} \\ \frac{sin(\theta_2 - \theta_4)}{sin(\theta_3 - \theta_2)} \\ 1 \end{pmatrix} \qquad P = \begin{pmatrix} 1 \\ \frac{sin(\theta_1 - \theta_3)}{sin(\theta_3 - \theta_2)} \\ \frac{sin(\theta_2 - \theta_1)}{sin(\theta_3 - \theta_2)} \\ 0 \end{pmatrix} \tag{3.8}$$

Si l'organe terminal se trouve à l'intérieur de la zone II. Les intervalles des angles des câbles sont :

$$0 \leq \theta_1 \leq 45^0 \quad , \quad 90^0 \leq \theta_2 \leq 135^0 \quad , \quad 225^0 \leq \theta_3 \leq 270^0 \quad , \quad 315^0 \leq \theta_4 \leq 360^0$$

Avec :

$$45^0 \leq \theta_4 - \theta_3 \leq 135^0$$

$$-225^0 \leq \theta_2 - \theta_4 \leq -180^0$$

$$45^0 \leq \theta_2 - \theta_1 \leq 135^0$$

$$90^0 \leq \theta_3 - \theta_2 \leq 180^0$$

Donc toutes les fonctions sinus en (3.8) sont positif ou nul et et n'importe quelle combinaison de N et P (avec a et b positif) comporte toujours des composantes positives.

En conclusion, la deuxième zone appartient à l'espace de travail statique.

Les deux derniers cas sont similaires.On peut choisir :

$$N = \begin{pmatrix} 0 \\ 1 \\ \frac{sin(\theta_2 - \theta_4)}{sin(\theta_4 - \theta_3)} \\ \frac{sin(\theta_3 - \theta_2)}{sin(\theta_4 - \theta_3)} \end{pmatrix} \qquad P = \begin{pmatrix} 1 \\ 0 \\ \frac{sin(\theta_1 - \theta_4)}{sin(\theta_4 - \theta_3)} \\ \frac{sin(\theta_3 - \theta_1)}{sin(\theta_4 - \theta_3)} \end{pmatrix} \tag{3.9}$$

Constituent une base appropriée pour le secteur III. Et :

$$N = \begin{pmatrix} \frac{sin(\theta_3 - \theta_4)}{sin(\theta_4 - \theta_1)} \\ 0 \\ 1 \\ \frac{sin(\theta_1 - \theta_3)}{sin(\theta_4 - \theta_1)} \end{pmatrix} \qquad P = \begin{pmatrix} \frac{sin(\theta_2 - \theta_4)}{sin(\theta_4 - \theta_1)} \\ 1 \\ 0 \\ \frac{sin(\theta_1 - \theta_2)}{sin(\theta_4 - \theta_1)} \end{pmatrix} \tag{3.10}$$

Constituent une base appropriée pour le secteur IV.

La conclusion pour chaque cas est identique, à savoir les troisième et quatrième zones font également partie de l'espace de travail statique, y compris tous les bords des triangles internes.

Le seul point que nous n'avons pas pris tiennent compte est le centre du carré, mais dans ce cas la base de l'espace vide est constitué par un seul vecteur N = $\{1\ 1\ 1\ 1\}^T$, clairement ce cas particulier est à l'intérieur de l'espace de travail statique car il est facilement vérifie (3.6).

Les bords du carré de base (qui correspond à des singularités cinématiques) et tous les points à l'extérieur de la base carrée sont en dehors de l'espace de travail statique.

3.2 Calcul des couples optimums

Après avoir trouvé à chaque instant les vecteurs { S(X) τ } de la section précédente, on cherche à calculer les couples à fournir par les quatre moteurs[10], sachant que notre système présente une redondance d'actionnement de degré deux, donc pour une position donnée de l'effecteur à l'instant (t), on a une infinité de solutions possibles (infinité de couples possibles).

Les couples seront trouvés après le calcul du vecteur qui représenté la force résultante. Nous devons calculer les couples exigés pour le contrôle.Et en même temps en essayant de maintenir la tension des câbles positive. On utilise la même méthode pour calcul des tensions (notion de solution particulière et homogène) pour le calcul des couples.

$$\tau = S^+Q + (I_n - S^+S)Z \tag{3.11}$$

Où :

I_n : est la matrice identité de dimension (n * n).
Z :est un vecteur arbitraire de dimension n..
S^+ : est la pseudo- inverse de S par la méthode de Moore-Penrose de dimension (n * 4).

L'équivalant de l'équation (3.11) pour une redondance d'actionnement de degré deux est :

$$\tau = \begin{pmatrix} \tau_{p1} \\ \tau_{p2} \\ \tau_{p3} \\ \tau_{p4} \end{pmatrix} + a \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} + b \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \tag{3.12}$$

Le premier terme de l'équation représente la solution particulière.La solution homogène est exprimée par les deux vecteurs N=$\{ \ n_1 \ , \ n_2 \ , \ n_3 \ , \ n_4 \ \}^T$ et P = $\{ \ p_1 \ , \ p_2 \ , \ p_3 \ , \ p_4 \ \}^T$ multipliés par deux scalaires a, b respectivement, tel que N et P sont les noyaux de la matrice jacobienne S.

Le but de calcul des couples consiste à trouver la solution optimale τ_{opt} en respectant :

- chaque composante de τ_{opt} doit être supérieur ou égale à τ_{min} spécifier.
- La norme de τ doit être minimisée.

La première condition assure que les couples des moteurs soient à chaque instant supérieurs à τ_{min} . Si on ne prend pas en considération l'effet dynamique, les tensions dans les câbles doivent être supérieures à r * τ_{min} ; on l'appelle la condition pseudo-statique, si les τ_{min} sont suffisamment grands les câbles resteront sous tension à chaque instant, en réalité à de grandes vitesses et à cause de l'effet dynamique les câbles se relâcheront, dans ce cas la valeur de τ_{min} peut être estimée pour chaque câble en temps réel.

La condition imposée sur le modèle dynamique, est que toutes les tensions des câbles doivent être à chaque instant supérieurs à (T_{min}) dans notre cas $T_{min} = 0$. D'où :

$$T_i = \frac{1}{r}\left(\tau_i - J_i\ddot{\beta}_i - C_i\dot{\beta}_i\right) > 0 \qquad i = 1, 4. \tag{3.13}$$

Donc les couples minimums pour chaque moteur afin de maintenir les tensions positives sont :

$$\tau_{min_i} = max\left\{\left(J_i\ddot{\beta}_i + C_i\dot{\beta}_i\right); 0\right\} \quad i = 1, 4. \tag{3.14}$$

La fonction "max" prend en compte l'effet dynamique du système, le couple minimum exigé pour maintenir les câbles sous tension ne peut pas être négatif car l'équation (3.14) nous donne toujours des valeurs positives.

Le choix du couple optimum se fait à partir de la comparaison des normes des vecteurs couples calculé par la relation :

$$\|\tau(a,b)\| = \sum_{i=1}^{n} \tau_i \tag{3.15}$$

La combinaison de l'équation (3.12) aux résultats de (3.15) nous ramène à un système linéaire non homogène d'inégalité où les scalaires a et b sont inconnus :

$$\tau_{pi} + an_i + bm_i \geq \tau_{min} \qquad i = 1, 4. \tag{3.16}$$

Dans l'espace de la solution $\langle a, b\rangle$ chaque équation de (3.16) représente un semi plan borné par les droites d'équations :

$$\tau_{pi} + an_i + bm_i = \{\tau_{min}\} \qquad i = 1, 4. \tag{3.17}$$

On notera ϕ l'ensemble des points d'intersection de ces droites, il a été démontré que la solution optimal $\langle a, b\rangle$ est l'un des points d'intersection de ces droites [9]. En suite on choisi les points d'intersection qui vérifient :

- Tous les couples supérieurs ou égaux à τ_{min} .
- La norme la plus petite des vecteurs couple.

Donc on à un problème typique de recherche opérationnel c'est la programmation linéaire sous contraintes inégalités dont les variables sont les scalaires a et b avec :

Fonction objectif est :

$$\|\tau(a,b)\| = \sum_{i=1}^{n} \tau_i \qquad i = 1, ..., n = 4. \tag{3.18}$$

Sous les contraintes :

$$\tau_{pi} + an_i + bm_i \geq \tau_{min} \qquad i = 1, 4. \tag{3.19}$$

Pour résoudre ce problème on utilisé deux méthodes, l'une c'est l'algorithme du simplexe et l'autre c'est l'algorithme d'optimisation par essaims de particules.

3.2.1 Calcul des couples optimums par la méthode du Simplexe

3.2.1.1 Optimisation par l'algorithme du simplexe [11]

Cet algorithme, réalisé par George Dantzig à partir de 1947, permet de résoudre efficacement les problèmes de programmation linéaire, bien évidemment quand ces derniers admettent une solution. Ainsi, étant donné un ensemble d'inégalités linéaires sur n variables réelles, l'algorithme permet de trouver la solution optimale pour une fonction objectif, qui est elle aussi linéaire (l'algorithme fonctionne encore quand la fonction est croissante en chacune de n variables).

On définit le polytope comme l'espace convexe contenant l'ensemble des solutions réalisables, dont les arêtes sont les inéquations de contraintes et les sommets les intersections entre inéquations de contrainte.

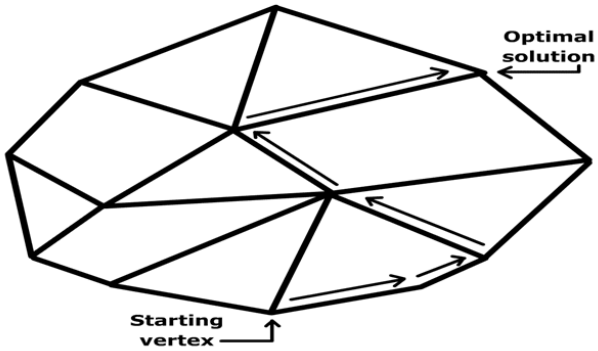


FIGURE 3.3 – Polytope d'un problème de programmation linéaire (source : Wikipedia)

L'algorithme, partant d'un point aléatoire du polytope (tous les points du polytope sont réalisables) appelé solution de base réalisable, progresse à chaque itération vers le point optimal en choisissant parmi les voisins du sommet courant un des sommets qui augmente (ou diminue, selon le sens de l'optimisation) la valeur de la fonction objective. Lorsqu'il n'y a plus de voisin vérifiant cette condition, l'algorithme est terminé et l'on a trouvé le point optimal.

Lorsque plusieurs voisins différents optimisent la fonction objectif, un choix doit être effectué entre eux et la règle de sélection qui permet de trancher est appelée règle de pivotage. Le choix de cette règle est capital afin d'assurer l'efficacité de l'algorithme.

L'algorithme se termine nécessairement car le nombre de sommets du polytope est fini. Un nombre fini d'inéquations de contraintes implique un nombre fini d'intersections entre ces mêmes inéquations si les inéquations sont distinctes deux à deux. De plus, il n'y a pas de boucle dans le parcours (chaque arête parcourue augmente la fonction objective), et il n'existe pas de maximum local qui ne soit pas un maximal global.

Au final l'algorithme se termine en un sommet du polytope qui constitue le sommet optimal, solution au problème de programmation linéaire. Bien que l'algorithme se termine de façon certaine en un temps fini, certains problèmes peu classiques demanderont un temps exponentiel avant d'être résolus.

3.2.1.2 Les Étapes de l'algorithme du simplexe

Notre problème d'optimisation consiste minimisé la fonction objectif (3.15) et à trouver a,b optimums sous les contraintes (3.17) , donc on peut écrire le problème sous la forme [18] :

$$min\left(\|\tau(a,b)\| = \sum_{i=1}^{n} \tau_i\right) \qquad i = 1, ..., n = 4. \tag{3.20}$$

Sous contraintes inégalités :

$$\begin{cases} \tau_{p1} + a.n_1 + b.m_1 \geq \tau_{min} \\ \tau_{p2} + a.n_2 + b.m_2 \geq \tau_{min} \\ \tau_{p3} + a.n_3 + b.m_3 \geq \tau_{min} \\ \tau_{p4} + a.n_4 + b.m_4 \geq \tau_{min} \end{cases} \quad avec \quad (a,b) \geq 0 \tag{3.21}$$

Étape 1 : Mise sous forme standard

Avant que l'algorithme du simplexe puisse être utilisé pour résoudre un problème de programmation linéaire, ce dernier doit être converti en un programme équivalent où toutes les contraintes technologiques sont des équations et toutes les variables sont non négatives. Pour les contraintes de type ($\geq$) : Pour chaque contrainte de ce type, on retranche une variable d'excédent e_i , tel que e_i, est une variable positive ou nulle. Donc l'équation (3.21) sera :

$$\begin{cases} \tau_{p1} + a.n_1 + b.m_1 - e_1 = \tau_{min} \\ \tau_{p2} + a.n_2 + b.m_2 - e_2 = \tau_{min} \\ \tau_{p3} + a.n_3 + b.m_3 - e_3 = \tau_{min} \\ \tau_{p4} + a.n_4 + b.m_4 - e_4 = \tau_{min} \end{cases} \quad avec \quad (a,b) \geq 0 \tag{3.22}$$

Étape 2 : Implémentation d'algorithme dans un calculateur [19]

Quand le problème est mis sous forme standard, on peut appliquer l'algorithme du simplexe. Les différentes équations linéaires sont placées dans un tableau. Chaque équation est une ligne du tableau (voir la figure (3.4)).

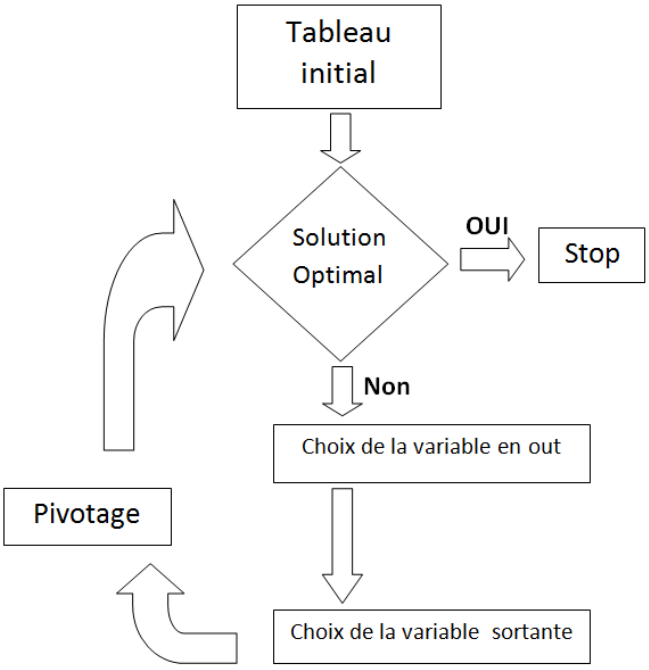


FIGURE 3.4 – Schéma d'implémentation de l'algorithme Simplexe

3.2.2 Optimisation par essaims de particules

3.2.2.1 Un peu d'historique [13]

L'optimisation par Essaim de particule (OEP) ou bien (PSO Particle swarm optimization), a été inventée par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995. Au départ J. Kennedy et R. Eberhart cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale. Le modèle qu'ils ont proposé à ensuite été étendu en un algorithme simple et efficace d'optimisation.

3.2.2.2 Définitions [14]

Optimisation Par Essaims de Particule :

L'optimisation par Essaim de particule (OEP) est une technique utilisée pour explorer l'espace de recherche d'un problème quelconque pour trouver l'ensemble des paramètres qui maximise/minimise un objectif particulier. Cet objectif est atteint en suivant un algorithme dédié que l'on verra par la suite.

Notion de voisinage :

Le voisinage constitue la structure du réseau social. Les particules à l'intérieur d'un voisinage communiquent entre-elles. En général, pour une nuée d'oiseaux, le voisinage suit trois types de topologies :

- Topologie en étoile (Figure (3.5)(a)) : le réseau social est complet, donc une communication complète et une attirance vers la meilleure particule.
- Topologie en anneau (Figure (3.5)(b)) : chaque particule communique avec n voisines immédiates. Chaque particule tend à se déplacer vers la meilleure dans son voisinage local.
- Topologie en rayon (Figure (3.5)(c)) : une particule "centrale" est connectée à toutes les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres.

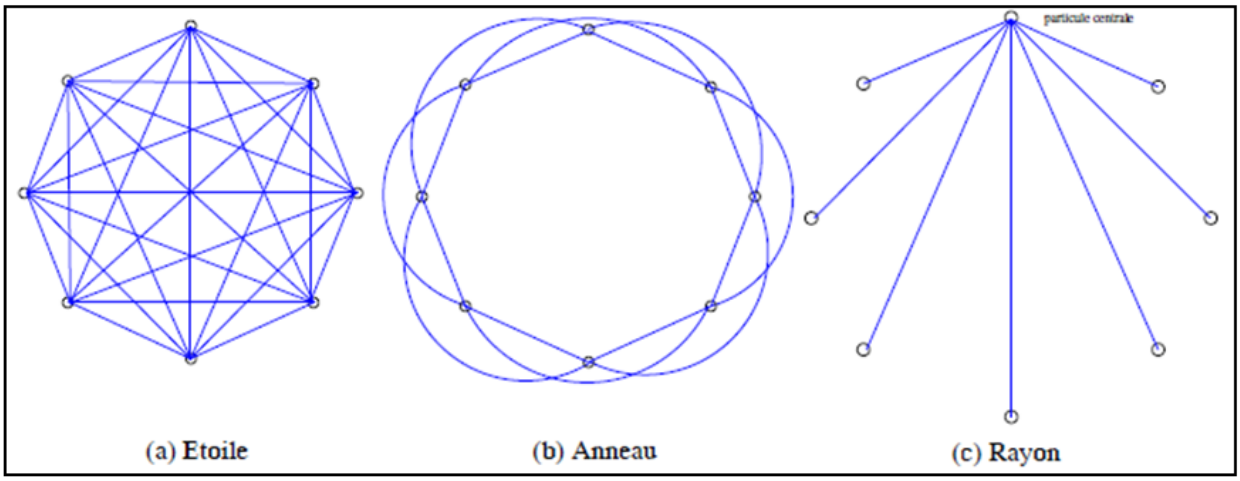


FIGURE 3.5 – différents types de topologie pour un essaim de particules [14]

3.2.2.3 L'algorithme PSO [15]

Chaque particule représente une solution potentielle dans l'espace de recherche. La nouvelle position d'une particule est déterminée en fonction de sa propre valeur et celle de ses voisines. Soit $\overrightarrow{x_i}$ (t) la position de la particule P_i au temps t, sa position est modifiée en ajoutant une vitesse $\overrightarrow{v_i}$ (t) à sa position courante :

$$\overrightarrow{x_i}(t) = \overrightarrow{x_i}(t-1) + \overrightarrow{v_i}(t) \tag{3.23}$$

La vitesse de chaque particule est mise à jour suivant l'équation suivante :

$$v_i(t+1) = \underbrace{\omega v_i(t)}_{Inertie} + \underbrace{c_1 r_1[x_{pi}(t) - x_i(t)]}_{composante\ cognitive} + \underbrace{c_2 r_2[g(t) - x_i(t)]}_{composante\ sociale} \tag{3.24}$$

v_i (t) est la vitesse de particule i à l'instant t et x_i (t) est la position de particule i à l'instant t, les paramètres w, c_1, et c_2 ($0 \leq w \leq 1.2$, $0 \leq c_1 \leq 2$, et $0 \leq c_2 \leq 2$) sont des coefficients constants fixés par l'utilisateur, r_1 et r_2 sont des nombres aléatoires tirés à chaque

itération, g(t) est la meilleure solution trouvée jusqu'à l'instant t et x_{pi} (t) est la meilleure solution trouvée par le particule i. v_i (t), c'est le vecteur vitesse qui dirige le processus de recherche et reflète la "sociabilité" des particules.

Les variables et paramètres de l'algorithme :

- N : Nombre de particules ;
- $\overrightarrow{x_i}$ (t) : Position de la particule P_i ;
- $\overrightarrow{v_i}$ (t) : Vitesse de la particule P_i ;
- P_{best_i} : Meilleure finesse obtenue pour la particule P_i ;
- $\overrightarrow{x_{p_{best_i}}}$: Position de la particule P_i pour la meilleure finesse ;
- $\overrightarrow{x_{g_{best_i}}}$: Position de la particule ayant la meilleure finesse de toutes ;
- ρ_1 , ρ_2 : Valeurs aléatoires positives.

Initialisations :

Initialiser aléatoirement la population.

Traitement :

Si $(F(\overrightarrow{x_i}) > p_{best_i})$ Alors

$p_{best_i} \longleftarrow F(\overrightarrow{x_i})$

$\overrightarrow{x_{p_{best_i}}} \longleftarrow \overrightarrow{x_i}$

Fin Si

$Si(F(\overrightarrow{x_i}) > g_{best_i})$ Alors

$g_{best_i} \longleftarrow F(\overrightarrow{x_i})$

$\overrightarrow{x_{g_{best_i}}} \longleftarrow \overrightarrow{x_i}$

Fin Si

Fin Pour

Pour i de 1 à N faire

$\overrightarrow{v_i} \longleftarrow \overrightarrow{v_i} + \rho_1(\overrightarrow{x_{p_{best_i}}} - \overrightarrow{x_i}) + \rho_2\overrightarrow{x_{g_{best_i}}} - \overrightarrow{x_i})$

$\overrightarrow{x_i} \longleftarrow \overrightarrow{x_i} + \overrightarrow{v_i}$

Fin Pour

Jusqu'à ce que (le processus converge)

La variation de la vitesse est proportionnelle à l'éloignement d'une solution par rapport à la solution globale.

Les variables aléatoires ρ_1 et ρ_2 peuvent être définies de la façon suivante :

$$\begin{cases} \rho_1 = r_1 c_1 \\ \rho_2 = r_2 c_2 \end{cases}$$

r_1 et r_2 suivent une loi uniforme sur [0...1] et c_1 et c_2 sont constantes et représentent une accélération positive, avec $c_1 + c_2 < 4$.

Le critère de convergence peut être un nombre fixe d'itérations, suivant la fitness ou bien la variation lorsqu'elle tend vers 0.

On remarque qu'il y a six paramètres qui rentrent en ligne de compte :

1. La dimension du problème.
2. Le nombre de particules.
3. Les valeurs des coefficients ρ.
4. La taille du voisinage.
5. La vitesse maximale.
6. L'inertie.

La vitesse peut être limitée par une vitesse maximale V_{max} et une vitesse minimale V_{min} pour éviter que les particules se déplacent trop rapidement ou trop lentement d'une région à une autre dans l'espace de recherche.

Un facteur d'inertie Φ peut être appliqué à la vitesse pour contrôler l'influence de celle-ci.

3.2.2.4 Application de l'algorithme PSO pour le calcul des couples optimums

Pour appliquer l'algorithme de PSO dans notre problème de recherche des couples optimums on va chercher les couples optimums à travers la recherche de deux scalaires a,b qui donnent le minimum de la fonction objective (3.16) (dans la théorie de PSO sera appelé fonction de fitness), tout on restant dans l'espace admissible défini par les contraintes (2021), pour une explication claire on va proposer un exemple numérique de calcul des couples optimums par PSO.

Exemple :

Considérons le robot à 4 câbles avec deux dégrées de redondances, on donne la force appliquée sur l'effecteur est $F = \begin{pmatrix} -1.03 \\ 1.05 \end{pmatrix} (N)$, et on va déplacer l'effecteur vers la positon

X=$\begin{pmatrix} 0.04 \\ -0.23 \end{pmatrix}$ (m) avec le couple minium est $\{\tau_{min}\}_i = 0.10\ (N_m)$, i=1,4.

Pour les données précédentes la matrice Jacobienne S(X) et les noyaux (N, P) de S(X) donnés par :

$$S(X) = \begin{bmatrix} -0.97 & 0.95 & 0.46 & -0.56 \\ -0.26 & -0.32 & 0.89 & 0.83 \end{bmatrix} \qquad \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} = \begin{bmatrix} -0.74 \\ -0.64 \\ -0.48 \\ 0.11 \end{bmatrix} \qquad \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} -0.35 \\ -0.69 \\ 0.29 \\ -0.79 \end{bmatrix}$$

On utilise la méthode de Moore Penrose pour déterminer la solution particulière

$$\begin{bmatrix} \tau_{P1} \\ \tau_{P2} \\ \tau_{P3} \\ \tau_{P4} \end{bmatrix} = \begin{bmatrix} -0.35 \\ -0.69 \\ 0.29 \\ -0.79 \end{bmatrix}$$

Pour trouver les couples optimums il suffit de trouver les deux scalaires a,b qui minimisent la fonction de fitness :

$$min\left(\|\tau(a,b)\| - \sum_{i=1}^{n} \tau_i\right) \qquad i = 1, ..., n = 4. \tag{3.25}$$

Sous les contraintes :

$$\begin{cases} \tau_{p1} + a.n_1 + b.m_1 \geq \tau_{min} \\ \tau_{p2} + a.n_2 + b.m_2 \geq \tau_{min} \\ \tau_{p3} + a.n_3 + b.m_3 \geq \tau_{min} \\ \tau_{p4} + a.n_4 + b.m_4 \geq \tau_{min} \end{cases} \quad avec \quad (a,b) \geq 0 \tag{3.26}$$

La figure (3.6) suivante montre l'espace admissible de solution :

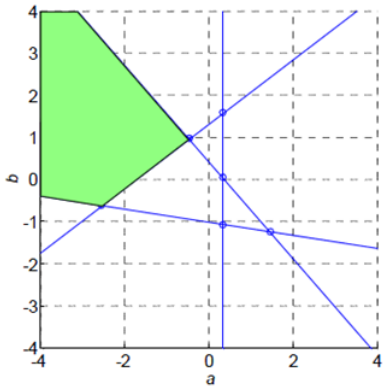


FIGURE 3.6 – $\langle a, b \rangle$ Espace admissible et contraintes d'inégalité avec intersections

Implémentation de l'algorithme de PSO pour trouver les a,b optimums :

L'algorithme PSO décrit ci-dessous est celui qui considère le voisinage d'une particule comme étendu à toute la population (Topologie en anneau) :

1. Création d'une population P de N particules réparties uniformément sur l'espace de recherche.
2. Chaque particule pi ∈ P est évaluée à l'aide de la fonction de fitness (3.16) mesurant l'adéquation de cette solution potentielle avec le problème.
3. Si la position x_i de la particule i est meilleure au sens de la fonction de fitness (3.16) que sa meilleure position jamais rencontrée p_{best_i} , mettre à jour p_{best_i}.
4. Déterminer la meilleure particule $\overrightarrow{x_{g_{best_i}}}$ parmi la population courante.
5. Mettre à jour la vitesse v_i (t) de chaque particule i selon la règle suivante :

$$v_i(t+1) = \underbrace{\omega v_i(t)}_{Inertie} + \underbrace{c_1 r_1[x_{pi}(t) - x_i(t)]}_{composante\ cognitive} + \underbrace{c_2 r_2[g(t) - x_i(t)]}_{composante\ sociale} \tag{3.27}$$

   Où ω, c_1 et c_2 sont les paramètres de l'algorithme qui règlent l'inertie de la particule ou l'influence des différents attracteurs. r_1 et r_2 sont deux variables aléatoires uniformes sur [0 , 1].
6. Déplacer les particules à leurs positions $\overrightarrow{x_i}$ (t) selon la règle suivante :

$$\overrightarrow{x_i}(t) = \overrightarrow{x_i}(t-1) + \overrightarrow{v_i}(t)$$

7. Reboucler sur la seconde étape (itération t + 1) jusqu'à ce qu'un critère de fin soit vérifié.

La Figure (3.7) illustre la stratégie de déplacement d'une particule.

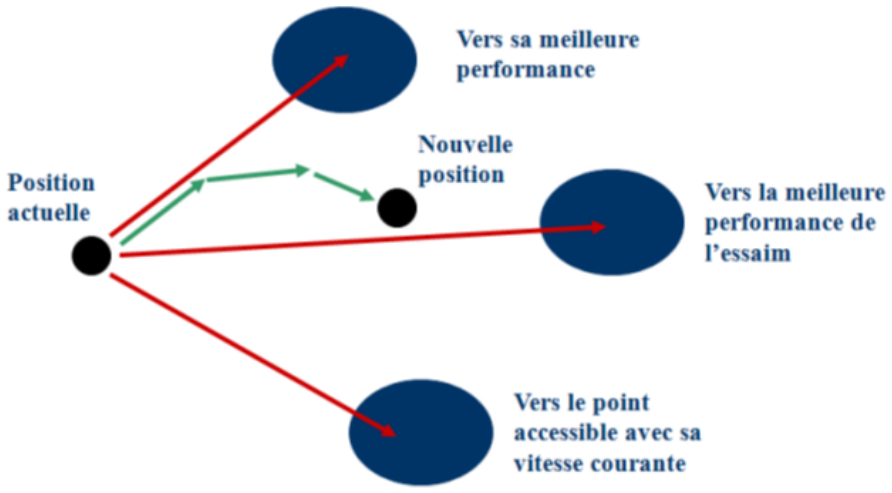


FIGURE 3.7 – Déplacement d'une particule

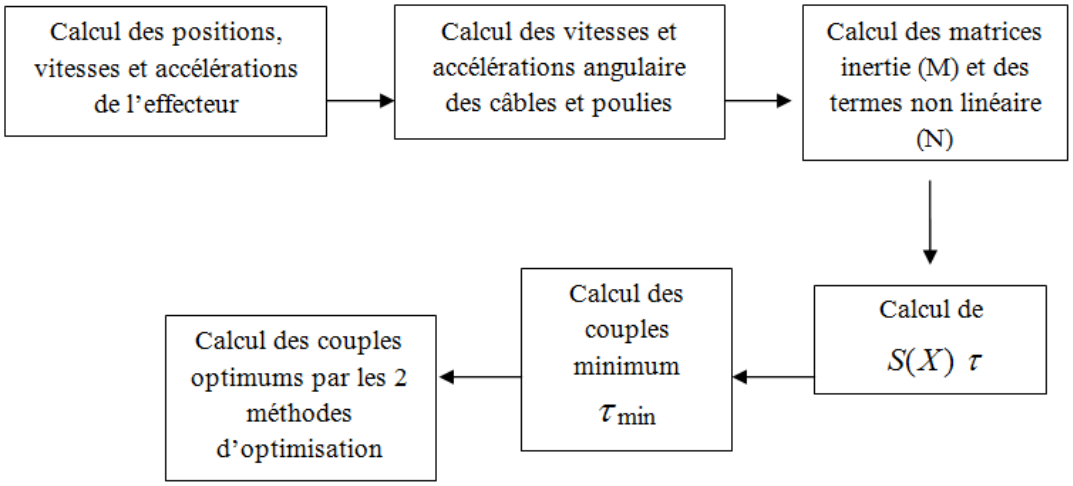
Après l'implémentation de cette algorithme sous logiciel Matlab, on a trouvé les valeurs a, b suivantes :

$$\langle a, b \rangle = (-0.4496, 0.9677)$$

Les résultats de notre simulation sont relativement les même que ceux obtenus par (Robert L. Williams II 2003) Université de l'Ohio [10]. Dans la recherche de a,b par la méthode graphique (a,b) = (-0.450,0.970).

3.2.3 Applications des Algorithmes d'optimisation

Les étapes de calculs de notre programme réalisé sur Matlab sont schématisées ci-dessous :



3.2.3.1 Application des algorithmes d'optimisation sur une trajectoire circulaire :

Cette partie présente un exemple de contrôle dynamique en boucle ouverte du robot à 4 câbles avec un degré de redondance de deuxième ordre. Cet exemple est exécuté selon l'architecture de contrôle avec calcul en temps réel des couples minimums, afin de maintenir les câbles sous tension à chaque instant en utilisant les algorithmes pour trouver les couples optimums.

La base carré est de côté L_b = 0,658 m, le rayon du cercle à effectuer est R = 0,2165 m, La figure (3.8).

Schématise le modèle pris en considération.

L'angle polaire ϕ est définit comme étant un paramètre indépendant du cercle, il est mesuré par rapport à l'axe des x dans le sens antihoraire, ϕ est donnée de 0 à π, pour retrouver les paramètres de la deuxième moitié du cercle on utilise la symétrie

$$\alpha = \ddot{\phi} = 8\pi, \quad \dot{\phi} = \alpha.t, \quad \phi = \alpha.t^2/2$$

Pour comparer nos résultats avec ceux obtenus par (Robert L. Williams II 2003) de l'Université d'Ohio [9], on a utilisé les même données suivantes :

- Masse : m = 1 kg ;
- Inertie poulie/rotor : J_i = 0,0008 kg.m^2 ;
- Coefficient de frottement : C_i = 0,01 N.m.s ;
- Rayon des poulies : r_i = 5 c.m.

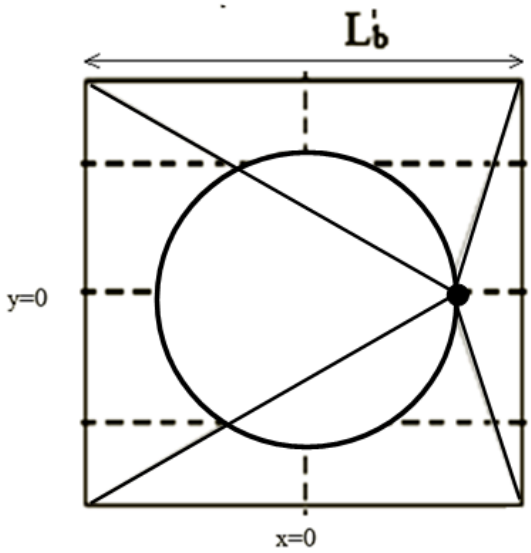


FIGURE 3.8 – Trajectoire de l'effecteur

La figure (3.9) montre les couples minimums à chaque instant, pour que les câbles soient sous tension durant l'exécution de la trajectoire.

En prenant en considération les couples min calculés à chaque instant qui sont tous supérieurs ou égale à un couple donné 0.05 (N.m), les quatre couples développés par les moteurs doivent être tous supérieurs aux couples minimums,

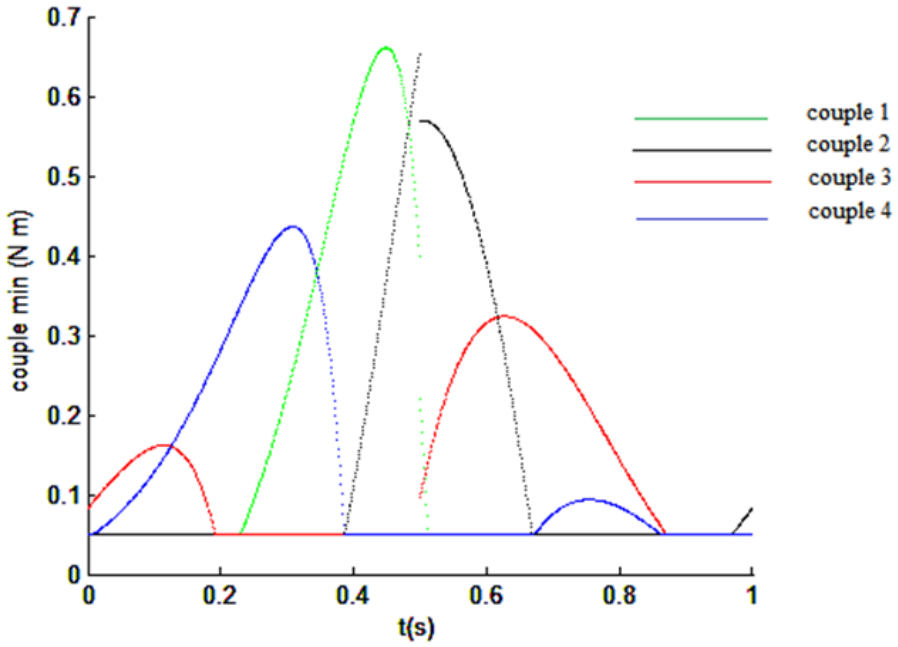


FIGURE 3.9 – Couples minimums des quatre moteurs

Par la méthode d'optimisation du simplexe on obtient les couples optimums représentés dans la figure (3.10).

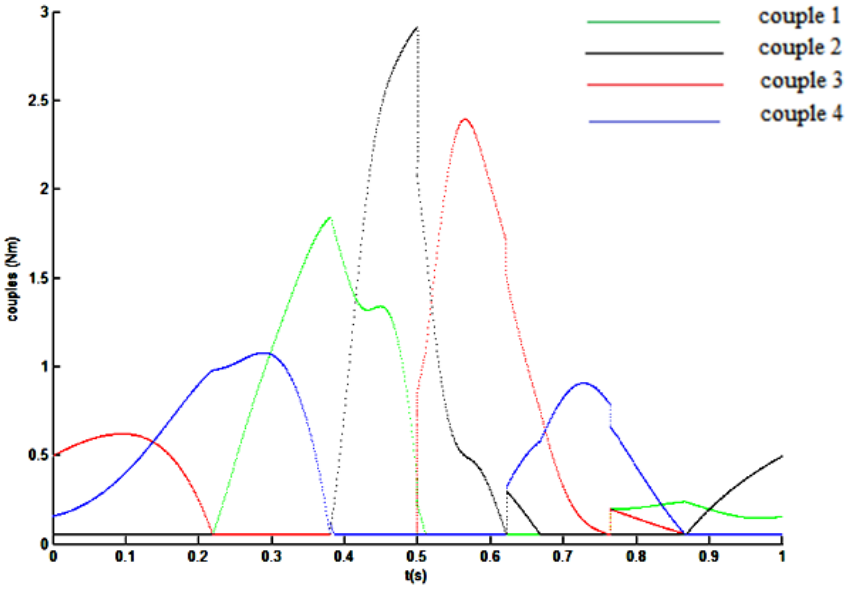


FIGURE 3.10 – Couples optimums des quatre moteurs par l'algorithme du Simplexe

Par la méthode d'optimisation PSO on obtient les couples optimums représentés dans la figure (3.11).

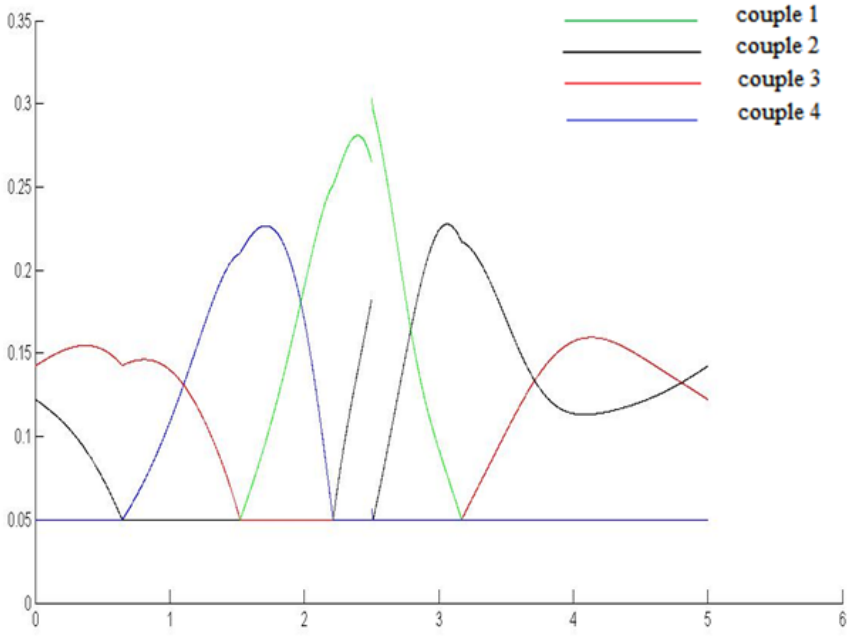


FIGURE 3.11 – Couples optimums des quatre moteurs par l'algorithme du PSO

On remarque que le calcul des couples optimums par l'algorithme d'essaim de particulaire a donné les meilleurs résultats que l'algorithme d'optimisation du Simplexe par rapport au critère (3.16), mais l'inconvénient majeur de cette méthode est le temple de calcul qui est relativement grand, la chose qui nous a obligé à utiliser la méthode du Simplexe car elle est simple à implémenter dans une carte de commande et aussi on n'aura pas un problème de choix des paramètres de l'algorithme.

Les résultats de notre simulation des couples optimums par la programmation linéaire (Méthode du Simplexe) sont relativement les mêmes que ceux obtenus par (Robert L. Williams II 2003) à l'Université de l'Ohio (voir figure (3.12)). Dans ce qui suit les trajectoires seront définies par des équations polynomiales.

Dans tout le reste du travail on va utiliser l'algorithme du Simplexe pour le calcul des couples optimums.

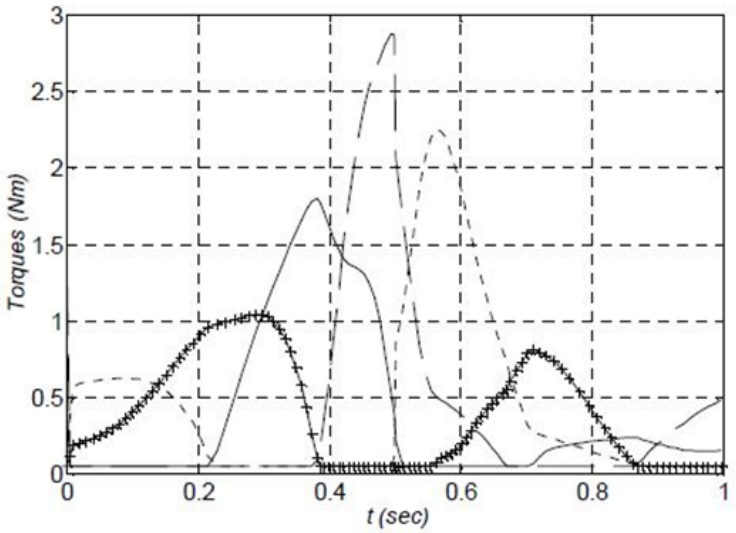


Figure 11b. Actual Actuator Torques
τ_1 (solid), τ_2 (long dash), τ_3 (short dash), τ_4 (*)

FIGURE 3.12 – Les résultats de simulation des couples optimums par la méthode du Simplex Obtenus par (Robert L. Williams II 2003) Université d'Ohio, Canada [10].

Chapitre 4

Commande en Boucle Ouverte

4.1 Introduction

Dans cette section, on commencera par la présentation de l'équation dynamique du robot à 4 câbles ainsi que sa représentation d'état. Ensuite, on simulera les réponses en boucle ouverte dans le cas d'une entrée impulsionnelle, indicielle et pour des impulsions de durée Taux.

4.2 Établissement de de l'équation dynamique du système à 4 câbles

Comme on a déjà vue dans la section $(II-30)$ l'équation dynamique du système à quatre câbles est donnée par :

$$M(x)\ddot{X} + N(X,\dot{X}) = S(X)\tau \tag{4.1}$$

D'où :

$$\ddot{X}(t) = -M^{-1}(X)N(X,\dot{X}) + M^{-1}(X)S(X)\tau \tag{4.2}$$

Où :

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \tag{4.3}$$

4.3 Établissement de la représentation d'état du système à 4 câbles

A partir de l'équation dynamique ($IV.1$) on effectue les changements suivants :

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = \dot{x}_1(t) \\ x_3(t) = y(t) \\ x_4(t) = \dot{x}_3(t) \end{cases} \Longrightarrow \begin{cases} \dot{x}_1(t) = x_2(t) \\ M_{11}\dot{x}_2(t) + M_{12}\dot{x}_4(t) = u_1(t) \\ x_3(t) = y(t) \\ x_4(t) = \dot{x}_3(t) \end{cases} \tag{4.4}$$

Donc la représentation d'état peut être écrite comme suit :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} \\ 0 & 0 & 1 & 0 \\ 0 & M_{21} & 0 & M_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -N_{11} & 0 & -N_{12} \\ 0 & 0 & 0 & 1 \\ 0 & -N_{21} & 0 & -N_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ u_1(t) \\ 0 \\ u_2(t) \end{bmatrix} \tag{4.5}$$

Où :

$$M_{2d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & M_{11} & 0 & M_{12} \\ 0 & 0 & 1 & 0 \\ 0 & M_{21} & 0 & M_{22} \end{bmatrix}, N_{2d} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -N_{11} & 0 & -N_{12} \\ 0 & 0 & 0 & 1 \\ 0 & -N_{21} & 0 & -N_{22} \end{bmatrix} U(t) = \begin{bmatrix} 0 \\ u_1(t) \\ 0 \\ u_2(t) \end{bmatrix} \tag{4.6}$$

Il en résulte alors l'équation d'état suivante :

$$\dot{X}(t) = f(X) + g(X) * U(t) \tag{4.7}$$

Avec :

- $\dot{X}(t)$: est le vecteur d'état du système ;
- $f(X)$ et $g(X)$: sont des fonctions non linéaires ;
- $U(t)$: est le vecteur de commande.

4.4 Etude de la stabilité du système en boucle ouverte

Dans notre cas les paramètres du système sont non linéaires, pour cette raison on adopte une des principales méthodes d'étude des systèmes non linéaires qui est la méthode de la stabilité locale ou la stabilité autour d'un point de fonctionnement.

4.4.1 Stabilité locale par la méthode de Lyapnouv

En mathématiques et en automatique, la notion de stabilité de Lyapunov (ou, plus correctement, de stabilité au sens de Lyapunov) apparait dans l'étude des systèmes dynamiques[21]. De manière générale, la notion de stabilité joue également un rôle en mécanique, dans les modèles économiques, les algorithmes numériques, la mécanique quantique, la physique nucléaire ...etc. point d'équilibre x_0.
Si on considère le système non linéaire.
$\dot{X}(t) = f(X)$
Au voisinage de x_0 le point d'équilibre x_0.On a : $x = x_0 + \partial x_0$
Le développement en série de Taylor de f au voisinage de x_0 donne :
$f(x_0 + \partial x_0) = f(x_0) + \left(\frac{\delta f}{\delta x}\right) x_0(\partial x) + g(\partial x)$
On aura alors :
$\dot{X} = f(x_0 + \partial x_0) = \dot{x}_0 + \partial \dot{x}_0 = f(x_0) + \left(\frac{\delta f}{\delta x}\right) x_0(\partial x) + g(\partial x)$
De plus, on peut simplifier en écrivant :
$\dot{X}_0 = f(x_0)$
Donc le système non linéaire devient :
$\partial \dot{x}_0 = \left(\frac{\delta f}{\delta x}\right) x_0(\partial x) + g(\partial x) = A\partial x + g(\partial x)$
Avec :
$A\partial x$: Partie Linéaire.
$g(\partial x)$: Partie non Linéaire.
Le 1^{er} théorème de Lyapnouv nous permet d'étudier la stabilité des systèmes non linéaires qui ramène à un système linéaire (facile à étudier par la suite via les valeurs propres) [21].
Notons également que :

1. Si la partie linéaire est localement exponentiellement stable (E.S), c'est-à-dire les parties réelles des valeurs propres sont négatives, $Re(vp(A) < 0)$, alors on peut dire que le système non linéaire est localement asymptotiquement stable (L.A.S).
2. Si la partie linéaire est instable autour du point de fonctionnement (L.IS) c'est-à-dire qu'il existe une valeur propre de A dans la partie réelle est positive, alors le système non linéaire est instable (L.IS).

4.4.2 Application du 1^{er} théorème de Lyapnouv

Afin d'appliquer la méthode de Lyapnouv qui consiste à la linéarisation de l'équation (7.1) autour d'un point de fonctionnement , il faut trouver ces points de fonctionnement tel que :
$f(x) = 0$

Pour obtenir les points de fonctionnement ou les points singuliers, on peut réécrire :

$$f(x) = 0 \Longleftrightarrow \begin{cases} f_1(x) = 0 \\ f_2(x) = 0 \\ f_3(x) = 0 \\ f_4(x) = 0 \end{cases} \tag{4.8}$$

La solution de cette équation oblige que les vitesses (x_1 et x_4) soient nulles, avec des positions quelconque. Donc on trouve une infinité de points de fonctionnement. Pour cette raison là, on a choisi le point milieu comme un point singulier qui a comme valeur : $(0.001, 0, 0, 0)$. Ce point nous permet de calculer la matrice linéaire A dont la forme est donnée par :

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{(0.001,0,0,0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{M_{12}N_{21} - M_{22}N_{11}}{M_{22}M11 - M_{12}M_{21}} & 0 & \frac{M_{12}N_{22} - M_{22}N_{12}}{M_{22}M_{11} - M_{12}M_{21}} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{M_{11}N_{21} - M_{21}N_{11}}{M_{22}M_{11} - M_{12}M_{21}} & 0 & \frac{M_{11}N_{22} - M_{21}N_{12}}{M_{22}M_{11} - M_{12}M_{21}} \end{bmatrix} \tag{4.9}$$

Les valeurs propres de A sont calculées à l'aide :
$\mid \lambda I - A \mid = 0$
On a les valeurs suivantes :
$\lambda_1 = \lambda_2 = 0$
$\lambda_3 = -2.7171$
$\lambda_4 = -5.3875$
λ_3,λ_4 : remarquons que ce sont des nombres réels négatifs. on peut en conclure alors que notre système est stable autour de ce point.

4.5 Simulation de la Réponse du Robot

Dans cette partie on s'intéresse à simuler la réponse du robot à 4 câbles en utilisant la méthode numérique de Runge Kutta, à cause de la non linéarité qu'il présente le système étudié, pour les conditions suivantes :

- Les inerties du rotor et de la poulie de chaque moteur : $J_i(i = 1 : 4) = 0.0008$ kg.m^2 ;
- Les coefficients d'amortissement visqueux de chaque arbre du moteur : $C_i(i = 1 : 4) = 0.01$ N.m.s ;

- La masse de l'organe $m = 1$ k ;
- Le rayon de la poulie de chaque moteur $r_i(i = 1 : 4) = 0.05$ m .

Tout en supposant que le repère de notre système est au point milieu.

4.5.1 Pour une entrée impulsionelle

Pour simuler cette entrée on suppose que les forces F_x et F_y appliquées sur l'effecteur terminal soient égales à 1 N , pour une durée très courte.
Les profils des positions (x) et (y) en boucle ouverte sont présentés dans les Figures (4.1) et (4.2) respectivement.

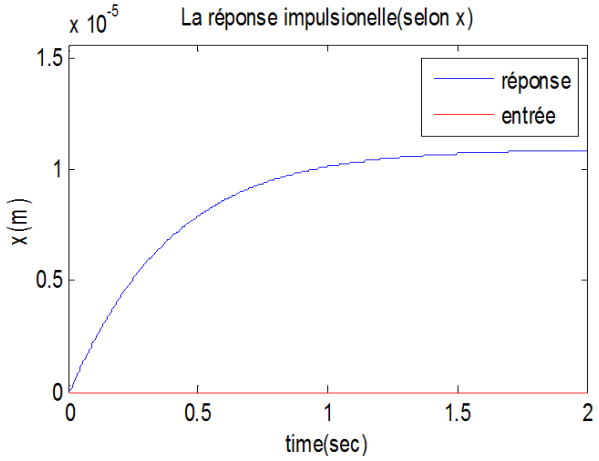


FIGURE 4.1 – La réponse impulsionnelle du robot à 4 câbles (selon l'axe des x)

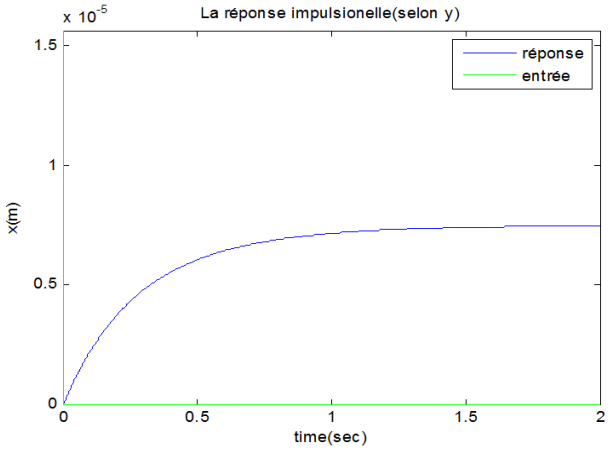


FIGURE 4.2 – La réponse impulsionnelle du robot à 4 câbles (selon l'axe des y)

4.5.2 Pour une entrée indicielle

Pour simuler cette entrée on suppose que les forces F_x et F_y appliquées sur l'effecteur terminal soient égales à 1 N, pour une durée non limitée.
Les profils des positions (x) et (y) en boucle ouverte sont présentés dans les Figures (4.3) et (4.4)respectivement.

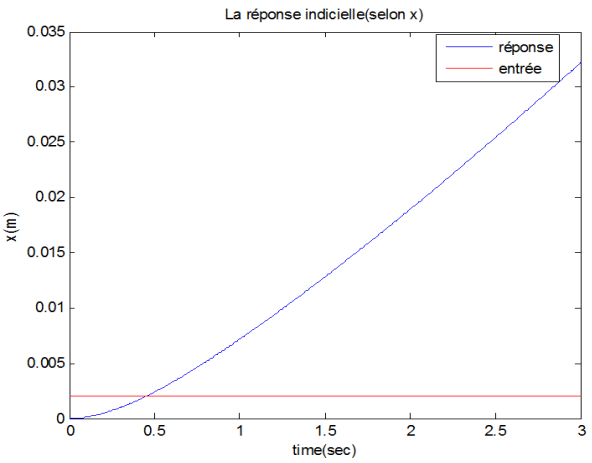


FIGURE 4.3 – La réponse indicielle du robot à 4 câbles (selon l'axe des x)

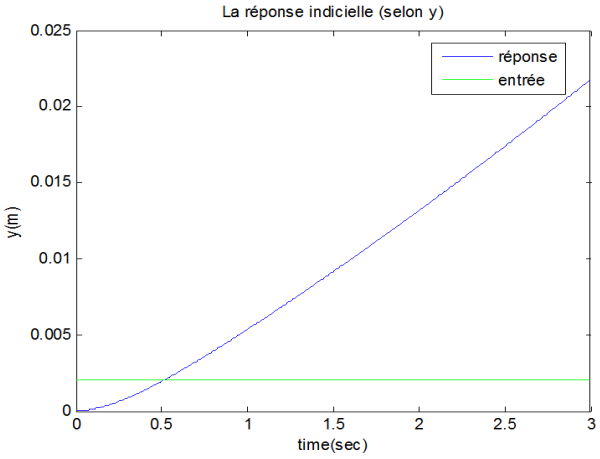


FIGURE 4.4 – La réponse indicielle du robot à 4 câbles (selon l'axe des y)

4.5.3 Pour une entrée indicielle de durée τ

Pour simuler cette entrée on suppose que les forces F_x et F_y appliquées sur l'effecteur terminal soient égales à 1 N, pour une durée τ limitée avec $\tau = 30$ ms.
Le profil de position (x) en boucle ouverte est présenté dans la Figure (4.5).

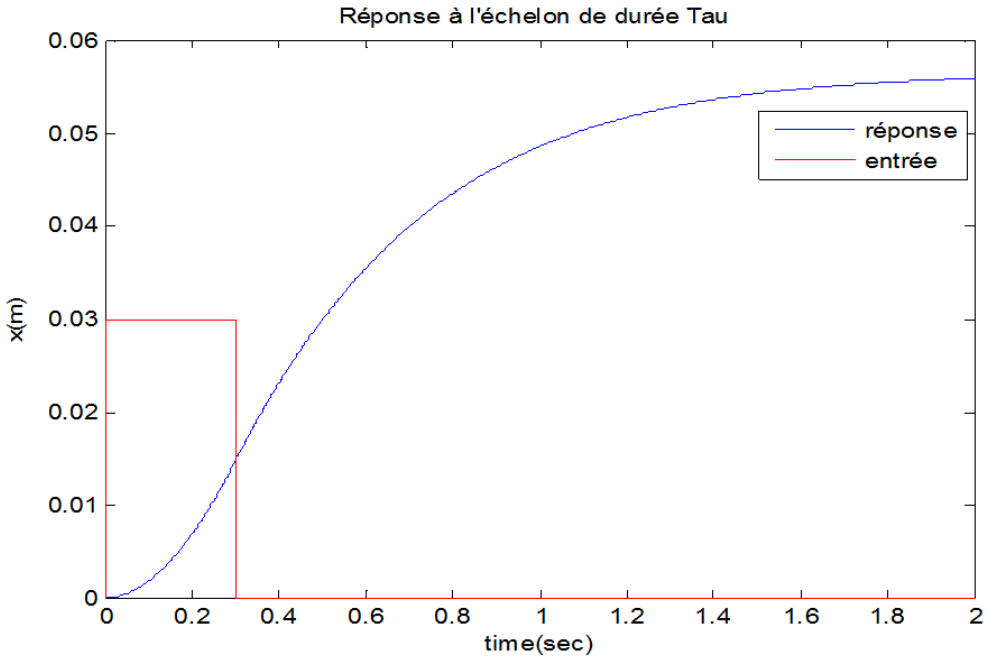


FIGURE 4.5 – Réponse à un échelon de durée $\tau = 30$ ms

4.6 Conclusion

- Les réponses du robot à quatre câbles, aux signaux de type impulsions montrent que ses comportements sont également de type « plastique ». Donc il est possible de généraliser cette hypothèse pour des robots parallèles à câbles dont les structures soient plus complexes.
- La commande en boucle ouverte du robot nous permettra de :
  - Valider le modèle dynamique trouvé et mettre en évidence ses limites.
  - Justifier la nécessité de synthétiser une commande en boucle fermée.

Chapitre 5

Simulation avec MATLAB et MSC-ADAMS

5.1 Introduction

Dans cette section on a développé un modèle dynamique virtuel sous logiciel MSC-ADAMS pour la simulation des comportements réels du robot, cette simulation nous permanant aussi la validation des modèles cinématique et dynamique de robot et aussi la validation de l'algorithme du Simplexe pour le calcul des couples optimums en temps real.

5.2 Simulation avec MSC-ADAMS

Pour pouvoir valider les résultats des couples obtenus par Matlab, nous utilisons un logiciel de simulation dynamique, en l'occurrence MSC-ADAMS dont l'interface graphique est présenté sur la Figure (5.1).

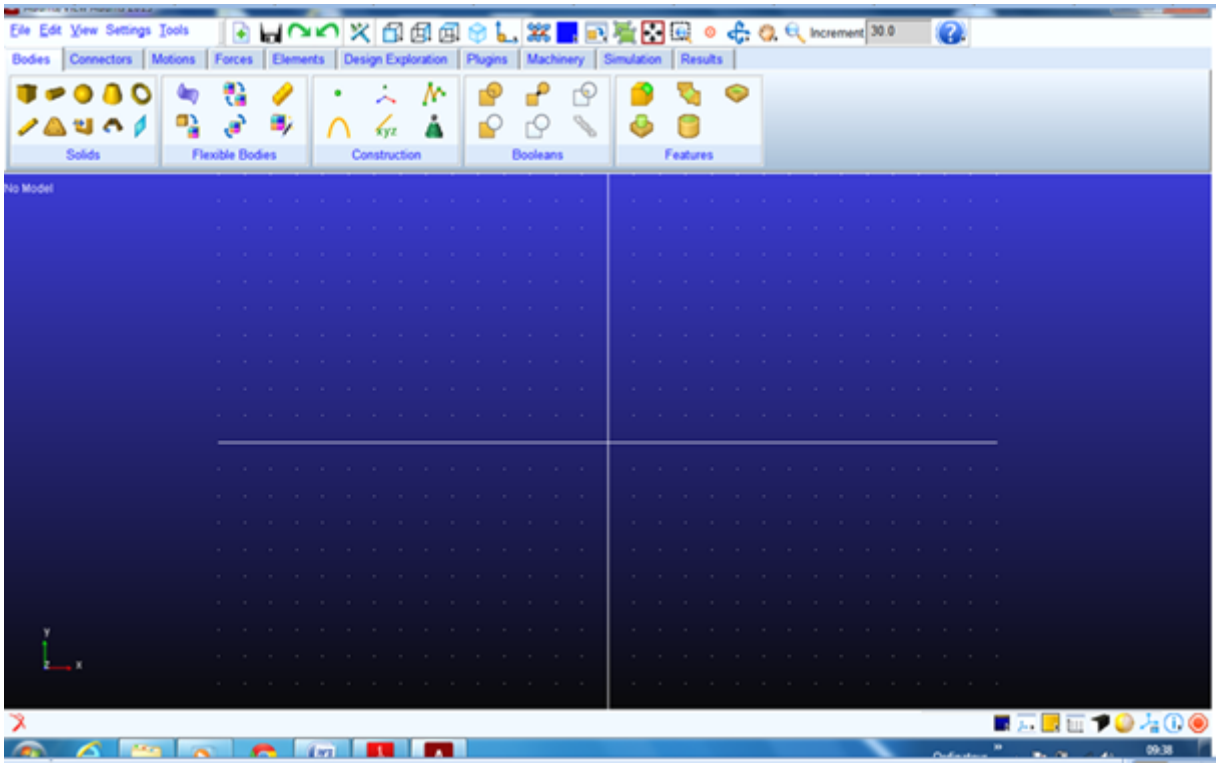


FIGURE 5.1 – Interface graphique MSC-ADAMS

Pour faire notre simulation MATLAB-MSCADAMS nous allons créer un modèle dynamique virtuel. Tout au long de notre projet on a admis l'hypothèse de câbles inextensible, de ce fait chaque câbles est modélisées par deux forces de même direction et de sens opposés, comme le montre la figure (5.2). Où les tensions sont représentées en rouge entre l'effecteur et les points fixes (les sommets d'espace de travail carré). On aurait pu commander l'effecteur par un système câbles/poulies mais cette bibliothèque n'est pas disponible dans la version étudiant d'ADAMS (Student version), pour cette raison on a choisi de commander l'effecteur par les tensions obtenues dans la partie simulation sous MATLAB.

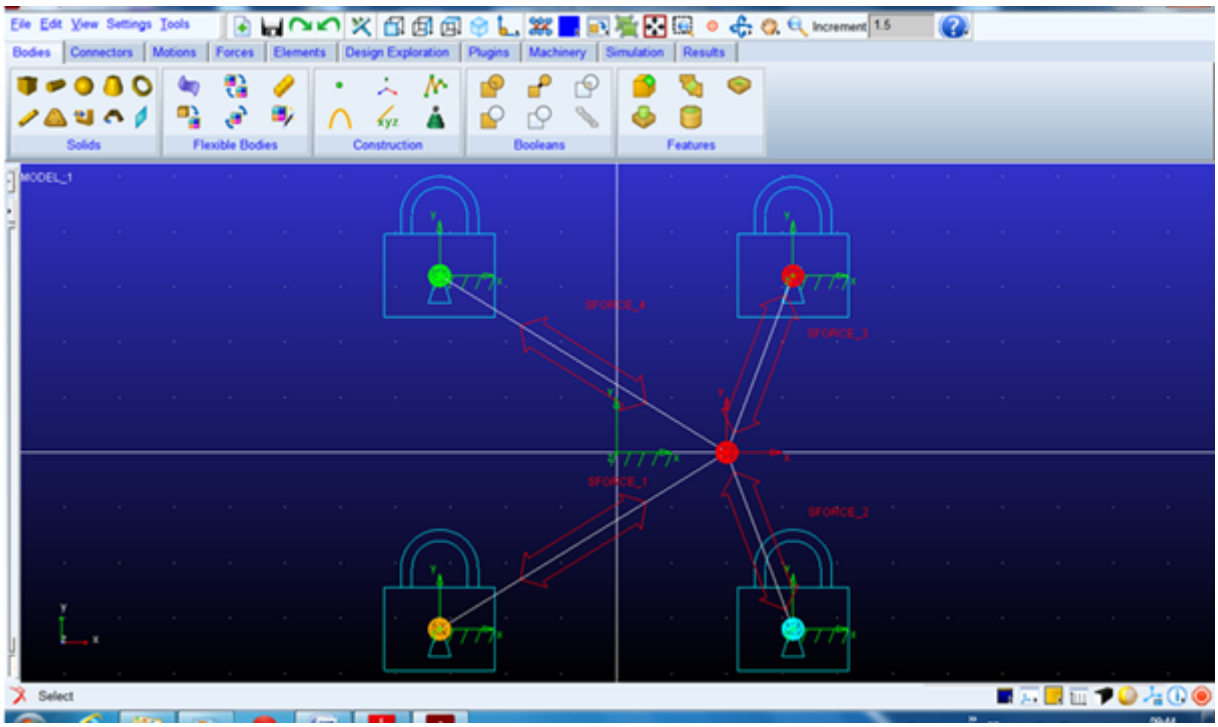


FIGURE 5.2 – Modèle virtuel du robot à 4 câbles sous MSC-ADAMS

5.3 Procédure de simulation Matlab-ADAMS

Après calcul des couples par Matlab en utilisant les données adéquates au problème, on crée un modèle virtuel sous ADAMS en respectant toutes les données nécessaire : les dimensions de l'espace de travail, la masse, la position initiale de l'effecteur... Ensuite, on importe les couples de Matlab en utilisant Simulink, ce dernier permet de faire l'interaction entre les deux logiciels ADAMS et Matlab,le modèle Simulink est représenté sur la figure (5.3) où les tensions T_i sont les outputs de Matlab et les inputs de ADAMS, et les outputs de ADAMS dans ce cas sont les positions x, y ainsi que les erreurs de position e(x), e(y).

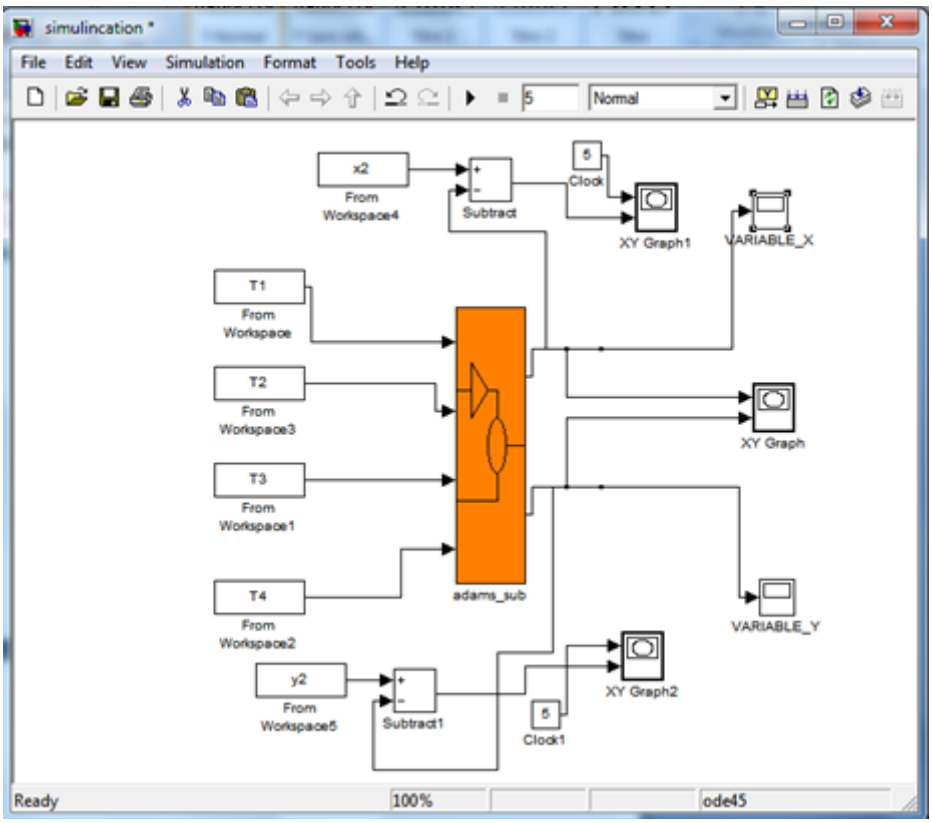


FIGURE 5.3 – Simulink pour ADAMS

5.4 Trajectoire rectiligne

On utilise la forme polynomiale de ϕ pour avoir des vitesses et accélérations nulles au début et à la fin de la course et aussi éviter les discontinuités des vitesses et des accélérations de l'effecteur, les durées du mouvement de l'effecteur sont de 5 sec et 20 sec.

Pour la simulation on a choisi :

$$\phi(t) = 9.4248\left(\frac{t}{k}\right)^5 - 23.5619\left(\frac{t}{k}\right)^4 + 15.7080\left(\frac{t}{k}\right)^3$$

Où :

- k : représente la durée d'exécution de la trajectoire ;
- La masse m = 1 kg ;
- Le couple minimum imposé : 0,05 N.m ;
- Le rayon des poulies : r = 0,05 m ;
- La base du carré : $L_b = 0.32$ m, $J_i = 0.0008$, $C_i = 0.01$.

La trajectoire est définie par :

$$Y(t) = -X(t) + 0.1$$

$$X(t) = \frac{0.1cos(\phi)}{cos(\phi) - sin(\phi)}$$

$$Y(t) = \frac{0.1sin(\phi)}{cos(\phi) - sin(\phi)}$$

L'angle ϕ varie de $[0, \frac{\pi}{2}]$, cette forme polynomiale est calculée pour satisfaire les conditions

aux limites de position, vitesse et accélération de l'effecteur.

Les conditions aux limites sont les suivantes :

$X(0) = 0.1$ m ; $Y(0) = 0$ m ;

$X(w) = 0$ m ; $Y(w) = 0.1$ m ;

$\dot{X}(0) = 0$ m/s ; $\dot{Y}(0) = 0$ m/s ;

$\dot{X}(w) = 0$ m/s ; $\dot{Y}(w) = 0$ m/s ;

$\ddot{X}(0) = 0$ m/s^2 ; $\ddot{Y}(0) = 0$ m/s^2 ;

$\ddot{X}(w) = 0$ m/s^2 ; $\ddot{Y}(w) = 0$ m/s^2 ;

Avec w = 5 sec et 20 sec.

Les résultats de simulation de la trajectoire rectiligne pour une durée de : 1 sec et 5 sec sont représentés sur les figures (5.4),(5.5),(5.6) et (5.7).

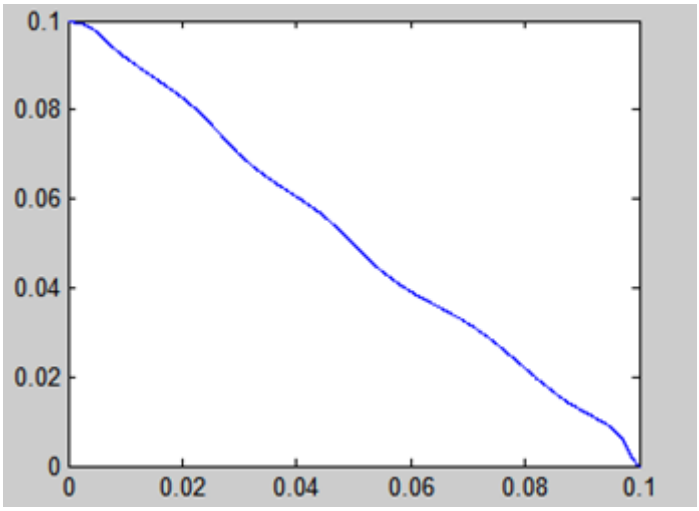


FIGURE 5.4 – Trajectoire rectiligne simulée par ADAMS pour 5 sec

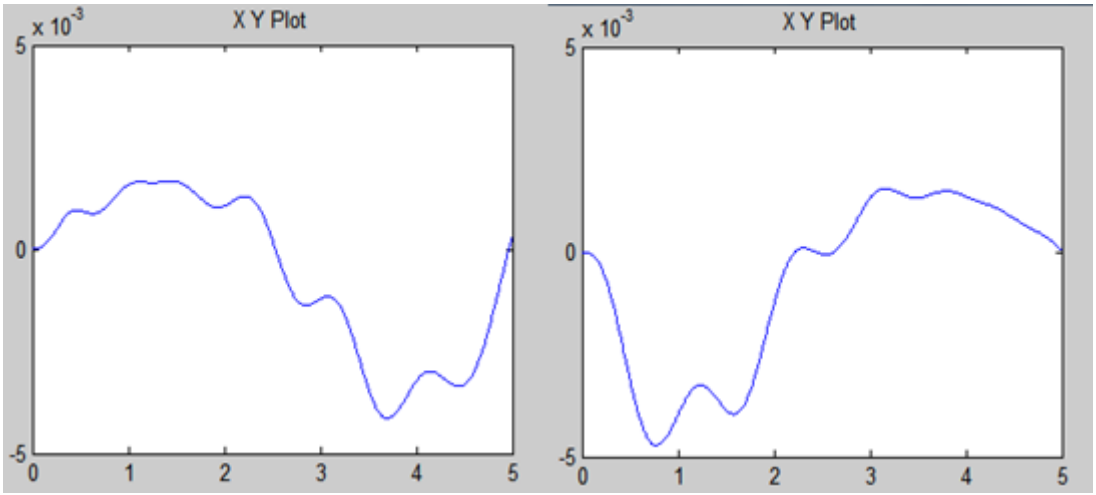


FIGURE 5.5 – Graphes des erreurs de positions e(x), e(y) pour 5 sec

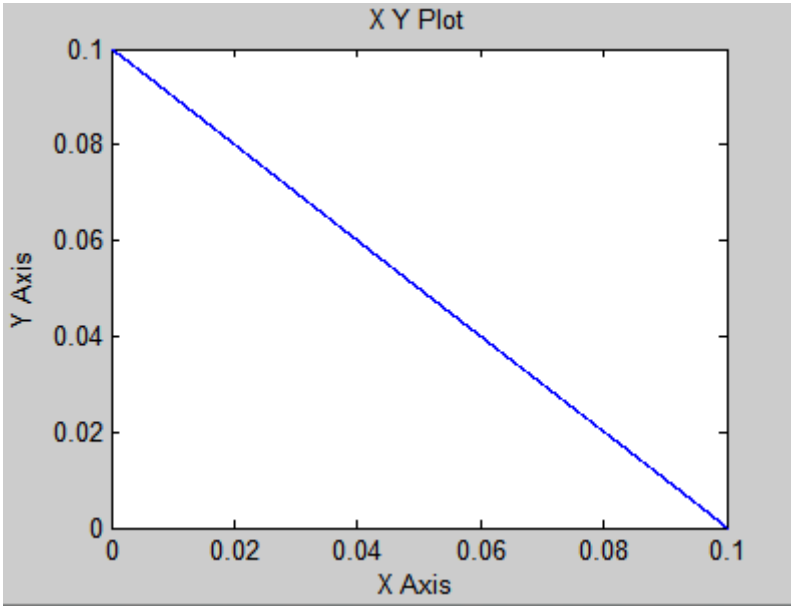


FIGURE 5.6 – Trajectoire rectiligne simulée par ADAMS pour 20 sec

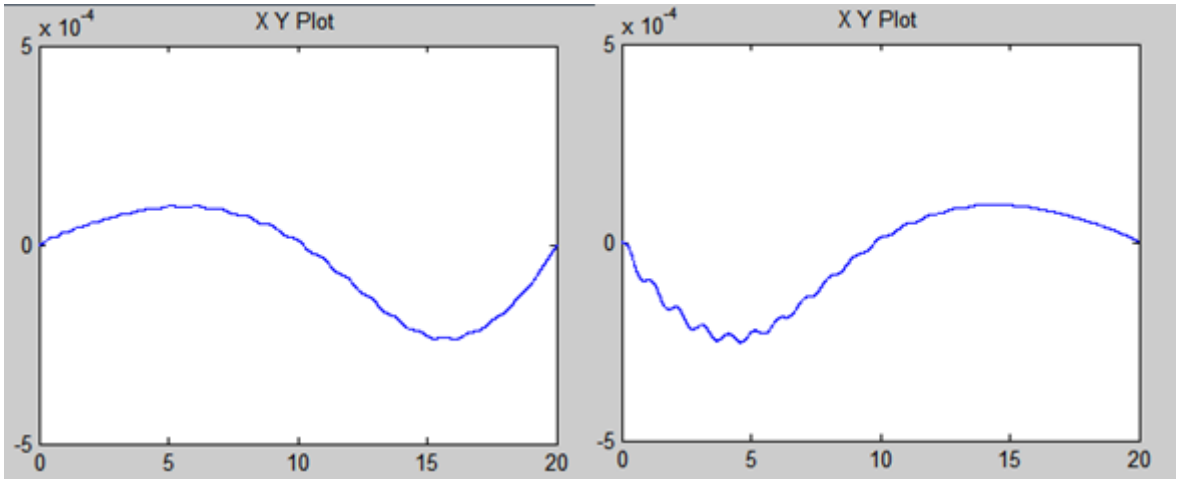


FIGURE 5.7 – Graphes des erreurs de positions e(x), e(y) pour 20 sec

5.5 Trajectoire circulaire

Pour la simulation on utilise une forme polynomiale en gardant les mêmes données que celles vu sous Matlab :

$$\phi(t) = 37.699\left(\frac{t}{k}\right)^5 - 94.248\left(\frac{t}{k}\right)^4 + 62.832\left(\frac{t}{k}\right)^3$$

$X(t) = Rcos(\phi)$

$Y(t) = Rsin(\phi)$

Où :

R=0.2165 m ;

On utilise la forme polynomiale de ϕ pour avoir des vitesses et accélérations nulles au début et à la fin de la course et aussi éviter les discontinuités des vitesses et des accélérations de l'effecteur, les durées du mouvement de l'effecteur sont de 1 sec, 5 sec et 60 sec.

Le logiciel MSC-ADAMS composé à Simulink nous donne les graphs de position de l'effecteur ainsi que les graphes leurs erreurs. L'angle ϕ varie de $[0, 2\pi]$, les conditions aux limites sont les suivantes :

$X(0) = 0.2165$ m ; $Y(0) = 0$ m ;
$X(w) = 0.2165$ m ; $Y(w) = 0.1$ m ;
$\dot{X}(0) = 0$ m/s ; $\dot{Y}(0) = 0$ m/s ;
$\dot{X}(w) = 0$ m/s ; $\dot{Y}(w) = 0$ m/s ;
$\ddot{X}(0) = 0$ m/s^2 ; $\ddot{Y}(0) = 0$ m/s^2 ;
$\ddot{X}(w) = 0$ m/s^2 ; $\ddot{Y}(w) = 0$ m/s^2 ;

Avec w = 1 sec, 5 sec et 60 sec.

Les résultats de simulation de la trajectoire circulaire pour une durée de : 1 sec, 5 sec et 60 sec sont montrés sur les figures ci-dessous :

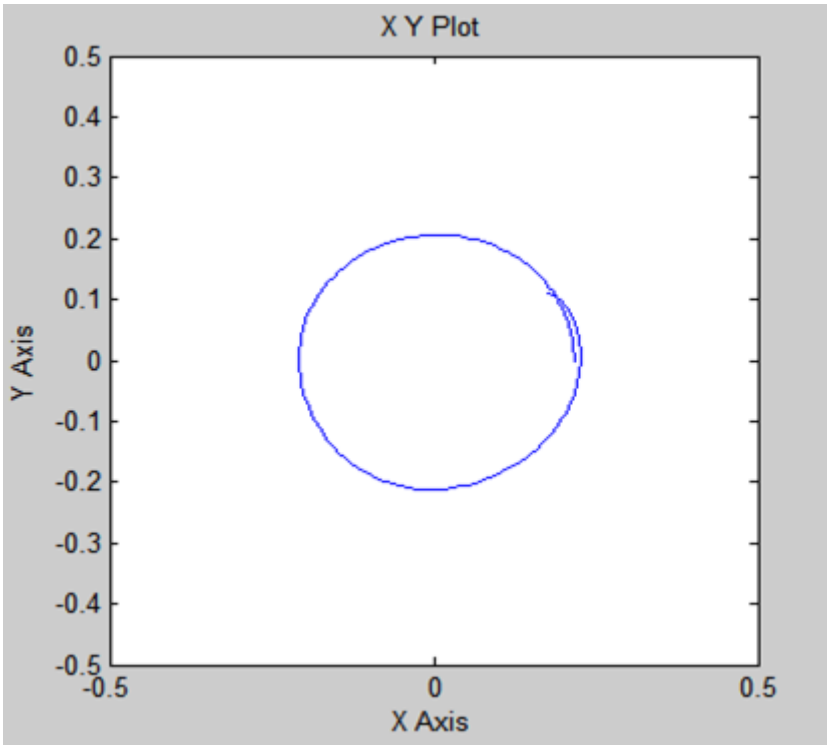


FIGURE 5.8 – Trajectoire circulaire simulée par ADAMS pour 1 sec

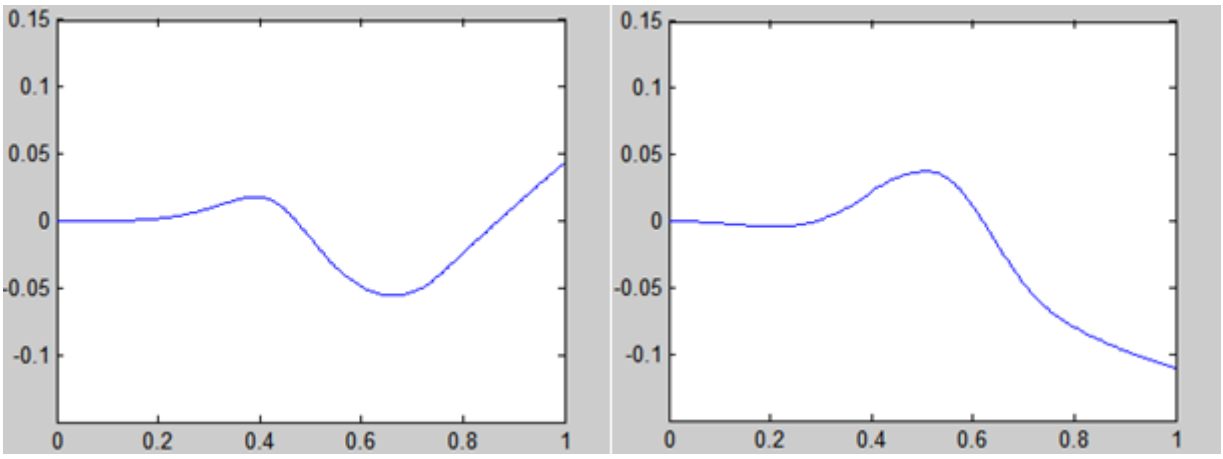


FIGURE 5.9 – Graphs des erreurs de positions e(x), e(y) pour 1 sec

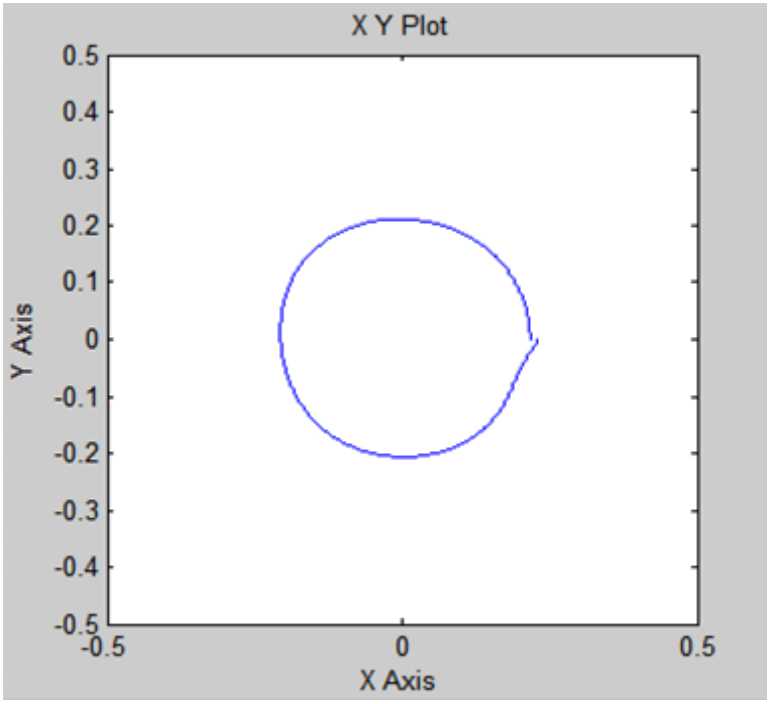


FIGURE 5.10 – Trajectoire circulaire simulée par ADAMS pour 1 sec

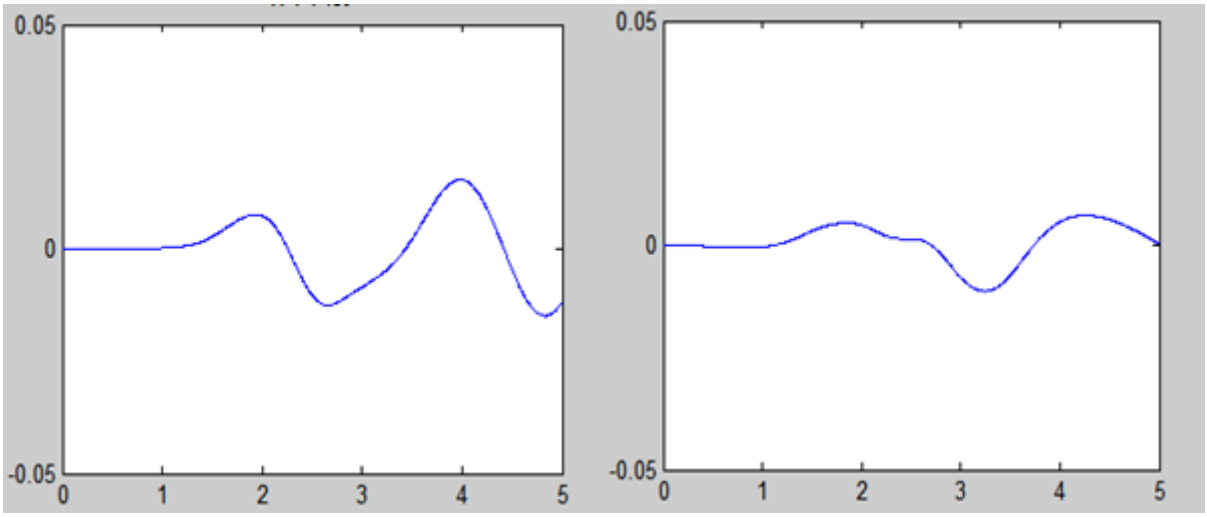


FIGURE 5.11 – Graphes des erreurs de positions e(x), e(y) pour 5 sec

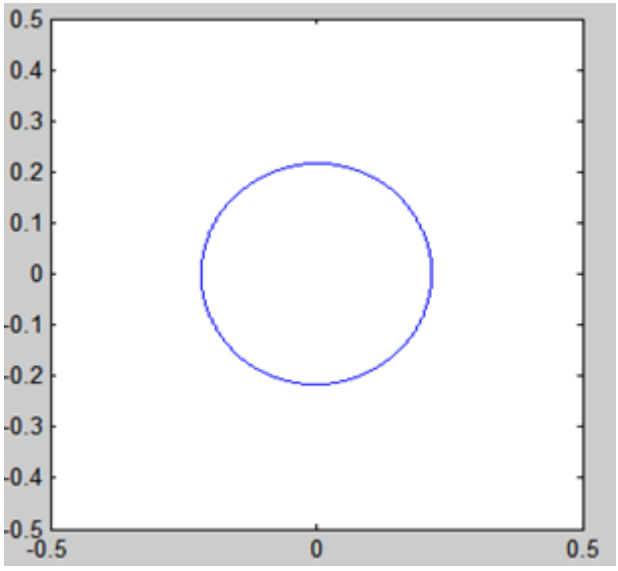


FIGURE 5.12 – Trajectoire circulaire simulée par ADAMS pour 60 sec

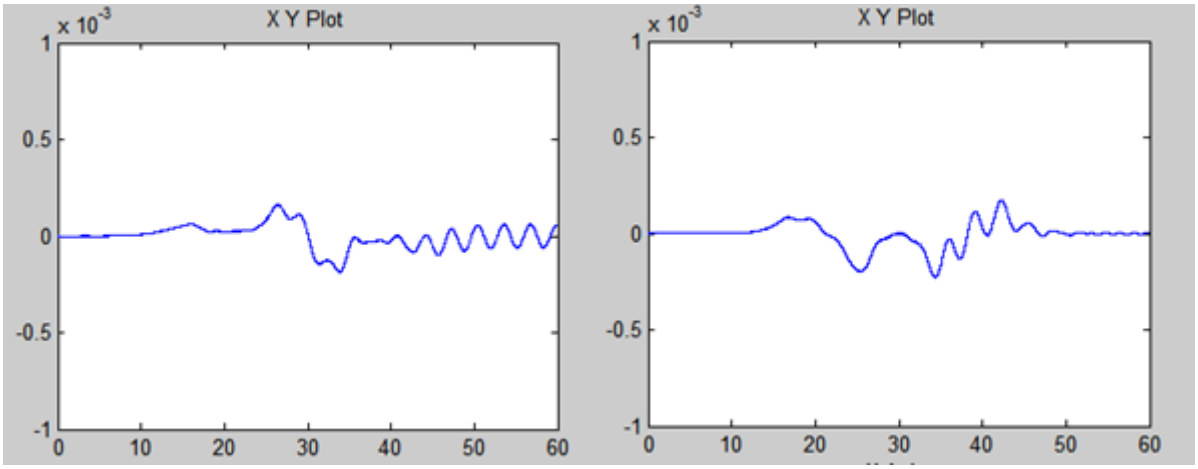


FIGURE 5.13 – Graphes des erreurs de positions e(x), e(y) pour 60 sec

On remarque bien dans la figure (5.8) et figure (5.10) que les trajectoires réalisées ne sont pas vraiment circulaire, dans les figures (5.9), figures (5.11) les erreurs de positions sont importantes, et elles sont dû principalement aux grandes accélérations de l'organe terminal qui influe sur notre méthode de calcul. En allongeant la durée d'exécution à 60 sec, comme le montre les figures (5.12) et (5.13) on obtient de plus petites erreurs.

5.6 Conclusion

Dans ce chapitre nous avons vu le comportement dynamique du robot pour des trajectoires spécifiques, et aussi présenté les résultats par des graphes en utilisant Matlab/MSC-ADAMS pour des trajectoires circulaire et rectiligne, on a aussi constaté que le temps d'exécution d'une trajectoire à une grande influence sur les erreurs.

La simulation dynamique en boucle ouverte du robot nous permettrons de :

- Valider encore une fois le modèle dynamique trouvé et mettre en évidence ses limites ;
- Justifier la nécessité d'une commande en boucle fermée.

Chapitre 6

Commande en boucle fermée

6.1 Introduction

Le contrôle des robots à câbles est encore un thème en cours de recherche. Pour cette raison, il existe très peu de travaux dans la littérature qui concernent la commande de ces robots. Toutefois, les quelques travaux dans ce domaine semblent en majorité utiliser la technique PID. Dans notre cas, nous allons explorer une technique de commande qui est le mode glissant.

6.2 Approche de commande par mode glissant

6.2.1 Introduction au mode glissant

La théorie des systèmes à structure variable (VSS) avec le mode glissant a été étudiée en détail pendant les trente dernières années. Elle est basée sur le concept de changement de la structure du contrôleur afin d'obtenir la réponse désirée. Il y a plusieurs avantages de la commande par mode glissant : la haute précision, la bonne stabilité, la simplicité, l'invariance, la robustesse ... . Ceci lui permet d'être particulièrement adaptée pour le système ayant un modèle imprécis [22]. Aussi, elle est peu sensible à certaines variations de paramètre et aux perturbations.

Par conséquent, l'approche de VSS a été largement appliquée à la conception de beaucoup des systèmes de contrôle pratique, tel que l'asservissement des systèmes, les robots manipulateurs, et les systèmes de contrôle de vol .... etc. Pour ces raisons, la commande en mode glissant va être explorée pour la commande de notre système.

6.2.2 Principe de base de la commande par mode glissant

Dans la commande des système a structure variable par mode de glissement, la trajectoire d'état est amenée vers une surface puis à l'aide de la loi de commutation, elle est obligée de

rester au voisinage de cette surface . Cette dernière est appelé surface de glissement et le mouvement qui se produit le long de celle-ci est appelé mouvement de glissement [11].

La trajectoire dans le plan de phase est constituée de trois parties distinctes, [12] :

1. Le mode de convergence « MC » : c'est le mode durant lequel l'état du système se déplace à partir de n'importe quel point initial dans le plan de phase, et tend vers la surface de commutation s(x)=0. Ce mode est caractérisé par la loi de commande et le critère de convergence.
2. Le mode de glissement « MG » : c'est le mode durant lequel la variable d'état a atteint la surface de glissement et tend vers l'origine du plan de phase. La dynamique de ce mode est caractérisée par le choix de la surface de glissement.
3. Le mode du régime permanent : ce mode est ajouté pour l'étude de la réponse du système autour de son point d'équilibre (origine du plan de phase), il est caractérisé par la qualité et les performances de la commande [22].

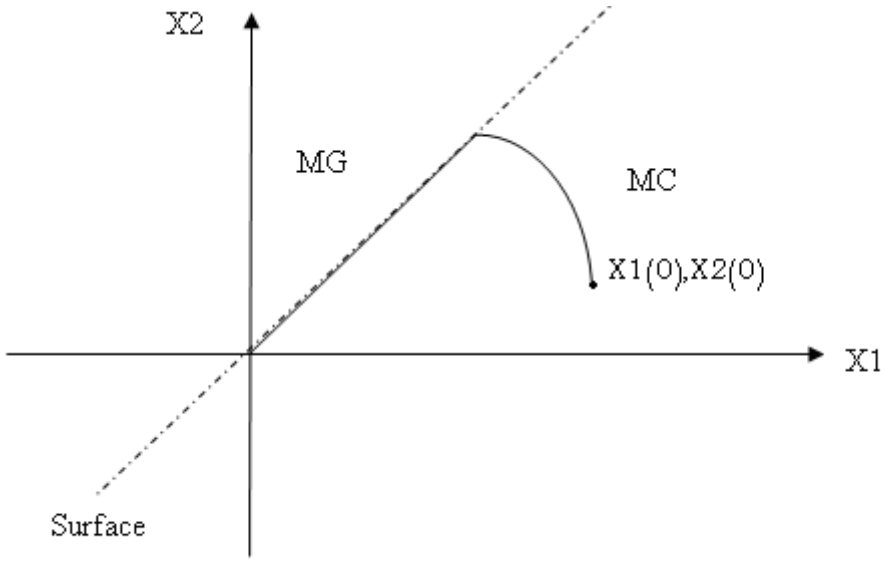


FIGURE 6.1 – Différents modes pour la trajectoire dans le plan de phase [12]

Le fonctionnement d'un système à structure variable est caractérisé par les caractéristiques suivantes :

- Puisque l'état d'équilibre du système est l'origine du plan de phase, le comportement du système en mode glissant est le comportement du système en mode transitoire.
- Pendant le mode glissant, les dynamiques du système sont uniquement déterminées par les paramètres décrivant la droite de commutation.

Il est à noter que ce mode de glissement est souvent qualifié idéal du fait qu'il requiert pour exister, une fréquence de commutation infiniment grande.En effet, tout système de commande comprend des imperfections telles que retards, hystérésis, qui imposent une fréquence de commutation finie. La trajectoire d'état oscille alors dans un voisinage de la surface de glissement, phénomène appelé chattering ou broutement.

6.2.3 Structure de la commande par mode glissant

La structure de ce type de contrôleur comporte deux parties : une partie continue représente la dynamique du système durant le mode glissant et une autre discontinue représente la dynamique du système durant le mode de convergence. Cette dernière est importante dans la commande non linéaire car elle a pour rôle d'éliminer les effets d'imprécision et des perturbations sur le modèle [12].

La conception de la commande peut être effectuée en trois étapes principales très dépendantes l'une de l'autre [12] :

1. le choix de la surface ;
2. l'établissement des conditions d'existence ;
3. et la détermination de la loi de commande.

6.3 La commande en mode glissant du robot à 4 câbles

6.3.1 établissement de la loi de commande en mode glissant

Dans le cas du robot à 4 câbles, on doit établir deux surfaces de glissement l'une selon x et l'autre selon y.

Conformément aux expressions obtenues dans la section (4.3), la représentation d'état de notre robot résulte en un système d'équations non linéaires couplées :

$$\dot{X}(t) = f(X) + g(X) * U(t) \tag{6.1}$$

La surface de glissement selon x peut s'exprimer sous la forme :

$$s_{dx} = C_{1dx} * (x_{1d}(t) - x_{1ref}) + C_{2dx} * x_{2d}(t) \tag{6.2}$$

De même, la surface de glissement selon y peut s'exprimer sous la forme :

$$s_{dy} = C_{1dy} * (x_{3d}(t) - x_{2ref}) + C_{2dy} * x_{4d}(t) \tag{6.3}$$

Ou :

C_{1dx}, C_{2dx}, C_{1dy} et C_{2dy} : sont des paramètres déterminés par la simulation ;

x_{1ref} : C'est la consigne selon x ;

x_{2ref} : C'est la consigne selon y.

On dérive l'équation (6.2), on obtient :

$$\dot{s}_{dx} = C_{1dx} * \dot{x}_{1d}(t) + C_{2dx} * \dot{x}_{2d}(t) \tag{6.4}$$

Comme on est déjà déterminé la représentation d'état du système à 4 câbles dans la section (4.3), donc :

$$\dot{x}_{1d}(t) = \dot{x}_{2d}(t)$$

Et :

$$\dot{x}_{2d}(t) = \frac{\alpha_1 * x_{2d}(t) + \alpha_2 * x_{4d}(t) + M_{22} * u_1(t) - M_{12} * u_2(t)}{\beta}$$

Alors l'équation (6.4) devient :

$$\dot{s}_{dx} = C_{1dx} * \dot{x}_{2d}(t) + C_{2dx} * \frac{\alpha_1 * x_{2d}(t) + \alpha_2 * x_{4d}(t) + \psi_x * u_{dx}(t)}{\beta} \tag{6.5}$$

Ou :

$\alpha_1 = M_{12} * N_{21} - M_{22} * N_{11}$

$\alpha_2 = M_{12} * N_{22} - M_{22} * N_{12}$

$\psi_x = M_{22} * M_{12}$

$\beta = M_{22} * M_{11} - M_{21} * M_{12}$

$u_{dx}(t) = u_1(t) - u_2(t)$

Pour déterminer la loi de commande on a travaillé avec une nouvelle méthode de synthèse qu'est l'approche de la loi d'arrivée de [22]. Sa fonction générale s'écrit :

$$\dot{s}_{dx} = -K_{dx} * s_{dx}(t) - Q_{dx} * sign(s_{dx}) \tag{6.6}$$

$$\dot{s}_{dx} = -K_{dx} * [C_{1dx} * (x_{1d}(t) - x_{1ref}) + C_{2dx} * x_{2d}(t)] - Q_{dx} * sign(s_{dx}) \tag{6.7}$$

On compare (6.5) avec (6.7) et on obtient la loi de commande :

$$u_{dx}(t) = -K_{1dx} * x_{2d}(t) - \sigma_x x_{4d}(t) - K_{2dx} * (x_{1d}(t) - x_{1ref}) - Q_x * sign(s_{dx}) \tag{6.8}$$

Ou :

- $K_{1dx} = \frac{1}{\psi_x}\left[\alpha_1 + \frac{\beta * C_{12dx}}{C_{22dx}} + K_{dx} * \beta\right]$
- $K_{2dx} = \frac{\beta * K_{dx} * C_{12dx}}{\psi * C_{22dx}}$
- $\sigma_x = \frac{\alpha_1}{\beta}$
- $Q_x = \frac{\beta * Q_{dx}}{\psi_y * C_{2dx}}$

On dérive l'équation (6.3) on obtient :

$$\dot{s}_{dy} = C_{1dy} * \dot{x}_{3d}(t) + C_{2dy} * \dot{x}_{4d}(t) \tag{6.9}$$

avec

$\dot{x}_{3d}(t) = x_{4d}(t)$

Et :

$$\dot{x}_{4d}(t) = -\frac{\rho_1 * x_{22d}(t) + \rho_2 * x_{4d}(t) + M_{21} * u_1(t) - M_{11} * u_2(t)}{\beta}$$

Alors l'équation (6.9) devient :

$$\dot{s}_{dy} = C_{1dy} * x_{4d}(t) + C_{2dy} * \frac{\rho_1 * x_{2d}(t) + \rho_2 * x_{4d}(t) + \psi_y * u_{dy}(t)}{(-\beta)} \tag{6.10}$$

Ou :

$\rho_1 = M_{22} * N_{21} - M_{21} * N_{11}$

$\rho_2 = M_{11} * N_{22} - M_{21} * N_{12}$

$\psi_y = M_{21} * M_{11}$

$\beta = M_{22} * M_{11} - M_{21} * M_{12}$

$u_{dy}(t) = u_1(t) - u_2(t)$

On peut s'écrit :

$$\dot{s}_{dy} = -K_{dy} * s_{dy}(t) - Q_{dy} * sign(s_{dy}) \tag{6.11}$$

$$\dot{s}_{dy} = -K_{2dy} * [C_{1dy} * (x_{3d}(t) - x_{2ref}) + C_{2dy} * x_{4d}(t)] - Q_{dy} * sign(s_{dy}) \tag{6.12}$$

On compare (6.11) avec (6.10) et on obtient la loi de commande :

$$u_{dy}(t) = -K_{1dy} * x_{4d}(t) - \sigma_y x_{2d}(t) - K_{2dy} * (x_{3d}(t) - x_{2ref}) - Q_y * sign(s_{dy}) \tag{6.13}$$

Ou :

- $K_{1dy} = \frac{1}{\psi_y}\left[\rho_2 + \frac{(-\beta) * C_{1dy}}{C_{2dy}} + K_{dy} * (-\beta)\right]$
- $K_{2dy} = -\frac{\beta * K_{dy} * C_{1dy}}{\psi * C_{2dy}}$
- $\sigma_y = \frac{\rho_1}{\beta}$
- $Q_y = -\frac{\beta * Q_{dy}}{\psi_y * C_{2dy}}$

Lorsque on applique les contraintes sur la commande, on remarque des broutements interviennent à cause de la fonction *sign*, dans le but de réduire ces broutements, on remplace la fonction *sign* dans les équations (6.8) et (6.13) par :

$\frac{s_d}{\mid s_d \mid + \nu}$

Ou :

$s_d = s_{dx}$ ou s_{dy}

Et :

$\nu = \nu_x$ ou ν_y

6.3.2 Simulation de la commande avec contraintes en mode glissant pour le robot à 4 câbles

Conformément aux équations gouvernant le mode glissant appliqué à notre robot à 4 câbles, on a développé un programme de simulation sous Matlab & Simulink. On notera que les commandes sont les forces appliquées sur l'effecteur final selon les directions x et y. Ces forces doivent être converties sous forme de tension par la relation de Moore Penrose. De par leurs natures, pour notre application, il faut que ces tensions soient toujours positives. De même, ces dernières doivent être limitées entre t_min et t_max, pour que les câbles restent toujours tendus. De plus, l'organe terminal ne doit pas dépasser l'espace de travail. La simulation du fonctionnement de notre système nécessite l'intégration du système d'équations sous les contraintes précitées. Elle a été effectuée par la méthode numérique de Runge Kutta.

La figure de principe de commande par mode glissant donne par :

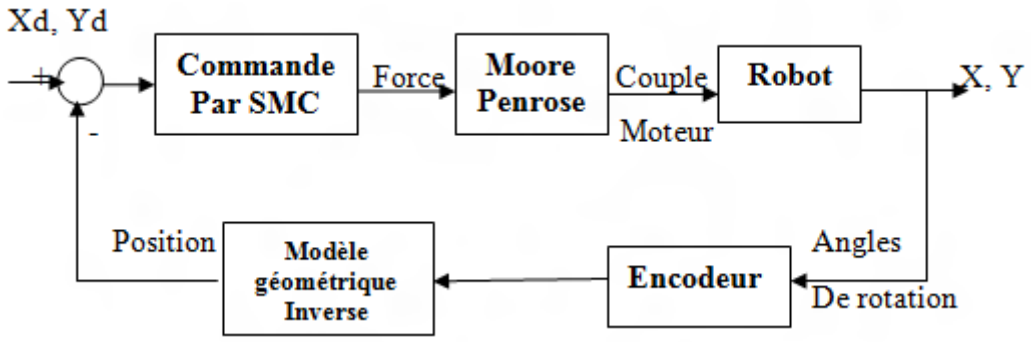


FIGURE 6.2 – Représentation de la loi de commande de SMC

En réalité, il faut ajouter deux blocs à la chaîne de retour pour obtenir la position. Le premier bloc est l'encodeur, son entrée est la sortie du système qui est l'angle de rotation des poulies, et sa sortie est la valeur des longueurs de câbles, et le deuxième est pour la modélisation géométrique directe qui nous donne la valeur de la position.

Le schéma de simulation correspondant aux expressions de la commande en boucle fermée par mode glissant est donné par la figure suivant :

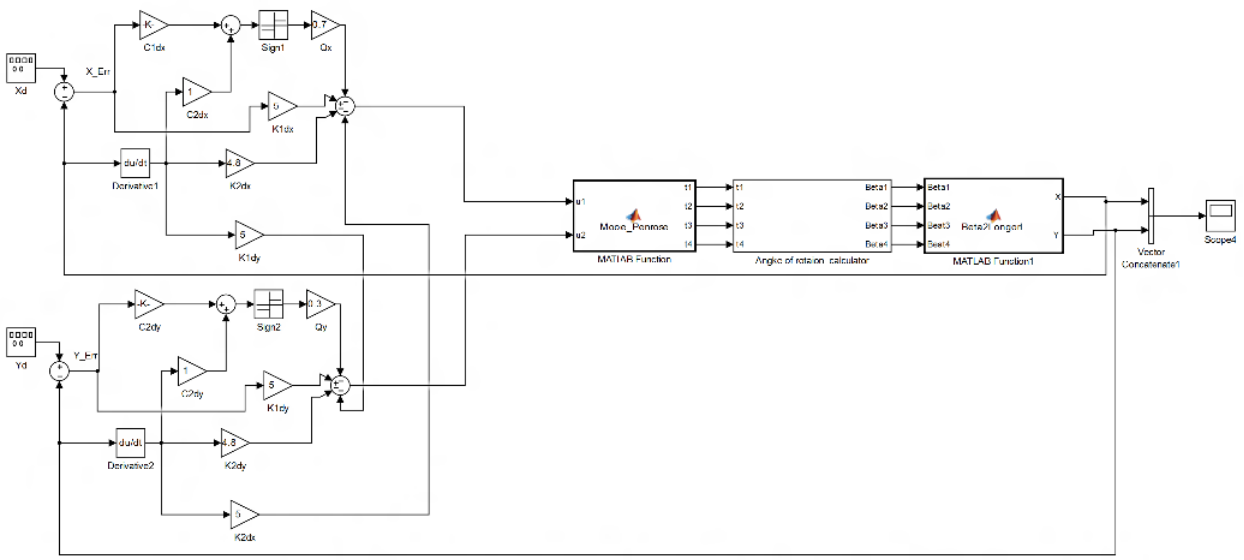


FIGURE 6.3 – Le schéma d'implémentation correspondant aux expressions de la commande en boucle fermée par SMC selon x et y

Le schéma précédent de la commande comprend quatre étages. Le premier permet l'introduction de la consigne x_{ref} et y_{ref}. Le deuxième représente le contrôleur SMC qui détermine les forces F_x et F_y. Ces dernières sont définies comme des entrées du troisième bloc qui permet de déterminer les couples moteurs τ_1, τ_2, τ_3 et τ_4 par la méthode de Moore Penrose. Après la détermination des couples optimums, ces derniers servent à commander effectivement les positions et les vitesses de l'effecteur final. A cet effet, on a joué sur les paramètres de commande afin d'obtenir de bonnes performances de l'organe terminal (stabilité, rapidité, précision).

Enfin, les paramètres choisis sont :

$C_{1dy} = C_{1dx} = 12.91$

$C_{2dy} = C_{2dx} = 1$

$Q_x = 0.7$

$Q_y = 0.3$

6.3.2.1 Exemple de simulation du robot à 4 câbles

On considère notre système à l'état de repos à la position :

$x_{1d}(0) = 0$ cm et $x_{3d}(0) = 0$ cm.

On souhaite transférer l'effecteur terminal à la position finale :

$x_{1d}(0) = 1$ cm et $x_{3d}(0) = 3$ cm.

Les résultats des positions selon (x et y) obtenus par la simulation sont présentés aux Figures (6.4) et (6.5) respectivement :

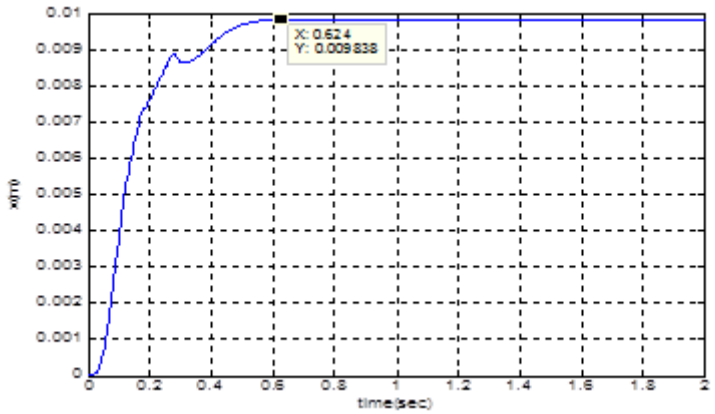


FIGURE 6.4 – Profil de la positon (x) par mode glissant

En examinant le profil de la position x(t) représenté la figure ci-dessus, on peut constater que la commande par mode glissant en boucle fermée envoie effectivement l'organe terminal au point désiré mais avec des petites perturbations causées par la commutation de la commande. Le temps de réponse est déterminé par le choix des paramètres de commande et les paramètres physiques du robot, dans notre système, le temps de réponse est de l'ordre de 0.6 sec.

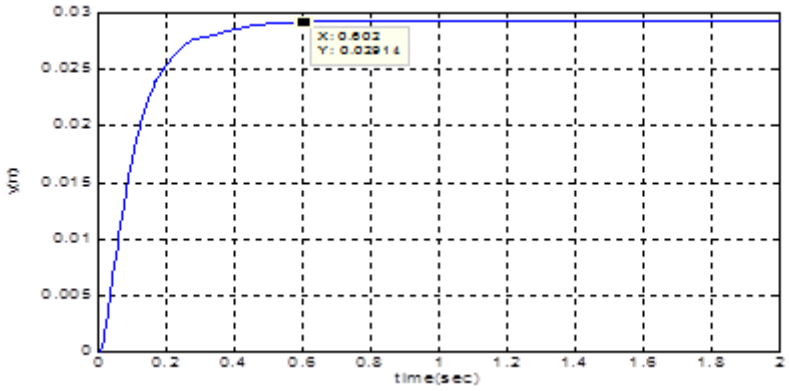


FIGURE 6.5 – Le profil de la positon (y) par le mode glissant

De même, en examinant le profil de la position y(t) représenté par la figure ci-dessous, on peut constater que la commande par mode glissant en boucle fermée envoie effectivement l'organe terminal au point désiré. Pour le temps de réponse est déterminé par la même minière que celui précédant.

Les profils des vitesses (V_x) et (V_y) par le mode glissant sont présentés aux Figures (6.6) et (6.7) respectivement :

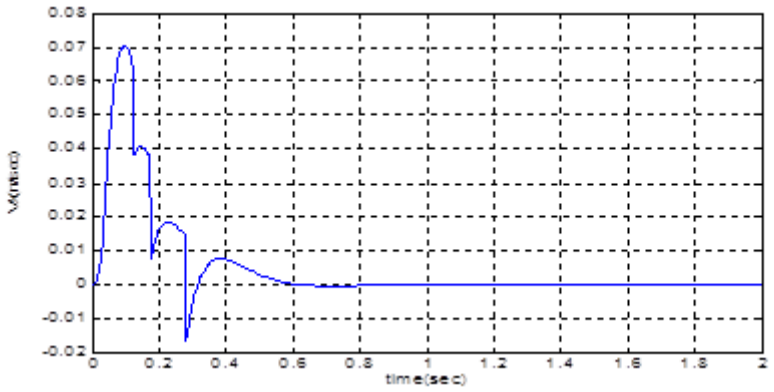


FIGURE 6.6 – Le profil de la vitesse (V_x) par le mode glissant

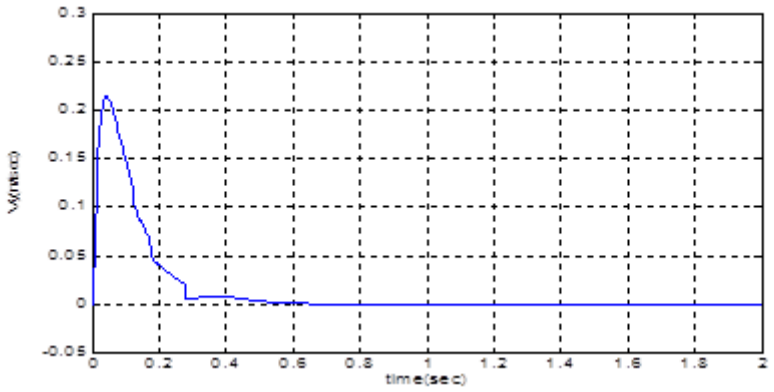


FIGURE 6.7 – Le profil de la vitesse (V_y) par le mode glissant

On constate que la vitesse a une forme impulsionnelle avec une décélération plus lente que la montée. Avec des petites déformations causées par la commutation de la commande.

Les trajectoires dans les plans de phase (x, V_x) et (y, V_y) par le mode glissant sont présentées aux figures (6.8) et (6.9) respectivement :

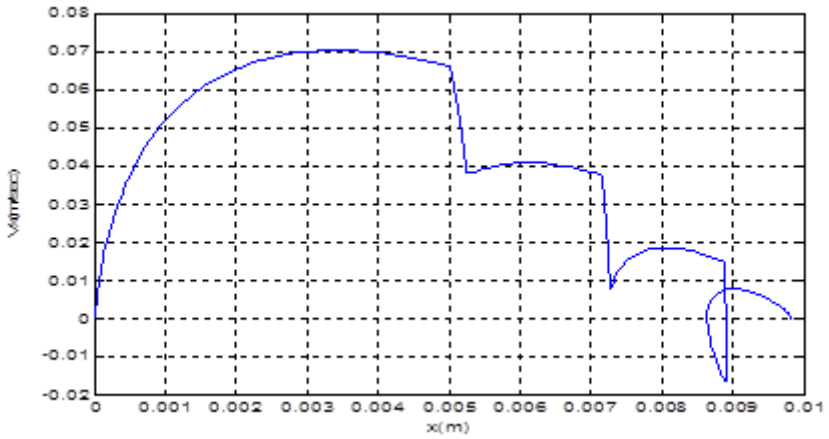


FIGURE 6.8 – Trajectoire du système représentée en plan de phase selon (x)

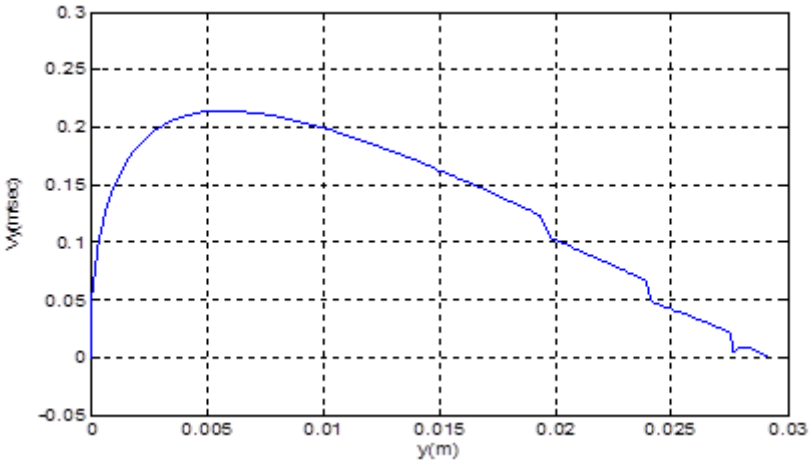


FIGURE 6.9 – Trajectoire du système représentée en plan de phase selon (y)

On remarque que la figure (6.8) montre que les broutements sur x sont plus grands que ceux sur y, car le paramètre $Q_x > Q_y$.

La commande représentée dans la figure ci-dessus permet à l'effecteur terminal de suivre la trajectoire représentée dans la Figure (6.10) :

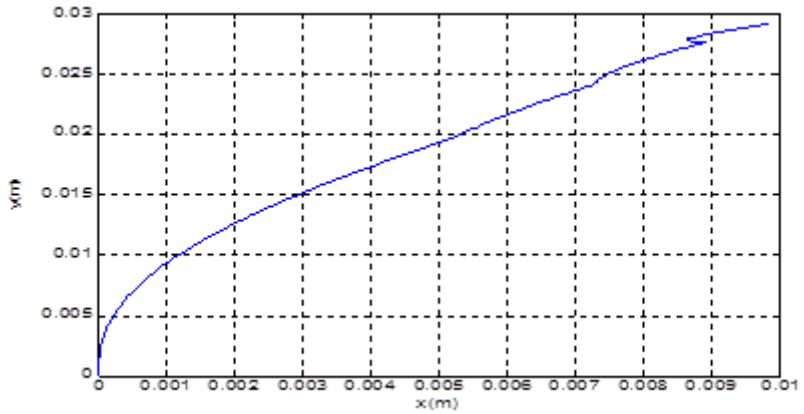


FIGURE 6.10 – La trajectoire suivie par l'effecteur terminal

6.4 Conclusion

Le mode glissant offre de bonnes performances (un très bon temps de réponse, stabilité et haute précision) pour commander les robots a câbles.

Chapitre 7

Réalisation expérimentale

7.1 Introduction

Dans cette section, on présente les démarches suivis pour la réalisation de robot parallèle à quatre câbles destiné à la réhabilitation des bras pour but d'aider les gents à des besoins spéciaux à l'écriture. Pour cela, on a exploité les notions théoriques dans les chapitres précédents (la modélisation, calcules des couples optimums, asservissement en boucle fermée ....etc.).

7.2 Partie mécanique du robot

Dans ce chapitre, nous détaillerons les étapes successives de la construction Mécanique de notre robot. Au départ d'une stratégie claire et précise, nous élaborerons les différentes parties du robot. Comment développer un robot parallèle planaire à quatre câbles ? Et comment peut-on faire pour aider les gents à des besoins spéciaux à l'écriture ? Quelle structure mécanique la plus pratique pour cette application ?

Telle était les questions qui nous préoccupaient sans cesse. Comme dans toute construction mécanique , un cahier de charges a été élaboré. Lors du développement du projet.

7.2.1 Les bâtis

Ils sont formé par quatre cornières en aluminium 550 x 30 x 4 mm soudées entre elles pour former un bâti carré, c'est la base du robot. Avec le même type de cornière fixée aussi par soudage, on ajoute des supports de 460 mm de hauteur pour former l'ossature du robot. La figure (7.1) illustre le montage utilisé.

FIGURE 7.1 – Les bâtis

7.2.2 La table

C'est une plaque en vitre d'une forme carré de 550 mm de côté. Elle constitue le terrain du dispositif sur lequel se déplace l'effecteur. Cette plaque comporte dans ses coins des trous de 8 mm de diamètre dont le centre est à une distance de 50 mm du sommet sur la diagonale comme le montre la figure (7.2). Ces orifices permettent le passage des axes des moteurs sur lesquels sont montées les poulies de 70 mm et de rainure à la circonférence de 5 mm.

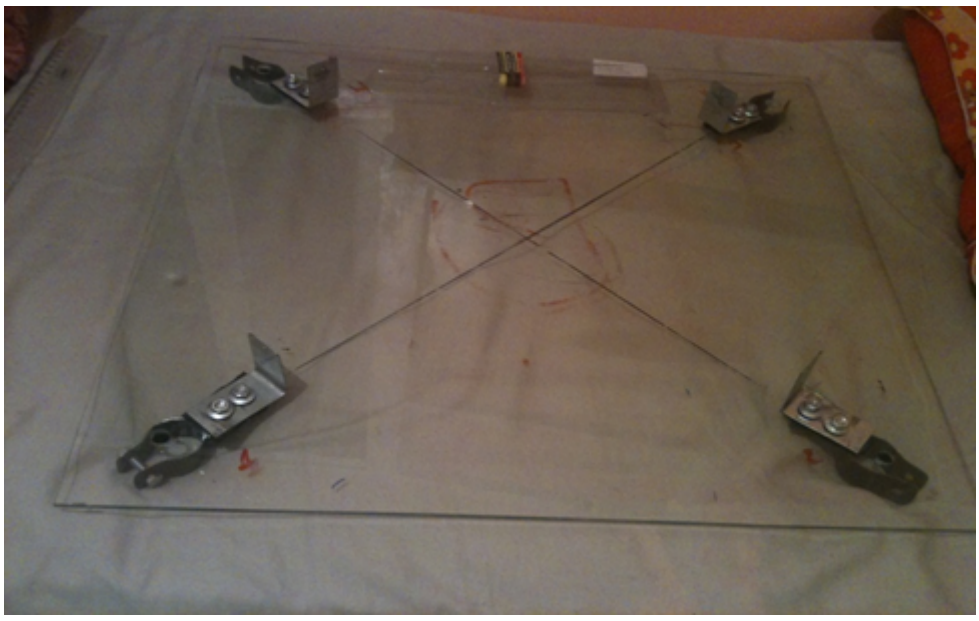


FIGURE 7.2 – La table

7.2.3 Les guides

Ce sont des plaques de 50 mm de hauteur en acier, fixés sur la table à l'aide de vis comme le montre la figure (7.3). Les guides servent à délimiter l'espace de travail du robot.

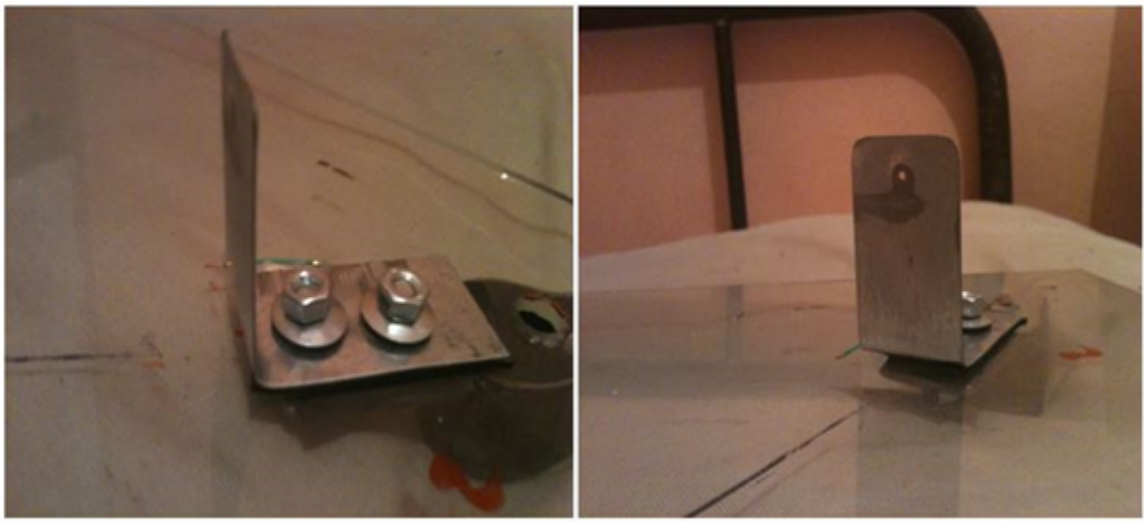


FIGURE 7.3 – Les guides

7.2.4 Les câbles

Les câbles sont des fils du pêche très résistants dont la masse est négligeable, et inextensibles, ce sont très pratiques pour notre application. La figure (7.4) montre les câbles utilisés.

FIGURE 7.4 – Les câbles

7.2.5 L'effecteur

C'est une pièce cylindrique en acier de 70 mm de diamètre et de 60 mm de hauteur, percé dans son centre un trou de 18 mm pour pouvoir mettre le stylo. Et dans sa base, des billes sont placées pour l'aider à déplacer sur l'espace de travail sans frottement comme le montre la figure (7.5).

FIGURE 7.5 – L'effecteur

7.3 Partie électrique du robot

L'électrique constitue un des trois piliers de la science que nous appelons Robotique. Elle complète parfaitement la Mécanique et l'Informatique. Elle s'intéresse aux circuits électriques, carte de commande, l'alimentation électrique du robot ... etc.

Dans le cadre de ce projet, l'électrique (électronique) est utilisée comme outil permettant le traitement d'informations. Elle nous permettra, à partir de données entrantes (inputs) via d'une interface d'utilisateur (Human/Machine Interface) de fournir des données sortantes (outputs).

7.3.1 Moteurs électriques

La commande par le modèle dynamique nous mène au choix de moteurs à courant continu de 12 V [30] avec réducteur (rapport 30 :1) pour générer des couples avec des valeurs considérables, et encodeur intégré de résolution de +/- un(1) dégrée pour mesurer le feedback en angle des quatre moteurs, et à travers le modèle géométrique inverse on peut déterminer les coordonnées de l'effecteur. Chaque moteur a une consommation de 530 mA moyenne en charge et 150 mA à vide, la vitesse de rotation : 170 t/min (216 t/min à vide)

Couple : 1,5 kg.cm. Encodeur : 360 impulsions par tour. Dimensions : ϕ28, 5 x 86,6 mm. Axe 4 mm de diamètre (voir la figure (7.6)).

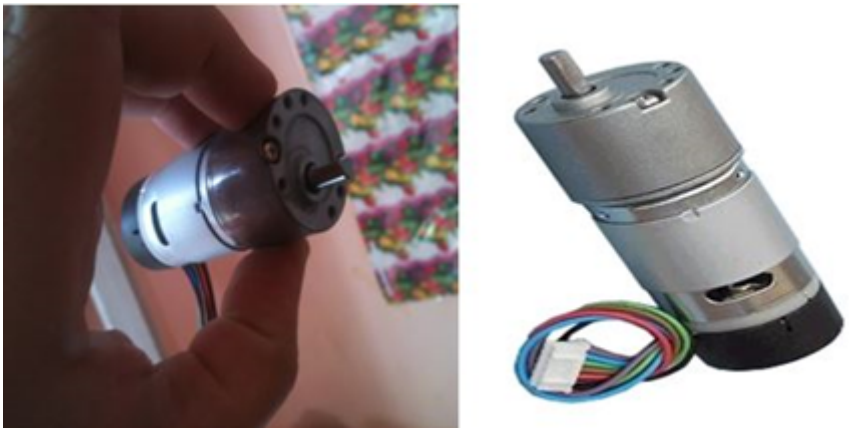


FIGURE 7.6 – Moteur à courant continu

7.3.2 Carte de commande

La carte de commande ARDUINO MEGA 2560 comme montre la figure (7.7) est un Circuit imprimé équipé de composants électroniques[25]. le circuit imprimé est en licence libre, mais certains composants ne sont pas libres de droit ; comme le microcontrôleur sur lequel on peut programmer pour analyser et produire des signaux électriques, de manière à effectuer des tâches diverses tel le contrôle des appareils domestiques - éclairage, chauffage... ; ou comme pour notre cas le pilotage d'un robot(moteurs).

C'est une plateforme basée sur une interface entrée/sortie simple, les sorties que nous utilisons envoient des signaux numériques et analogiques PWM (Pulse Width Modulation).

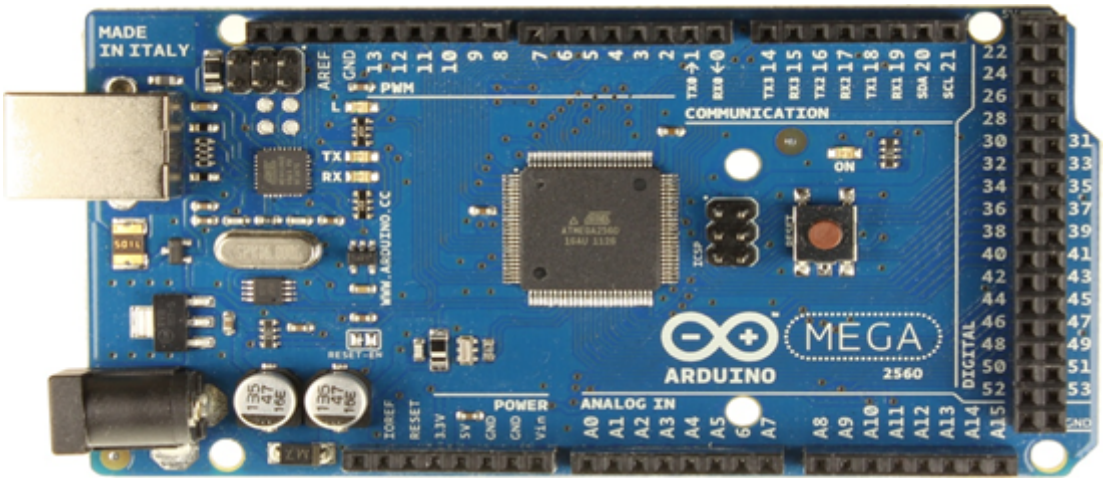


FIGURE 7.7 – La carte de commande ARDUINO MEGA 2560

7.3.3 Le circuit de commande L298N

Le circuit intégré L298N comme montre la figure (7.8) est le composant le plus utilisé lorsqu'il s'agit de piloter des moteurs à courant continu ou des moteurs pas à pas, il permet

en effet de commander n'importe quel moteur. Ce circuit n'est autre qu'un double pont de puissance possédant des sorties de mesure du courant consommé par le moteur ainsi que des entrées de validation. Il ne nécessite donc que très peu de composants externes et sa mise en œuvre est très simple.

Le circuit intégré commande deux moteurs a la fois d'où pour notre application on a besoin que deux circuits L298N pour commander les quatre moteurs (les actionneurs).

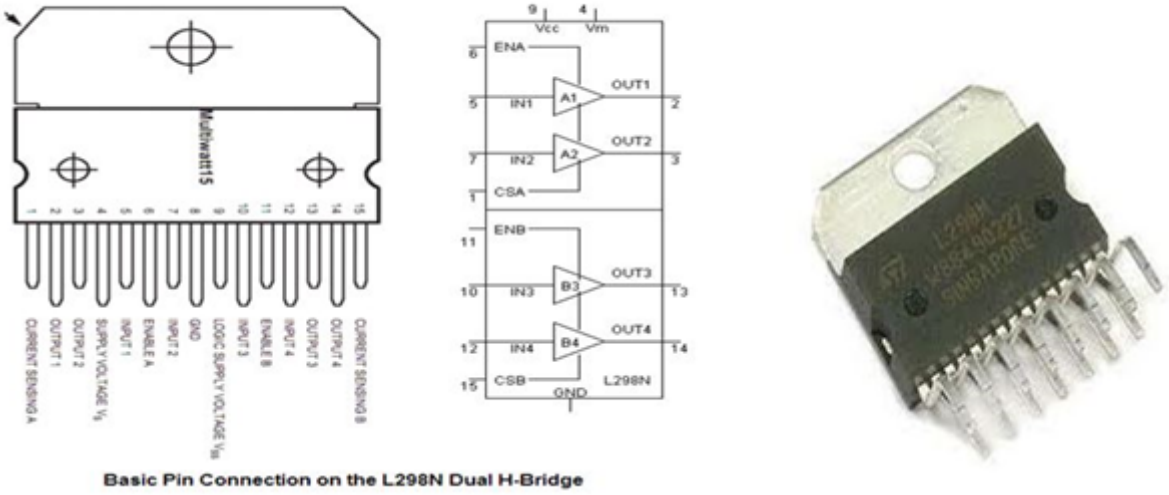


FIGURE 7.8 – Le circuit de commande L298N

7.3.4 Alimentations

Les moteurs reçoivent leur énergie à partir d'une alimentation séparée qui délivre une tension stable de 12 volts. Ils sont commandés par la carte ARDUINO. La carte elle-même reçoit son alimentation via un câble USB connecté à l'ordinateur ou via une batterie externe de 9 volts. Le dispositif est représenté sur le schéma de la figure (7.9).

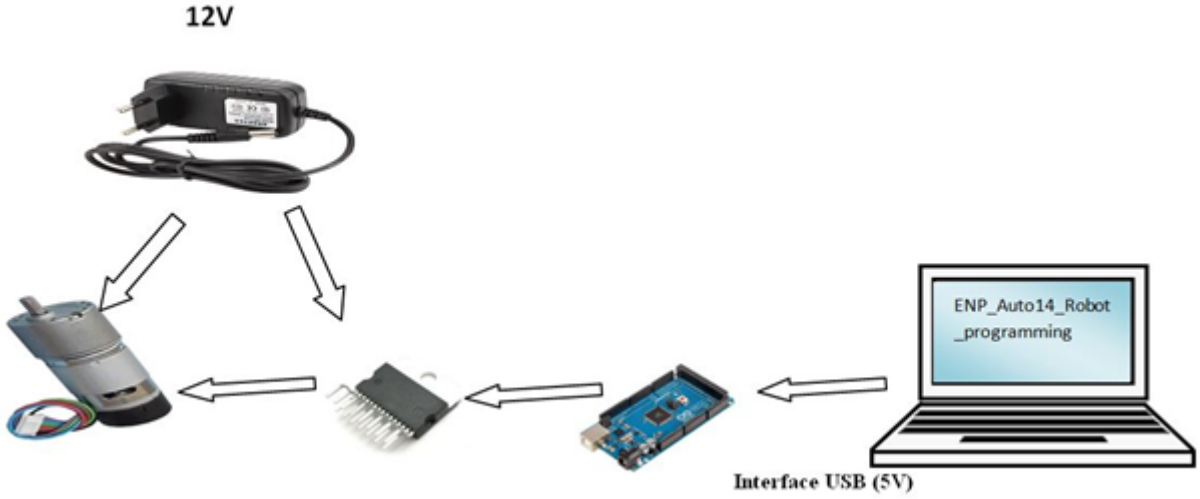


FIGURE 7.9 – Schéma descriptive de déférentes sources d'alimentation

7.3.5 Capteurs

Pour calculer le feedback en position et vitesse,on est obligé d'utiliser des moteurs à courant continus équipés par des capteurs de vitesse et de position.

Les moteurs (les actionneurs) utilisés dans notre robot ont équipes par des encodeurs à effet hall incrémental comme montre la figure (7.10).

FIGURE 7.10 – Encodeur à effet de hall incrémental [27]

Le codeur incrémental fournit deux signaux carrés en quadrature[28], comme sur la capture ci-dessous :

FIGURE 7.11 – Les signaux carrés en quadrature de l'encodeur [27]

Ces deux signaux permettent de mesurer à la fois la vitesse et le sens de rotation aussi la position.

1. **Comptage du nombre d'impulsions**

   Compter le nombre d'impulsions du codeur revient à compter le nombre de fronts montants et descendants des signaux jaune et bleu représentés sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher les deux signaux (les fils jaune et blanc sur le codeur utilisé) sur deux entrées « interruption » de la carte Arduino. Les deux autres fils (bleu et vert) seront respectivement branchés sur le 5 V et sur la masse de l'Arduino.

   Sur une carte Arduino Mega 2560 il y a six lignes d'interruption [29] (numérotées 0 jusqu'à 5), qui correspondent aux broches digitales 2 et 3, 21,20 le 19,18. L'intérêt d'une ligne d'interruption est qu'elle permet, comme son nom l'indique, d'interrompre le déroulement des calculs sur le micro-contrôleur pour effectuer un traitement spécifique, en l'occurrence de la mise à jour du compteur d'impulsions, avant de rendre la main à la boucle principale.

   La seule « difficulté » est de savoir s'il faut incrémenter ou décrémenter le compteur dans le traitement de l'interruption. Il suffit pour cela d'observer les courbes ci-dessus, obtenues alors que le moteur tourne dans le sens positif. On constate que :

   - Lorsque la voie A (en jaune) passe au niveau haut, la voie B (en bleu) est au niveau bas ;
   - Lorsque la voie A passe au niveau bas, la voie B est au niveau haut.

2. **Calcul de la vitesse et l'angle de rotation**

   La mesure de l'angle de rotation de chaque mateur se fait simplement en comptant le nombre d'impulsions, et pour la vitesse se fait en comptant le nombre d'impulsions pendant un temps fixe. Les données du problème sont les suivantes :

   - Le codeur est fixé à l'arbre moteur et non pas à l'arbre de sortie du réducteur (celui utilisé pour l'entrainement). Le rapport de réduction étant 30 :1, l'arbre moteur fait 30 tours lorsque l'arbre « principal » en fait un tour.
   - Le codeur génère 12 impulsions à chaque fois qu'il fait un tour (360/30).
   - La cadence d'échantillonnage utilisée pour l'asservissement sera de 0.01 s. Par conséquent, lorsque l'arbre principal fait un tour, le codeur génère : 30 * 12 = 360 impulsions.
     Donc pour chaque tour d'arbre moteur l'encodeur génère 360 impulsions d'où a chaque impulsion l'angle de rotation est 1 degrée.
     Pour la vitesse, si N est le nombre d'impulsions comptées en 0.01 s, la vitesse est (en rad/s, l'unité standard, sachant qu'un tour fait 2*π radians) : 2*π*N/ (0.01*1632).
     Un point très important concerne la résolution de la mesure, c'est-à-dire la plus petite valeur qu'il est possible de calculer. La formule est la suivante (en rad/s) [29] :

$2*\pi/(T_s*CPR*ratio)$, avec :

- T_s : cadence d'échantillonnage ;
- CPR : nombre d'impulsions par tour du codeur ;
- ratio : rapport de réduction du moteur.

Le but étant d'avoir la plus faible résolution possible (pour avoir une bonne précision de mesure), il faut avoir T_s et/ou CPR et/ou ratio le plus grand possible. Cependant, ratio est fixé par le besoin en couple ou en vitesse maximale alors que T_s doit être plus petit que le temps de réponse souhaité pour l'asservissement du moteur. Par conséquent, la seule réelle possibilité est de jouer sur CPR. Lors du choix d'un codeur incrémental, il est préférable de prendre celui qui permet d'obtenir le plus d'impulsions par tour.

Dans notre cas la résolution est la suivante : $2*\pi/(0.01*360) = 1.74$ rad/s ,en vitesse et +/- un(1) dégrée en angle.

Ce n'est pas exceptionnel, le seul moyen de faire mieux serait de réduire le temps de réponse de l'asservissement.

7.4 Partie informatique du robot

La programmation d'un robot est comme la programmation de tout programme : simple et complexe à la fois ! En effet, il s'agit que d'une succession de petites actions basiques ingénieusement combinées afin de créer des actions plus complexes en apparence.

Chaque petite action prise séparément reste cependant simple en elle-même. La complexité se situe quant à elle au niveau de l'optimisation du regroupement de ces actions basiques dans le but de réduire les temps de calcul,afin d'obtenir un meilleur fonctionnement.

Dans le cadre de ce projet, l'informatique permet avant toute chose la gestion du déplacement De l'effecteur dans l'espace de travail du robot à travers la programmation de modèle dynamique du robot simulé par Matlab& Simulink et validé par logiciel MCS-ADAMS. L'informatique est également utilisée comme outil permettant de traiter les données entrantes (inputs) provenant des encodeurs à effet hall et faire réagir le système en conséquence en fournissant des données sortantes (outputs).

7.4.1 Logiciel Arduino IDE (Integrated Development Environment)

Le logiciel de programmation des modules Arduino est une application Java multiplateformes (fonctionnant sur tout système d'exploitation) et un environnement de développement libre[29], servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module). En l'occurrence

Arduino IDE dont l'interface graphique est présentée sur la figure (7.12).

Le langage de programmation utilisé est le C^{+}+, compilé avec avr-g++ [30].

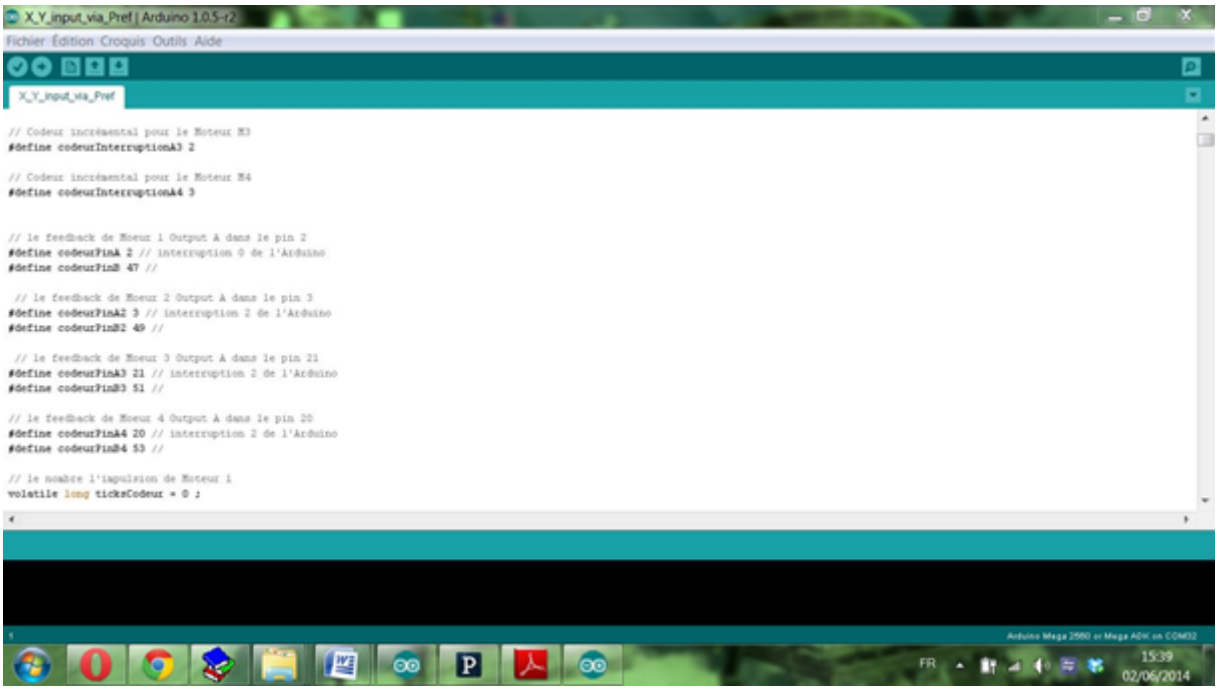


FIGURE 7.12 – Interface graphique Arduino IDE

Cet environnement de programmation nous permîmes de programmer le modèle dynamique du robot ainsi de l'implémentation d'un régulateur PID par la technique du couple calculé (voir la partie Master pour la théorie et la synthèse de régulateur).

Après la compilation de programme nous transférons vers la carte de commande ARDUINO MEGA 2560 comme le montre le schéma dans la figure (7.13).

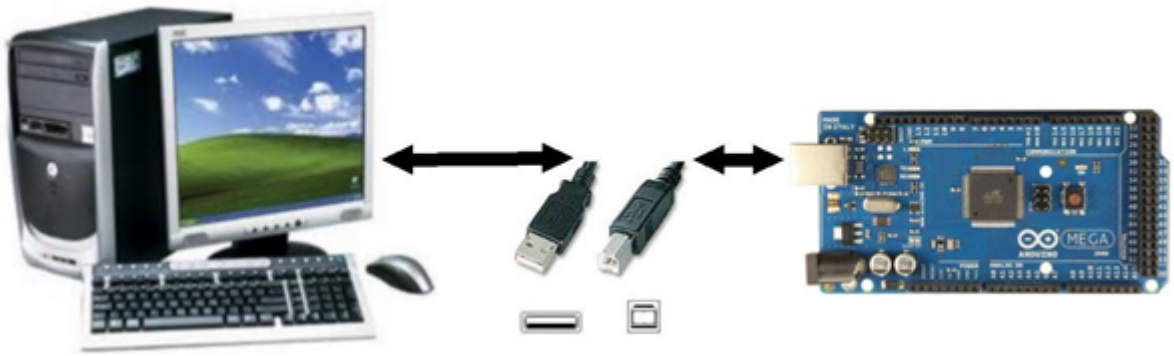


FIGURE 7.13 – Communication Série entre le PC et ARDUINO MEGA 2560 [31]

7.4.2 Logiciel Proceesing

Processing est une librairie java et un environnement de développement libre dont l'interface graphique est présentée sur la figure (7.14), créé par Benjamin Fry et Casey Reas, deux artistes américains[32]. Processing est le prolongement « Multimédia » de Design by

numbers, l'environnement de programmation graphique développé par John Maeda au Media Lab du Massachusetts Institute of Technology (MIT).

Processing est tout particulièrement adapté à la création plastique et graphique interactive. Le logiciel fonctionne sur Macintosh,Windows, Linux, BSD et Android. Il est basé sur la plate-forme Java, il permet d'ailleurs de programmer directement en langage Java.

Cet environnement de programmation nous permîmes de programmer une interface graphique (human machine interface) pour l'utilisation du robot.

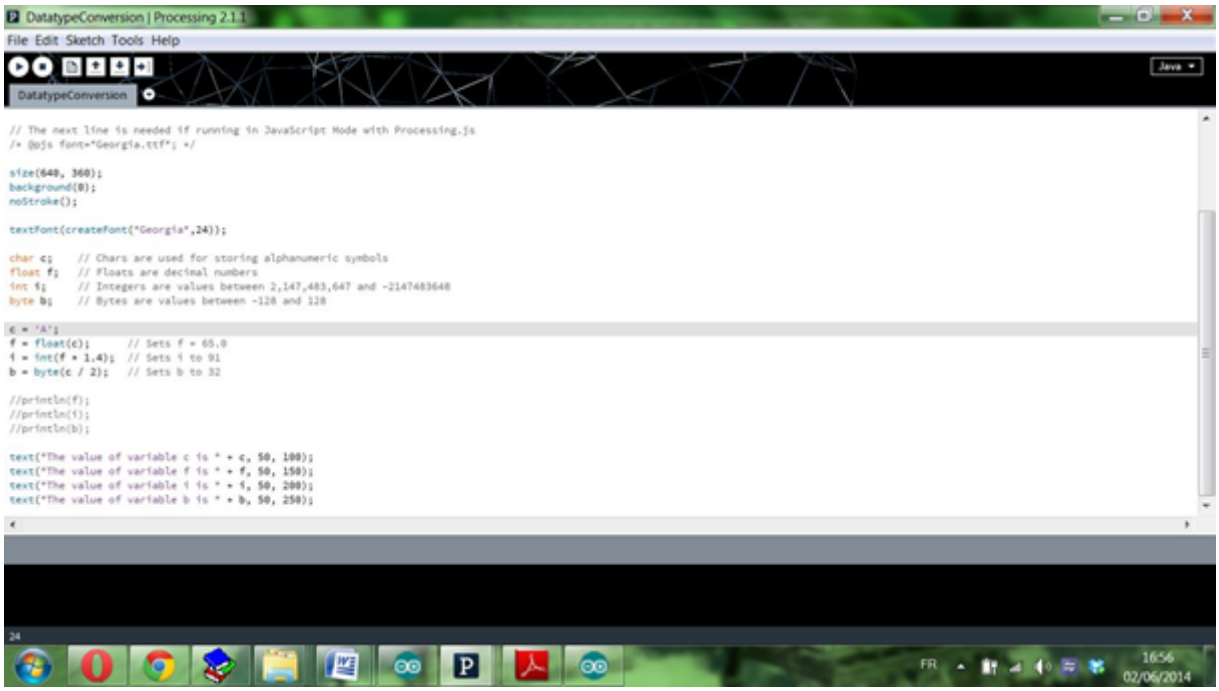


FIGURE 7.14 – Interface graphique Processing

7.5 Commande du robot

Un programme en langage ARDUINO est réalisé sur un logiciel de commande propre à cette marque. Doté d'une interface graphique, il permet la compilation et l'injection du programme par le biais d'un câble USB connecté entre l'ordinateur et la carte.

Malheureusement nous n'arrivons pas à faire un régulateur par mode glissant comme nous l'avons déjà vu au chapitre 6. Une des raisons de cet insuccès trouve son explication dans le fait que la carte de commande utilisée n'est pas aussi puissante pour supporter la complexité de programmation de cette commande.

La commande du robot se fait grâce à l'implémentation d'un régulateur PID robuste synthétisé avec la technique du couple calculé (computed torque methode).

Les contrôleurs PID par la technique du couple calculé sont bien connus et largement utilisés pour les commandes des robots par ce qu'ils sont simples, efficaces, robustes, et facilement réglés et implémentés.

Nous avons simulé le contrôleur PID avec la technique du couple calculé, en tenant compte des contraintes à satisfaire sur les tensions des câbles, à savoir que les tensions doivent toujours être positives et comprises entre T_min et T_max.

7.5.1 Synthèse d'un régulateur PID avec la technique de couple calculé

Le modèle dynamique du robot planaire à quatre câbles donné par sa forme canonique de Brunovski est le suivant :

$$M(X)\ddot{X} + N(X,\dot{X}) = F(X)\tau \tag{7.1}$$

$$F(X) = S(X)\tau$$

Tel que :

$$S(X) = \begin{pmatrix} -\cos(\theta_1) & -\cos(\theta_2) & -\cos(\theta_3) & -\cos(\theta_4) \\ -\sin(\theta_1) & -\sin(\theta_2) & -\sin(\theta_3) & -\sin(\theta_4) \end{pmatrix} \tag{7.2}$$

est la matrice de Jacobienne.

Et :

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} \tag{7.3}$$

sont les couples appliques sur chaque moteur (actionneur).

F(X) La force résultante appliquée sur l'organe terminal du robot comme montre la figure (7.15)

$$M = r * m + S(X)J\frac{\partial \beta}{\partial X}$$

Matrice d'inertie du robot

Et :

$$N(X,\dot{X}) = S(X)(J\frac{d}{dt}\frac{\partial \beta}{\partial X} + C\frac{\partial \beta}{\partial X})\dot{X}$$

Matrice des termes non linéaires

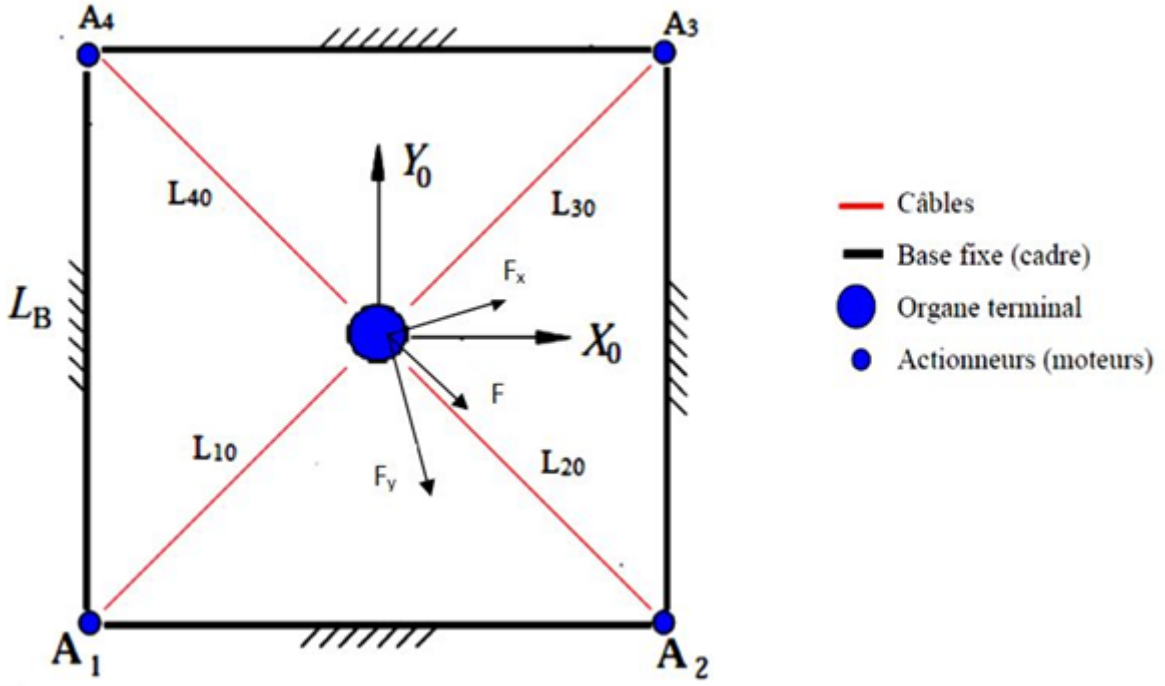


FIGURE 7.15 – Force résultante appliquée sur l'effecteur

La figure (7.16), montre le schéma de commande du robot :

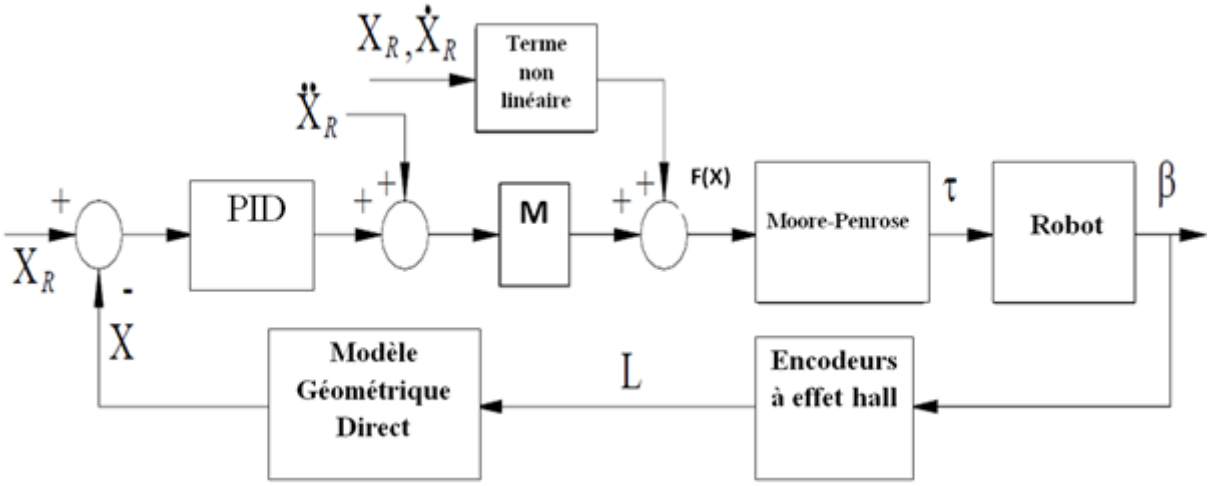


FIGURE 7.16 – Schéma de commande du robot

La loi de commande correspondante à cette structure donnée par [9] est :

$$F(x) = M(x).[\ddot{X}_d + K_v\dot{e} + K_p e + K_i\varepsilon] \tag{7.4}$$

Nous avons simulé le contrôleur PID avec la technique du couple calculé, en tenant compte des contraintes à satisfaire sur les tensions des câbles, à savoir que les tensions doivent toujours être positives et comprises entre T_min et T_max.

Pour examiner les performances de régulateur, en imposant une trajectoire sinusoïdale suivant X et Y :

X_d = A sin $(\left(\frac{2\pi}{T}\right).t)$ et Y_d=A cos $(\left(\frac{2\pi}{T}\right).t)$, avec A = 0.3 m est l'amplitude des signaux et T = 2 s est la période des signaux.

Les dimensionnements du robot sont les suivants :

- La base carré est de côté : L_b = 33.7 cm ;
- La masse : m = 0.8 kg ;
- L'inertie poulie/rotor : J_i = 0,0008 kg.m^2, i : 1,4 ;
- Le coefficient de frottement : C_i = 0,01 N.m.s ;
- Le rayon des poulies : r_i = 3.5 cm.

Les résultats de simulation sont illustrés dans les figures ci-dessous :

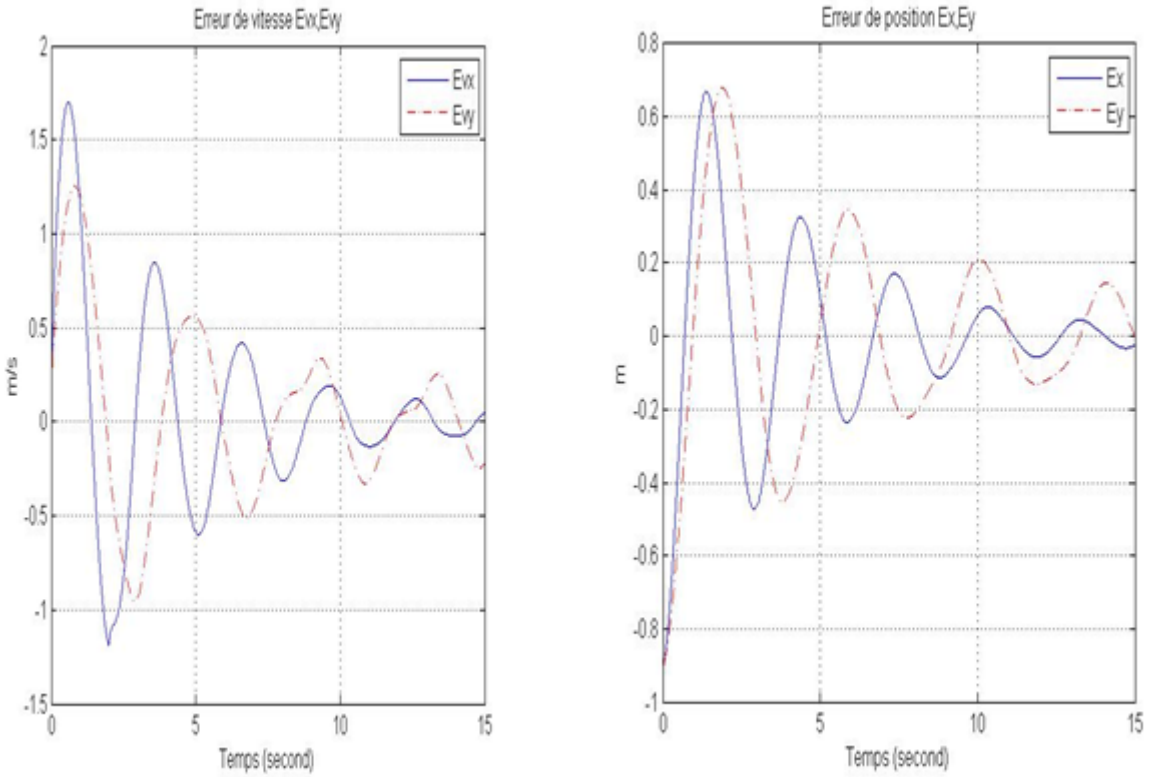


FIGURE 7.17 – L'erreur en positions et en vitesse

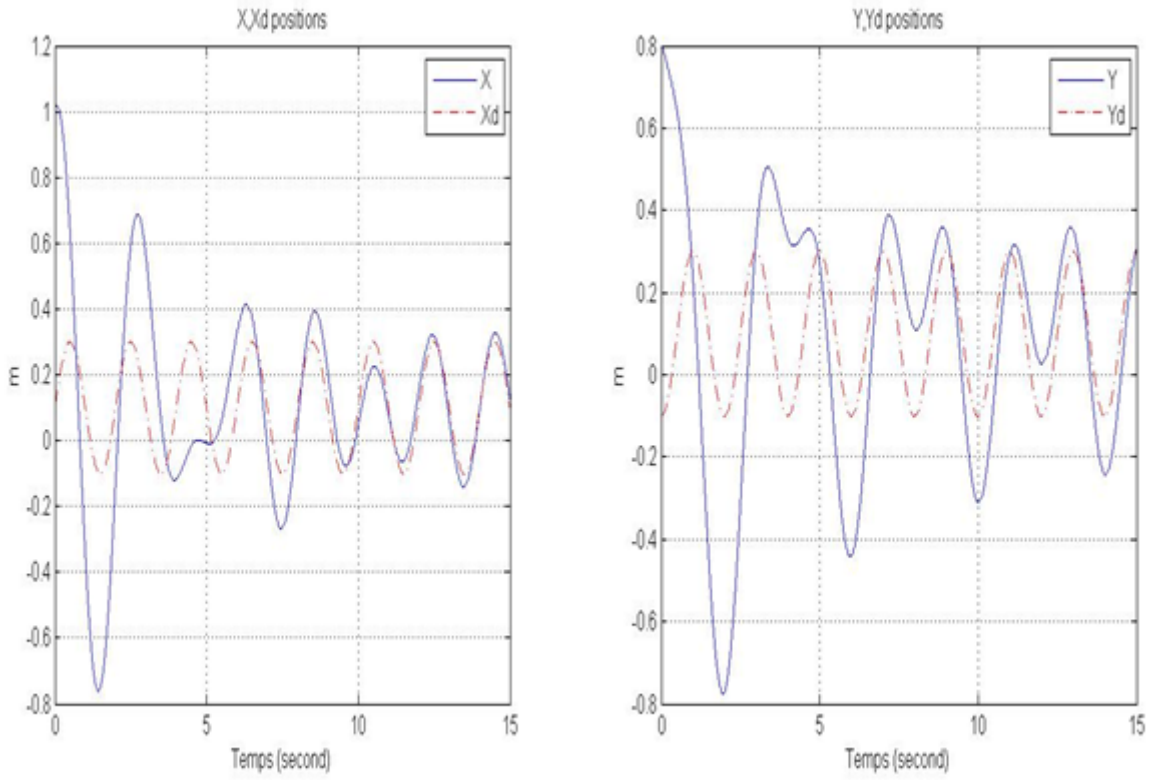


FIGURE 7.18 – Positionnement suivant les deux axes X, Y

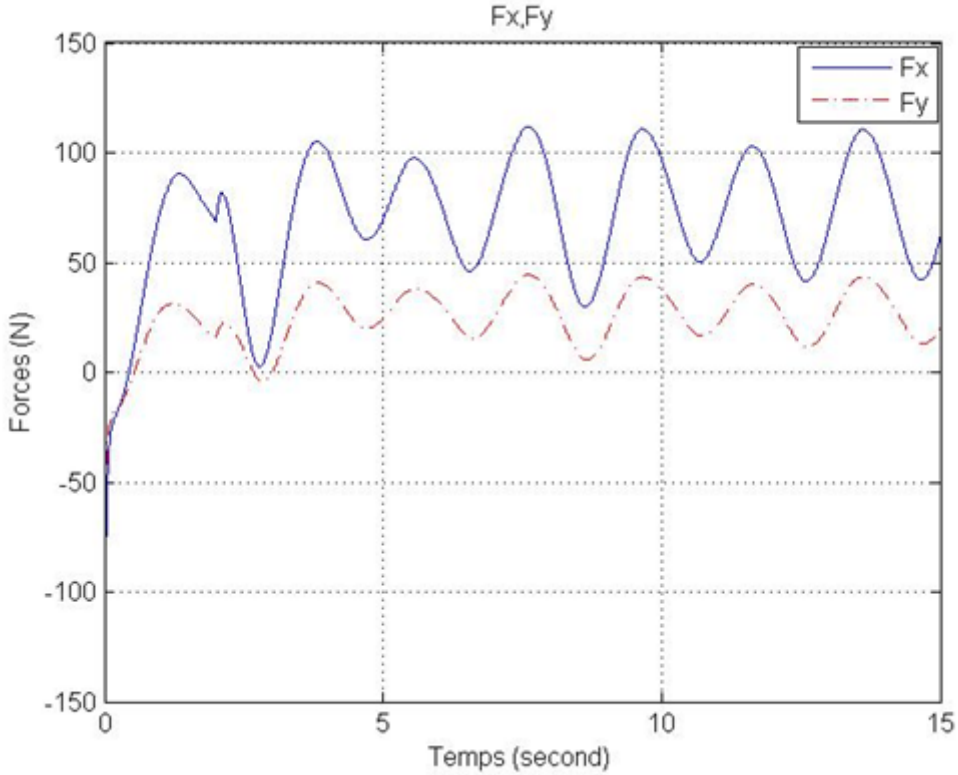


FIGURE 7.19 – Les forces appliquées sur l'effecteur suivant les deux axes X, Y

Les résultats de simulation obtenus confirment la possibilité d'utiliser effectivement ce mode pour commander notre robot à câbles et d'implémenter dans la carte de commande

Arduino MEGA 2560.

7.6 Interactions Homme-Machine(IHM) :

Les Interactions Homme-Machine (IHM) définissent les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine. Pour que cette communication soit la plus simple à faire et à réaliser, on utilise différents éléments. Les périphériques d'entrée, comme le clavier, la souris, le microphone et le scanner,qui permettent à l'homme de donner des renseignements ou des ordres à la machine. Les périphériques de sortie comme l'écran, des diodes, les haut-parleurs et l'imprimante, permettent à la machine de répondre aux ordres et d'afficher des informations.

Et comme tout travail de robotique, l'interaction entre l'utilisateur et le robot est une phase très importante, dans notre projet l'IHM nous permettons d'établir un manière de communication entre le robot et l'utilisateur(Assistant), pour cela, nous avons développé une application Java sous logiciel Processing (open source software), afin qu'un assistant puisse contrôler l'organe terminal du robot.

Pour cette objectif, on utilise la souris (périphérique d'entrée) comme un outil de commande qui sert à transférer les valeurs désirées X_d, Y_d. Comme montre le schéma dans la figure (7.20) ci-dessous :

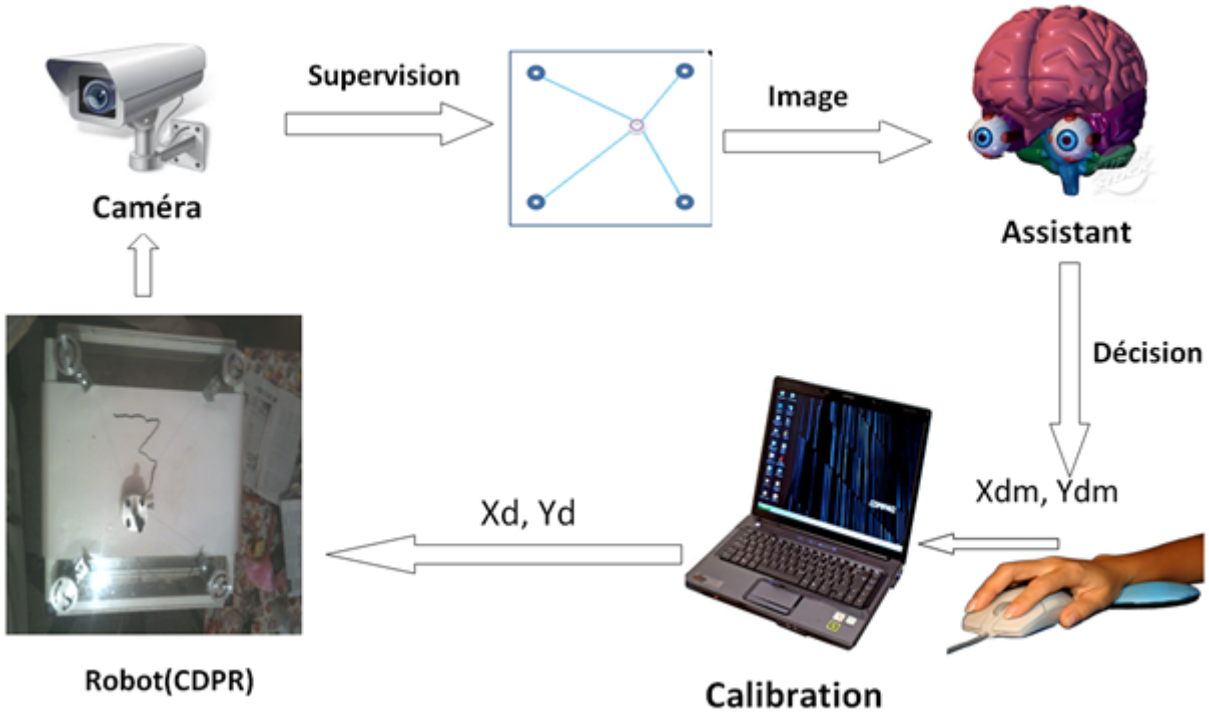


FIGURE 7.20 – Les différentes étapes pour l'utilisation du robot

L'application qui nous avons développé permet à l'assistant de contrôler la trajectoire de l'organe terminal ainsi de superviser ce dernier en temps réel.

Le robot suis la même trajectoire saisir par l'utilisateur avec même vitesse et accélération des coordonnées de la souris tel que X_{dm}, Y_{dm} sont les vecteurs de position, vitesse et accélération suivant les deux axes.

X_{dm}=[x_{dm}, v_{xdm}, a_{xdm}]^T , Y_{dm}=[y_{dm},v_{ydm},a_{ydm}]^T , l'indice ≪dm≫ signifie :*desired mouse.*

La calibration en robotique consistant à déterminer les valeurs réelles des paramètres cinématiques et dynamiques du robot.

Un robot calibré a une précision de positionnement absolue plus élevée que celle d'un non calibrée, c'est à dire, la position réelle de l'organe terminal effecteur du robot correspond mieux à la position calculée à partir du modèle mathématique du robot. Précision de positionnement absolue est particulièrement pertinente dans le cadre de robot de l'interchangeabilité et de la programmation hors ligne des applications de précision.

Pour notre application la calibration entre les coordonnées désirées via le périphérique d'entrée (la souris) et les coordonnées désirées réelles, elles ont calibrée de façon expérimentale après centaines d'essai tel que :

Si X_{dm} la position de la souris suivant l'axe X $\Longrightarrow$ X_d = (-1.0320496) * X_{dm}

Si Y_{dm} la position de la souris suivant l'axe Y $\Longrightarrow$ Y_d = (-2.09) * Y_{dm} Pour une explication claire voir la figure (7.21).

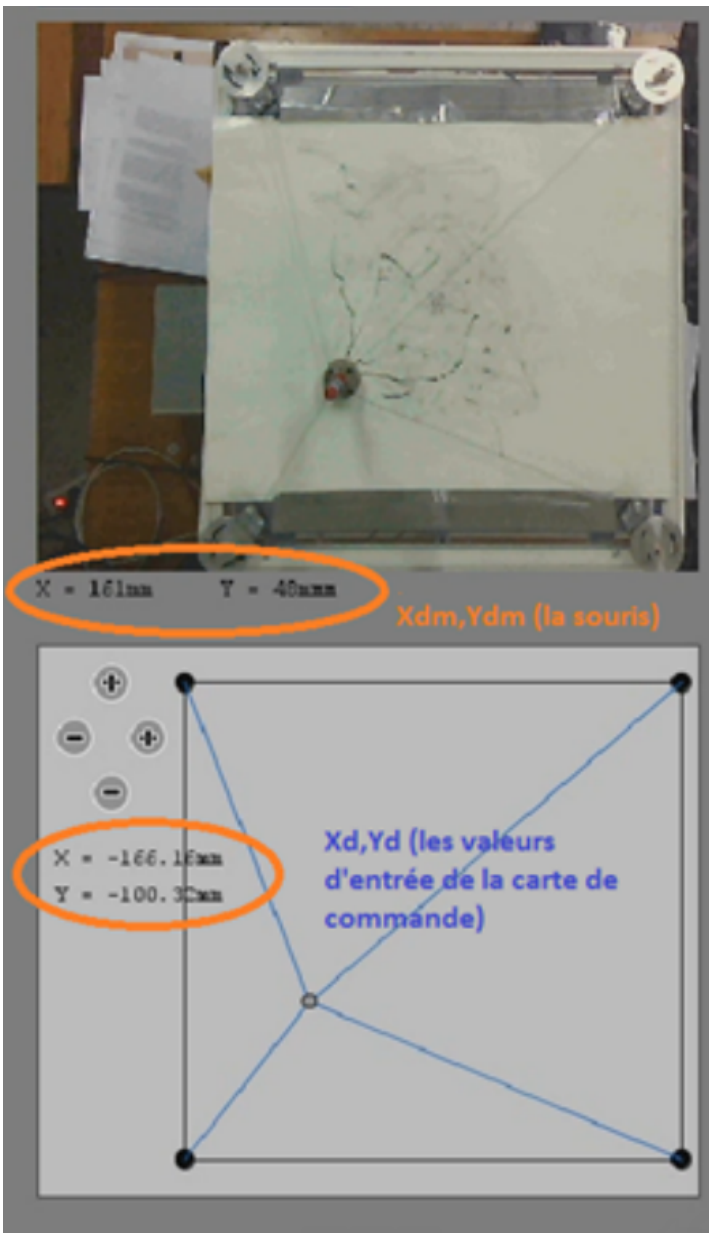


FIGURE 7.21 – Calibration des cordonnées

7.7 Montage final du robot

La conclusion du travail effectué pendant tout l'année présenté dans la figure (7.22) ci-dessous :

FIGURE 7.22 – Mise en œuvre pratique du robot

1. Étudiant joue le rôle d'handicapé ;
2. Assistant ;
3. Robot parallèle planaire à 4 câbles ;
4. L'organe terminal (support pour le stylo de l'écriture) ;
5. Caméra de supervision ;
6. Micro-ordinateur avec logiciel d'IHM installé ;
7. Alimentation en courant électrique de 12 volts continu ;
8. Câble USB de communication série entre le Robot et le PC ;
9. Multimètre pour superviser la tension d'alimentation du robot.

Le système c'est un robot parallèle à quatre câbles destiné à la réhabilitation des bras pour but d'aider les gents à des besoins spéciaux à l'écriture. Équipé d'un logiciel d'IHM (Interaction Homme-Machine) pour assister le déroulement normal d'opération.

Les figures suivantes montres quelques exercices qui nous avons faits :

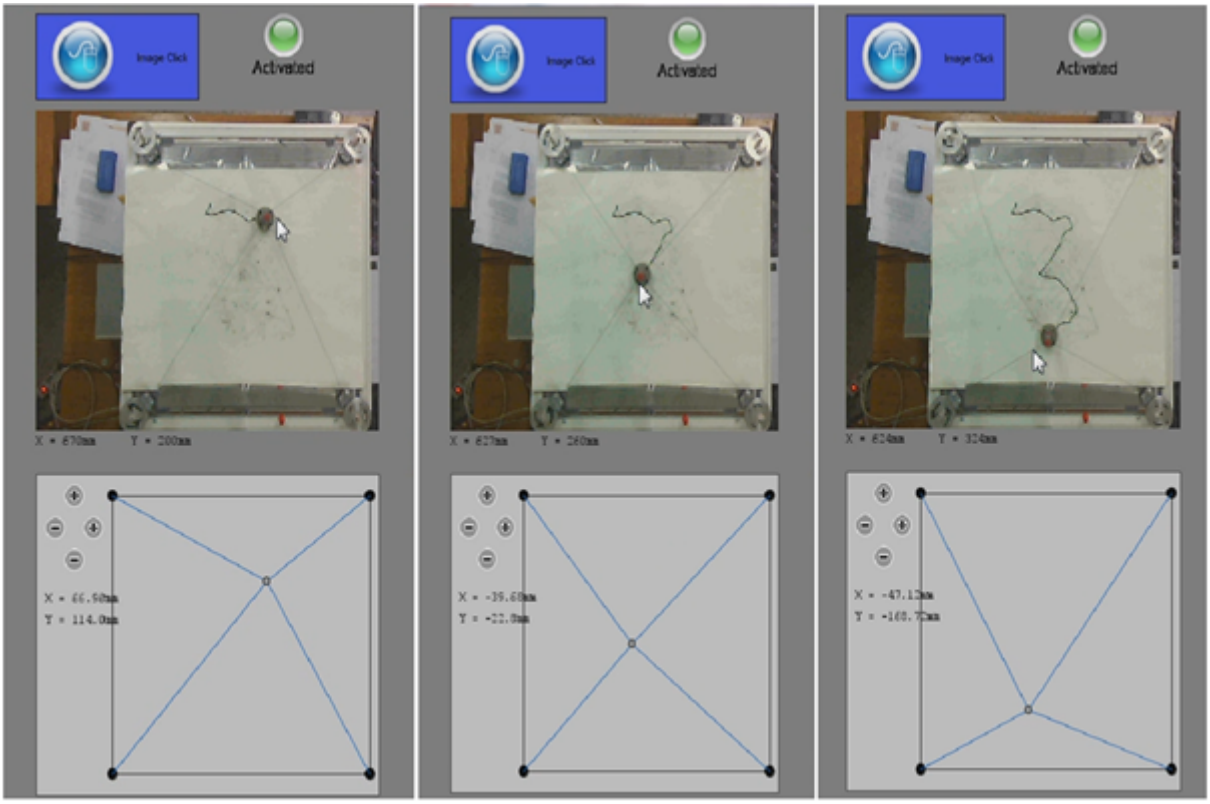


FIGURE 7.23 – Écriture de nombre 3 sans entraînement du bras de l'étudiant handicapé

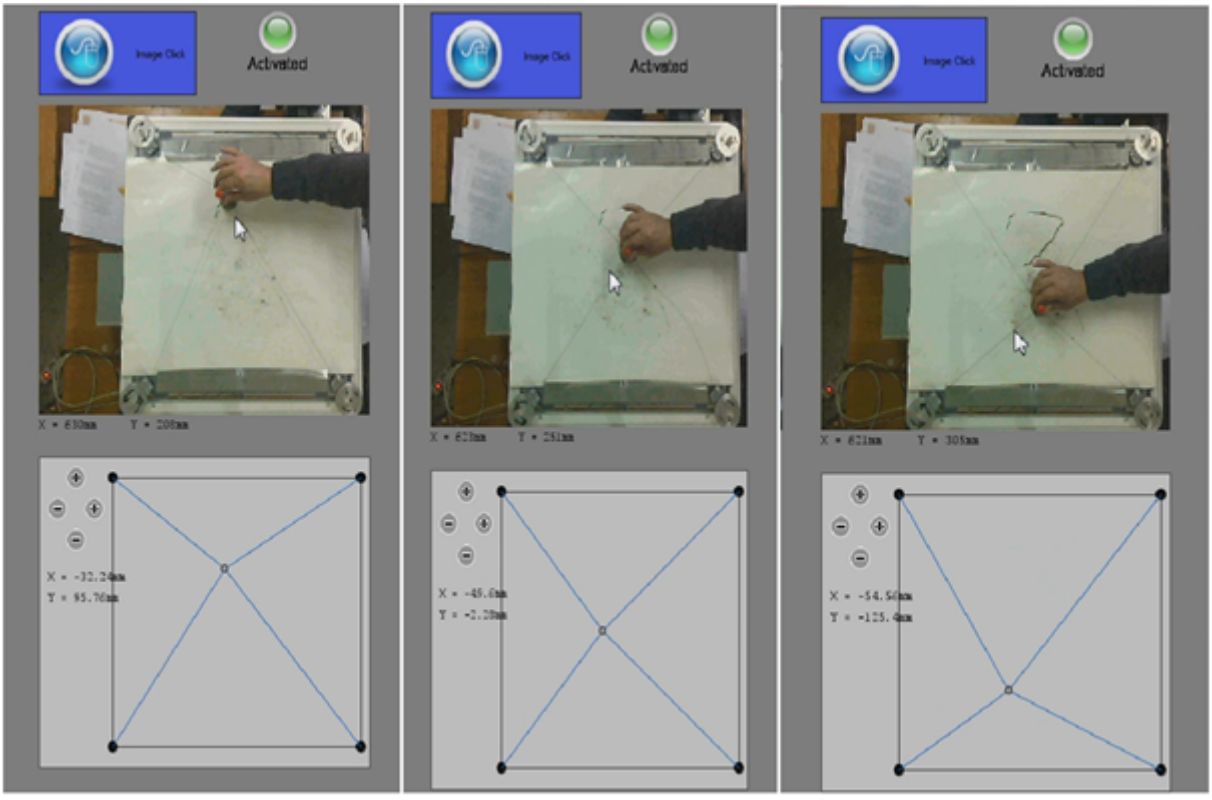


FIGURE 7.24 – Écriture de nombre 3 avec entraînement du bras de l'étudiant handicapé

7.8 Conclusion

Dans ce chapitre on a détaillé les étapes pour la réalisation et la commande du robot parallèle à quatre câbles pour la réhabilitation des bras pour but d'aider les gents à des besoins spéciaux à l'écriture. Aussi nous avons présenté une méthode simple et efficace pour l'interaction entre l'utilisateur et le robot.

Conclusion générale

Dans le cadre de ce projet de fin d'études, nous avons étudié et réalisé un prototype d'un robot à quatre câbles destiné à la réhabilitation des bras pour but d'aider les à gents des besoins spéciaux à l'écriture.

- Nous avons étudié la structure et le comportement des robots parallèles à 4 câbles.
- Nous avons développé un programme pour la commande en boucle ouverte en position de l'effecteur terminal basé sur le modèle géométrique inverse. Étant donnée la position désirée introduite, le programme détermine les longueurs des câbles et les angles, il permet la visualisation graphique du système.
- Par la méthode de Lagrange et Newton-Euler, on a établi les équations différentielles du mouvement qu'on a programmées sur Matlab pour calculer les couples optimums à délivrés pour les quatre moteurs, afin de faire suivre à l'effecteur une trajectoire donnée. Par une Co-simulation Simulink-MSC-ADAMS, on a validé nos calculs par un modèle virtuel du robot, réalisé sur ce logiciel de simulation.
- Pour l'analyse statique des force et calcule des couples optimums compte tenu de la spécification de notre système, à savoir que le nombre de câbles est supérieur au nombre de degrés de liberté, nous avons utilisé la méthode de Moore-Penrose, pour déterminer les tensions appliquées sur les câbles par les moteurs. A titre d'exemple, par la méthode du Simplexe, nous avons simulé les tensions optimales nécessaires pour que l'effecteur poursuive une trajectoire circulaire tout en imposant une force constante le long du parcours. On notera que ce même exemple a été étudié par [9], des profils comparables ont été obtenus.
- Nous avons entrepris l'étude du comportement du système étudié en boucle ouverte, le robot est modélisable par système d'équations différentielles non linéaire. Sa réponse aux signaux de type impulsion montre que son comportement est également de type « plastique ». La simulation du comportement du système a été effectuée sous Matlab et Simulink.
- Nous avons également entrepris l'étude de la commande de nos systèmes en boucle fermée. compte tenu de la non linéarité de notre système, nous avons simulé la commande par mode glissant. La simulation de ce mode de commande a permis de constater que notre robot est effectivement un système suiveur.

- Nous avons réalisé le fruit de notre travail théorique par une réalisation pratique d'un prototype à quatre câbles pour but d'aider les gents à des besoins spéciaux à l'écriture. La mise en œuvre des quelques expériences, nous a donnée des résultats très encourageants et ambitieux, mais à cause du temps limité et les contraintes technologiques, nous n'avons pas pu réaliser tous les objectifs, le projet reste ouvert pour le développement de l'implémentation d'un régulateur robuste (mode glissant, logique flou, Hinfini. . . etc.), aussi au niveau de l'interaction homme machine par traitement d'image et de parole .

Bibliographie

[1] Xiaoqiang Tang, *"An overview of the development for cable-driven parallel manipulator"*. The State Key Laboratory of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, P. R. of CHINA.

[2] M. Ceccarelli C. Avila Carrasco E. Ottaviano. *"Error Analysis and Experimental Tests of CATRASYS (Cassino Tracking System"*. DiMSAT - Laboratory of Robotics and Mechatronics, University of Cassino, Via Di Biasio 43, 03043, Cassino (Fr), Italy.

[3] SAMUEL BOUCHARD. *"Géométrie des Robots Parallèles Entraînés par des Câbles"*. Faculté des sciences et de génie, Université Laval, Québec, 2008.

[4] Surdilovic D. and Bernhardt R. *"STRIN-MAN : A New Wire-Robot ait Rehabilitation"*. Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA), New Orleans, 2004, pp. 2031-2036.

[5] Parallel Robot Projects at Ohio University. Robert L. Williams II. *"Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators"*. October 3-4, 2002, Quebec City, Canada.

[6] Jorge Juan Gil and Emilio Sanchez. *"Lower-Limb Robotic Rehabilitation : Literature Review and Challenges"*. 2011, Applied Mechanics Department, CEIT, Paseo Manuel Lardizabal 15, 20018 San Sebastian, Spain.

[7] Tobias Bruckmann Editors. *"Cable-Driven Parallel Robots book"*. Universität Duisburg-Essen, Duisburg Germany, page 119.

[8] Robert L. Williams II and Paolo allina. *"Planar Cable-Direct-Driven Robots, Part I : Kinematics and Statics"*. 2001 ASME Design Technical Conferences 27^th Design Automation Conference September 9-12, 2001, Pittsburgh, PA.

[9] Robert L. Williams II and Paolo Gallina and Aldo Rossi. *"Planar Cable-Direct-Driven Robots, Part II : Dynamics and Control"*. 2001 ASME Design Technical Conferences 27^th Design Automation Conference September 9-12, 2001, Pittsburgh, PA.

[10] Robert L. Williams II and Paolo Gallina and Aldo Rossi. *"Planar Cable-Direct-Driven Robots"*. Journal of Robotic Systems Vol. 20, N^o. 3, pp. 107-120, 2003.

[11] Richard C. Dorf. *"OPTIMAL CONTROL SYSTEMS"*. Book edition, 2003.

[12] Deghboudj Imen. *"Commande des Systèmes Non Linéaires par Mode Glissant d'Ordre Supérieur"*. Faculté des Sciences de la Technologie, Universite de Constantine, Algérie.

[13] BENYETTOU Mohamed. *"L'optimisation par Essaims de Particules. Le Concept Tiré des Nuées d'Oiseaux"*. Université des Sciences et de la Technologie d'Oran, Algérie.

[14] Yann COOREN. *"Perfectionnement d'un Algorithme Adaptatif d'Optimisation par Essaim Particulaire. Applications en Génie Médical et en Electronique"*. Université Paris 12.

[15] James Blondin. *"Particle Swarm Optimization : A Tutorial"*. September 4, 2009.

[16] Antoine Dutot et Damien Olivier. *"Optimisation par Essaim de Particules. Application au Problème des n-Reines"*. Laboratoire Informatique du Havre, Université du Havre.

[17] Magnus Erik Hvass Pedersen. *"Good Parameters for Particle Swarm Optimization"*. Hvass Laboratories Technical Report N^o. HL1001, 2010.

[18] Farres Boudjema. *"Optimisation et Recherche Optimal"*. Cours de 3^e*me* année Automatique, L'Ecole Nationale Polytechnique d'Alger, Algérie.

[19] Ionel Sorin CIUPERCA. *"Cours Optimisation"*. Cours à l'ISFA, en M1SAF.

[20] Etienne Dombre ,Wisama Khalil. *"Modeling, Performance Analysis and Control of Robot Manipulators"*.

[21] Mouhamed Tadjine. *"Automatique Avancée"*. Cours de 3^e*me* année Automatique, 2013, L'Ecole Nationale Polytechnique d'Alger, Algérie.

[22] W.B.Gao and J.C.Hung. *"Variable Structure Control of Nonlinear Systems : A New Approch"*, IEEE.

[23] A. Boubakir , F. Boudjema , C. Boubakir , N. Ikhlef. *"Loi de Commande par Mode de Glissement avec une Surface de Glissement Non Linéaire Appliquée au Système Hydraulique à Réservoirs Couplés"*. 4th International Conference on Computer Integrated Manufacturing, CIP'2007

[24] Jing Lei, XinWang, Yu-Mei She, and Tian-Jun Zhang. *"Variable Structure Disturbance Rejection Control for Nonlinear Uncertain Systems with State and Control Delays via Optimal Sliding Mode Surface Approach"*, 9 September 2013.

[25] Jean-Noël Montagné. *"Initiation à la Mise en Œuvre Matérielle et Logicielle de l'Arduino"*, Novembre 2006.

[26] www.electroschematics.com

[27] www.blog.3sigma.fr

[28] www.dfrobot.com

[29] www.arduino.cc

[30] fr.wikipedia.org/wiki/GNU$_C$*ompiler*$_C$*ollection*

[31] www.bacaterken.com

[32] www.dizart.fr/

[33] www.mscsoftware.com

[34] Arduino Mega 2560 Datasheet

[35] fr.flossmanuals.net

[36] Ian T. Young,Jan J. Gerbrands,Lucas J. van Vliet :Fundamentals of Image

[37] Marc Chabreuil : Les robots touchés par la vague verte, 29 Mai 2010. Processing

Annexe A

MSC-Adams

MSC-Adams est un logiciel d'analyse de systèmes mécaniques dont l'interface graphique est présentée sur la figure(7.25), qui entre dans le cadre multidisciplinaire IAO de MSC. MD Adams intègre des composants mécaniques, pneumatiques, hydrauliques, électroniques, et des technologies de contrôle des systèmes pour permettre aux ingénieurs de construire et tester des prototypes virtuels qui tiennent compte avec précision des interactions entre ces sous-systèmes.

Les industriels ont souvent du mal à appréhender les véritables performances de leurs systèmes avant d'être très avancés dans leur processus de conception. Les composants mécaniques, électriques et autres sous-systèmes sont validés séparément selon leurs critères spécifiques, mais les tests et la validation du système complet interviennent en aval, d'où un travail supplémentaire et des modifications techniques plus risquées et plus coûteuses que si elles avaient pu être faites en amont.

MD Adams augmente l'efficacité de l'ingénierie et réduit les coûts de développement de produits en permettant une validation systémique en amont. Les ingénieurs peuvent évaluer et gérer les interactions complexes entre disciplines y compris la cinématique, les structures, l'actionnement et les systèmes de contrôle/commande pour mieux optimiser les modèles en termes de performances, de sécurité et de confort. Avec ses nombreuses fonctionnalités d'analyse, MD Adams est optimisé pour les problèmes de grande taille, en profitant d'environnements de calcul à hautes performances.

Offertes :

• Simulation multidisciplinaire de systèmes mécaniques.
• Modélisation Conception et tests de véhicules.
• Intégration de corps flexibles.
• Une analyse de durabilité améliore la qualité du produit.
• Analyse vibratoire facilitée.

Industrielles de logiciel :

• Aérospatial & Défense : Systèmes de pilotage électronique, trains d'atterrissage et freins, systèmes d'alimentation électrique, turbomachines, vannes et commandes.
• Automobile : véhicules hybrides, systèmes de refroidissement/chauffage, transmissions, direction assistée.
• Administrations et marchés publics : dynamique de bras robotiques, système RCS (propulsion), dynamique de satellites et de systèmes de commande.

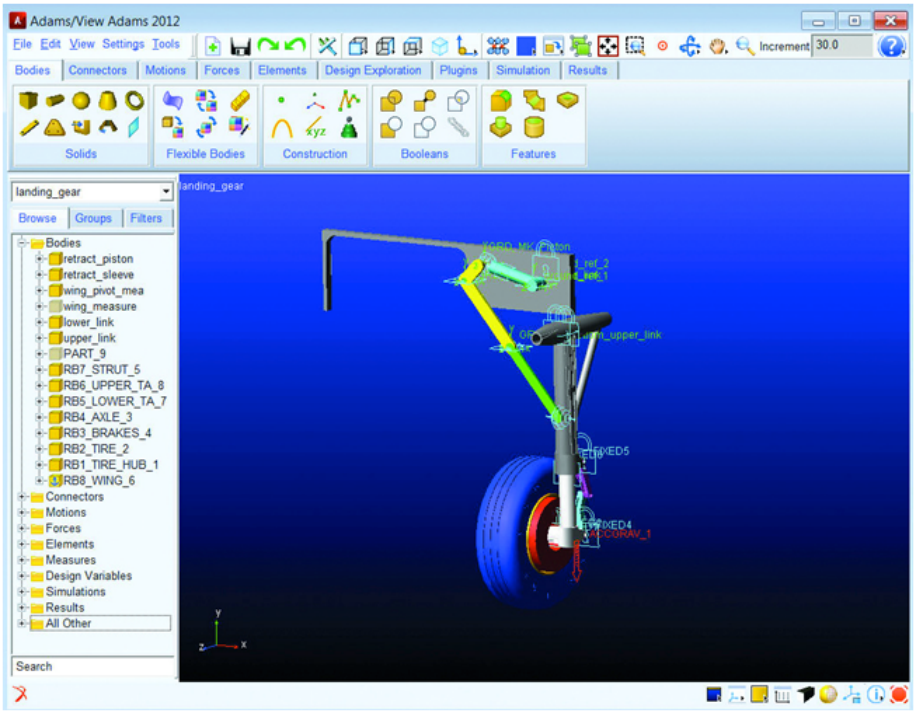
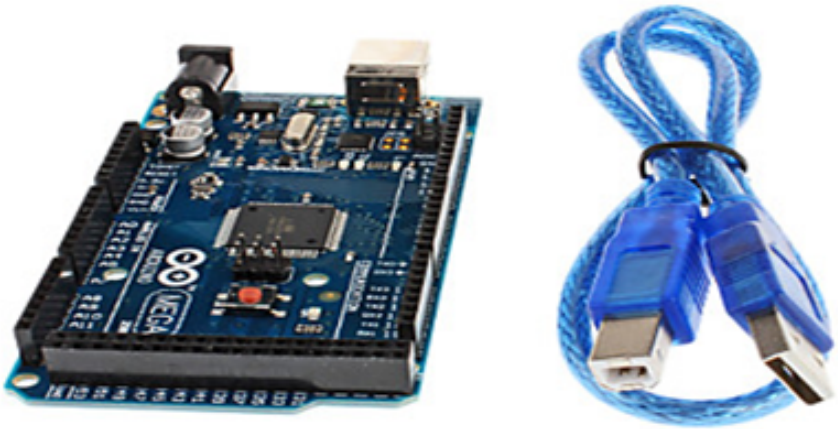


FIGURE 7.25 – Interface Graphique MSC-ADAMS.

• Transports : composants de systèmes hydrauliques, moteurs et transmission, lubrification de moteurs, chargeurs frontaux, transmission hydrostatique, circuits de démarrage, systèmes de commande électronique.

Annexe B

Arduino MEGA 2560



L'Arduino Mega 2560 est une carte microcontrôleur basée sur l'Microcontrôleurs . Il est doté de 54 broches d'entrée/sortie numériques (dont 14 peuvent être utilisées comme sorties MDI), de 16 entrées analogiques, de 4 émetteurs-récepteurs universels asynchrones (UART, ports de série de matériel), d'un oscillateur en cristal de 16 MHz (voire Figure (7.26)), d'une connexion USB, d'une prise de courant, d'une embase ICSP et d'un bouton de réinitialisation.

Il contient tout ce qui est nécessaire pour prendre en charge le microcontrôleur. Pour cela, branchez-le à un ordinateur au moyen d'un câble USB ou allumez-le avec une batterie ou un adaptateur CC/CA pour le démarrer. Le Mega est compatible avec la plupart des blindages conçus pour les Arduino Uno, Duemilanove ou Diecimila.

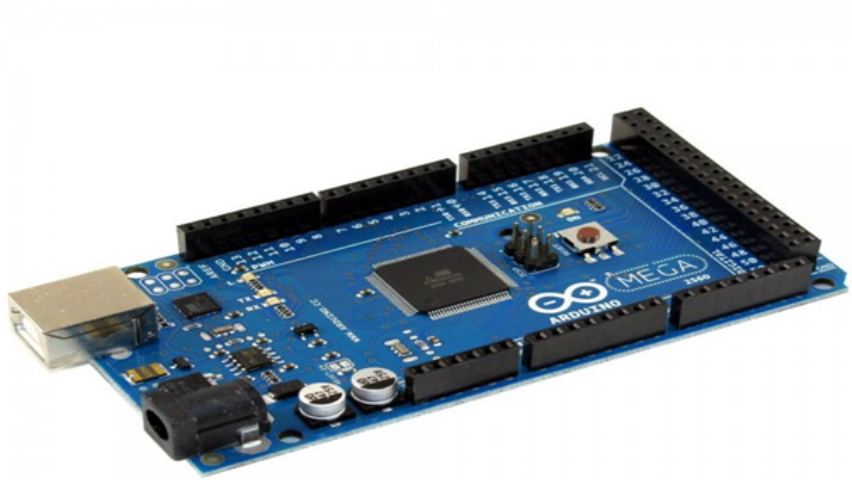


FIGURE 7.26 – Arduino MEGA 2560.

L'Arduino Mega peut être alimenté par la Câble USB de 6 Pieds ou par une Adaptateurs Muraux . Le Mega2560 diffère de toutes les cartes précédentes en ce qu'il n'utilise pas la puce pilote FTDI USB/série. Au lieu de cela, il dispose de l'Atmega8U2 programmé comme un convertisseur USB/série.

Chacune des 54 broches numériques sur le Mega peuvent être utilisées en tant qu'entrée ou sortie, en utilisant les fonctions pinMode(), digitalWrite(), et digitalRead() Il dispose également de 16 entrées analogiques, chacune d'elles disposant de 10 bits de résolution (c'est à dire 1 024 valeurs différentes).

L'Arduino Mega2560 dispose d'un certain nombre de moyens pour communiquer avec un ordinateur, un autre Arduino, ou d'autres microcontrôleurs. L'ATMEGA2560 fournit quatre

UART physiques pour des communications sérielles de type TTL (5 V).
Un ATmega8U2 sur la carte canalise l'un d'eux sur l'USB et fournit un port COM virtuel pour le logiciel sur l'ordinateur.

En outre, il accepte les communications I2C (TWI) et SPI. Le logiciel Arduino intègre une bibliothèque de câblage pour simplifier l'utilisation du bus I2C. Voyez la documentation sur le site Web de câblage pour plus de détails. Pour la communication SPI, utilisez la bibliothèque SPI.

L'Arduino Mega2560 peut être programmé au moyen de l'Arduino IDE gratuit et en "open source" par une connexion USB sans avoir besoin de matériel supplémentaire grâce à son chargeur de démarrage préchargé. Vous pouvez également éviter le chargeur de démarrage et programmer le microcontrôleur grâce à l'embase Programmeur PIC USB ICSP Cytron .
L'Arduino Mega2560 est conçu pour être compatible avec la plupart des Blindages Arduino .

Dimensions

10,16 cm x 5,33 cm avec le connecteur USB et la prise d'alimentation ressortant de la dimension de la base Spécifications.

Microcontrôleur : ATmega2560

- Tension de fonctionnement : 5 V.
- Tension d'entrée (recommandée) : 7 à 12 V.
- Tension d'entrée (limites) : 6 à 20 V.
- Broches E/S numériques : 54 (dont 14 fournissent la sortie MDI).
- Broches d'entrée analogiques : 16.
- Courant alternatif par broche d'E/S : 40 mA.
- Courant continu pour la broche de 3,3 V : 50 mA.
- Mémoire Flash : 256 Ko (dont 8 Ko utilisés par le chargeur initial de programme).
- SRAM : 8 Ko.
- EEPROM : 4 Ko.
- Vitesse de l'horloge : 16 MHz.

Annexe C

Arduino IDE (Integrated Development Environment)

L'environnement de programmation Arduino IDE (voir (7.27)) offre une interface simple et pratique pour programmer n'import quelle carte commande de firme Arduino.

Cependant, il existe quelques logiciels alternatifs qui permettent de programmer la carte Arduino.

Utiliser un langage de programmation qu'on maîtrise déjà permet de ne pas avoir à apprendre un nouveau langage pour programmer la carte Arduino. Cela permet aussi de réutiliser les librairies et programmes que l'on a éventuellement déjà développés pour d'autres familles de micro-contrôleurs. Pour les programmeurs confirmés, le langage C/C^{+}+ qui est traditionnellement utilisé pour programmer les micro-contrôleurs reste la solution la plus performante. D'autre part, si l'on possède des connaissances et l'on dispose de ressources techniques et de partenaires qui travaillent sur d'autres plateformes, rester sur celles-ci est peut-être un choix pertinent.

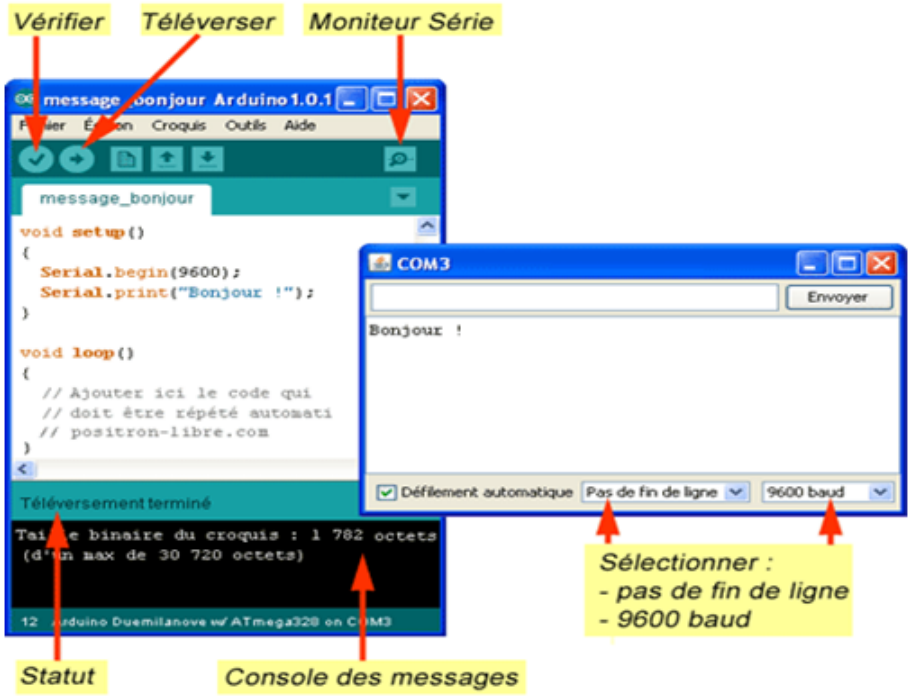


FIGURE 7.27 – Arduino Integrated Development Environment

Annexe D

Processing IDE (Integrated Development Environment)

Processing (PROCE55ING ou P5) est un environnement de programmation et un langage simple et complet, Il s'agit d'un logiciel libre (open-source), gratuit et multi-plateformes (Windows, Mac OS X et Linux).
L'environnement Processing (Integrated Development Environment ou IDE) dispose d'un éditeur de texte et d'une fenêtre pour visualiser les programmes (voir Figure (7.28)).

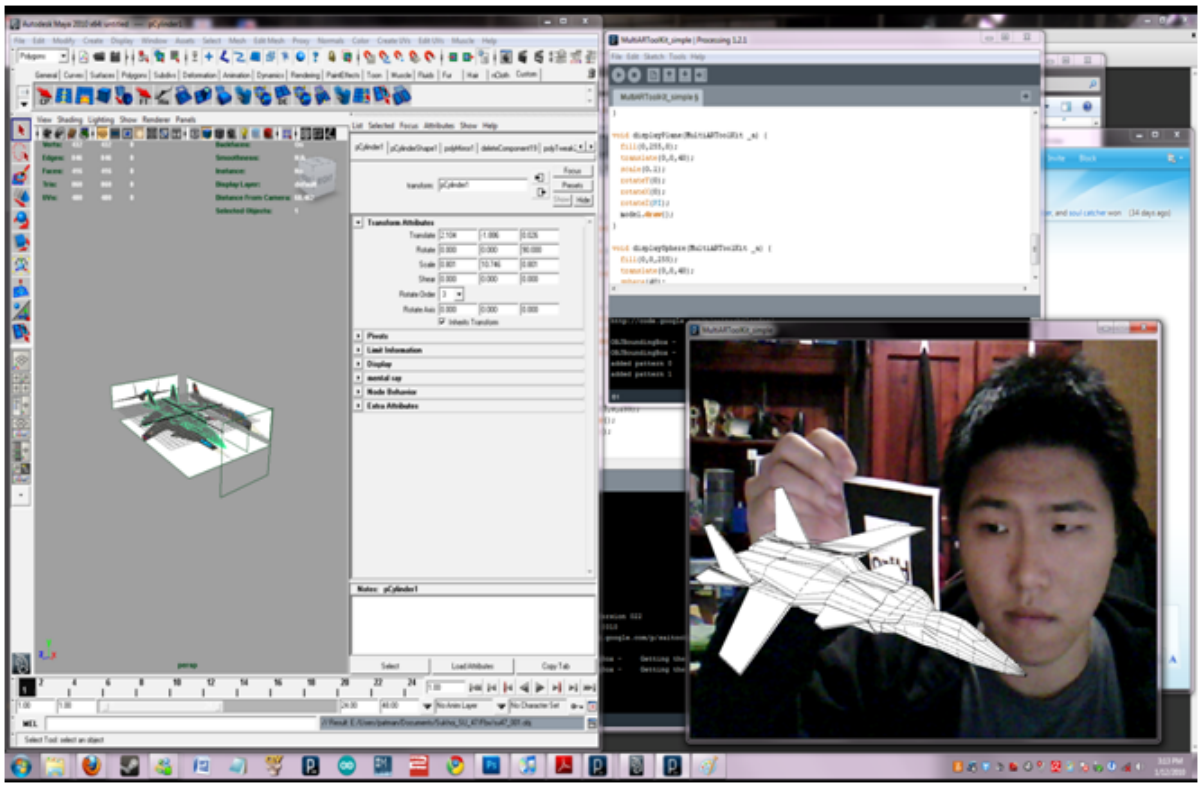


FIGURE 7.28 – Processing Integrated Development Environment.

Processing a été conçu par Casey Reas et Benjamin Fry comme outil de création et d'apprentissage fondamental à la programmation. Depuis son lancement en 2002, une véritable communauté s'est développée autour de PROCESSING, considérant le code comme matériau de création et de pratique artistique.

Ben Fry et Casey Reas sont d'ex-étudiants de John Maeda à l'Aesthetics & Computation Group du MIT.
John Maeda étant lui-même auteur de Design By Numbers, environnement précurseur d'apprentissage fondamental, on peut dire que Processing en est le descendant direct.

Processing a reçu le prix Golden Nica à Ars Electronica 2005 (catégorie NetVision).

Le logiciel possède un moteur d'affichage 2D/3D performant et permet la création de programmes visuels interactifs ou génératifs. De nombreux codes-source sont partagés par leurs auteurs, constituant ainsi une documentation variée et conviviale, propice à des expériences de toute nature.

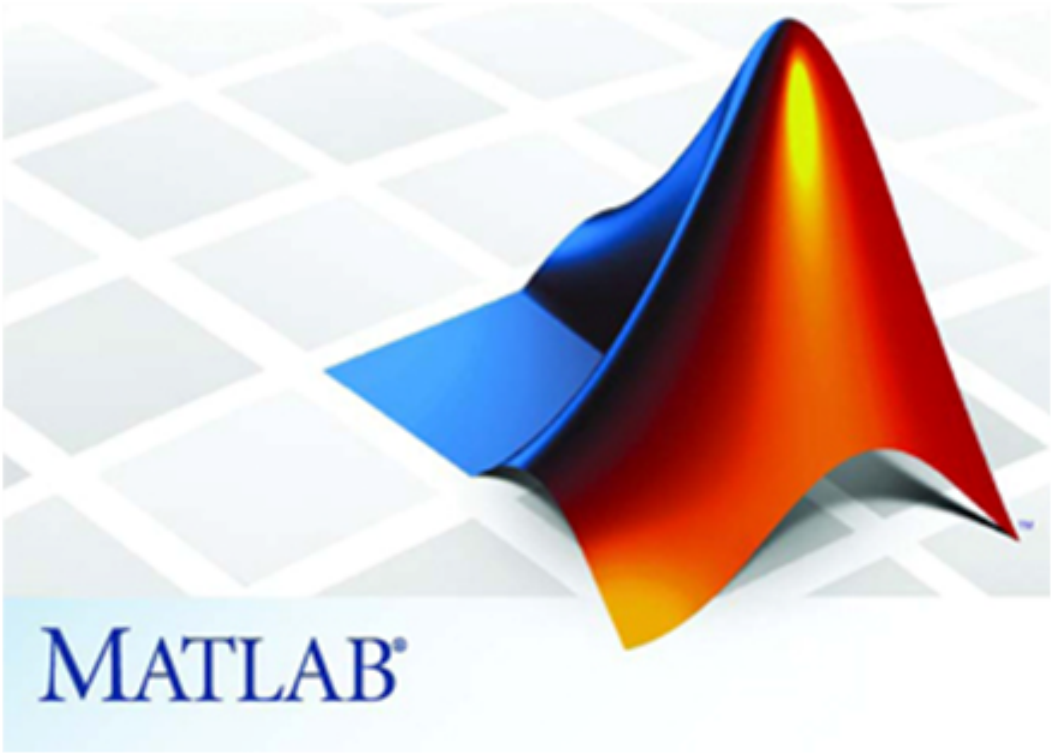
Il est fourni avec un bon nombre d'exemples, accessibles via l'environnement lui-même : par File ¿ Sketchbook ¿ examples.

Une galerie et un forum sont disponibles, alimentée par une communauté grandissante d'utilisateurs et d'artistes renommés tels que Golan Levin, Georges Legrady, ART+COM, Marius Watz, Martin Wattenberg, Jared Tarbell, Marc Napier, Lia, Ed Burton (SodaPlay), etc. . .

Ces raisons font de Processing un outil idéal pour apprendre l'art du code.

Annexe E

Les Programmes Matlab



Programme 1 : Représentation graphique de l'organe terminal.

```
%====Représentation graphique de l'organe terminal=======
%=========================================================
=================
clear all;clc;
% Insérer la longueur du côté du carré:
Lb=input('Insérer la longueur du côté du carré Lb: ');
% Les coordonnées des sommets du carré sont calculées à
partir de Lb,
% donc ne pas touchez les valeurs suivantes!
xa1=-Lb/2; ya1=-Lb/2;
xa2=Lb/2; ya2=-Lb/2;
xa3=Lb/2; ya3=Lb/2;
xa4=-Lb/2; ya4=Lb/2;
L10=Lb*sqrt(2)/2 ;
% Les coordonnées cartésiennes du point mobile:
x=input('insérer coordonnées cartésiennes du point mobile
X :');
y=input('insérer coordonnées cartésiennes du point mobile
Y :');
% Représentation graphique du carrée et le point mobile
lié aux sommets:
l1=sqrt((x-xa1).^2+(y-ya1).^2);
l2=sqrt((x-xa2).^2+(y-ya2).^2);
l3=sqrt((x-xa3).^2+(y-ya3).^2);
l4=sqrt((x-xa4).^2+(y-ya4).^2);
theta_a=atand(abs(y-ya1)/abs(x-xa1));
theta_b=180-atand(abs(y-ya2)/abs(xa2-x));
theta_c=270-atand(abs(xa3-x)/abs(y-ya3));
theta_d=360-atand(abs(y-ya4)/abs(xa4-x));
X=[xa1 ya1;xa2 ya2;xa3 ya3;xa4 ya4;x y];
dt = DelaunayTri(X);
valeurs=[l1 l2 l3 l4];
disp(['les valeurs L1 L2 L3 L4 les longeurs des càbles
sont respectivement: ', num2str(valeurs)]);
valeurs_Angles=[theta_a theta_b theta_c theta_d];
disp(['les valeurs des Angels des rotation sont
respectivement: ', num2str(valeurs_Angles)]);
triplot(dt);hold on;plot(0,0,'+');hold off;
axis equal
```

Programme 2 : Trajectoire circulaire

```
%======================carré + cercle====================
 clear all;clc;

% Insérer la longueur du côté du carré:
Lb=input('Insérer la longueur du côté du carré Lb: ');
% Les coordonnées des sommets du carré sont calculées à
partir de Lb,
% donc ne pas touchez les valeurs suivantes!
xa1=-Lb/2; ya1=-Lb/2;
xa2=Lb/2; ya2=-Lb/2;
xa3=Lb/2; ya3=Lb/2;
xa4=-Lb/2; ya4=Lb/2;
% Insérer le rayon du cercle parcouru par l'organe
terminal:
% PS: Ne pas dépasser Lb/2, sinon le point mobile sort du
contour!
Rmax=Lb/2;
R=input('Insérer le rayon du trajectoir cerclé par le
point mobile:  ');
Theta=input('Insérer L"Angle en degrée du trajectoir
cerclé par le point mobile:  ');
xxp=R*cosd(Theta)  ;
yyp=R*sind(Theta)  ;
X=[xa1 ya1;xa2 ya2;xa3 ya3;xa4 ya4;xxp yyp];
dt = DelaunayTri(X);
triplot(dt);hold on;plot(0,0,'+');hold on;
s=0:.01:Theta;
x=R.*cosd(s);
y=R.*sind(s);
plot(x,y,'r');hold off
axis equal
```

Programme 3 : Variations des longueurs

```
%================ Variations des longueurs ==============
%=========================================================
================
clear all;clc;
% Insérer la longueur du côté du carrée:
Lb=input('Insérer la longueur du côté du carré Lb: ');
% Les coordonnées des sommets du carrée sont calculées à
partir de Lb,
% donc ne pas touchez les valeurs suivantes!
xa1=-Lb/2; ya1=-Lb/2;
```

```
 xa2=Lb/2; ya2=-Lb/2;
xa3=Lb/2; ya3=Lb/2;
xa4=-Lb/2; ya4=Lb/2;
% Insérer le rayon du cercle parcouru par le point mobile:
Rmax=Lb/2;
R=input('Insérer le rayon du trajectoir cerclé par
l"organe terminal:  ');
Theta=input('Insérer L"Angle en degrée du trajectoir
cerclé par l"organe terminal:  ');
% Calcul des distances séparant le point mobile des
sommets:
for theta=1:Theta
[x,y]=pol2cart(deg2rad(theta),R);
l1(theta)=sqrt((x-xa1).^2+(y-ya1).^2)
l2(theta)=sqrt((x-xa2).^2+(y-ya2).^2)
l3(theta)=sqrt((x-xa3).^2+(y-ya3).^2)
l4(theta)=sqrt((x-xa4).^2+(y-ya4).^2)
end
% Représentation polaires des variations de ces distances
par rapport à
% l'angle polaire que fait le point mobile:
K=menu('choisir la Representation de variations de
longeurs des cables   :','1- Représentation polaires
','2-Représentation graphique');
if(K==1)
t=[1:Theta].*pi./180;
polar(t,l1),grid;hold on;
polar(t,l2,'r');hold on;
polar(t,l3,'g');hold on;
polar(t,l4,'y');hold off;
end
% Représentation graphique des variations de ces distances
par rapport à
% l'angle polaire que fait l’organe terminal :
if (K==2)
    t=1:Theta;

plot(t,l1); hold on;
plot(t,l2,'r'); hold on;
plot(t,l3,'g');  hold on;
plot(t,l4,'y');
title('Représentation graphique des variations de ces
distances par rapport à l"angle polaire que fait le point
mobile');
xlabel( 'Theta');
ylabel('Les longeurs des Cable')
hold off;
```

```
 end
```

Programme 4 : Variations des angles.

```
%=========================================================
================
%============ Variations des angles======================
%=========================================================

clear all;clc;
% Insérer la longueur du côté du carrée:
Lb=input('Insérer la longueur du côté du carré Lb:: ');
% Les coordonnées des sommets du carrée sont calculées à
partir de Lb,
% donc ne pas touchez les valeurs suivantes!
xa1=-Lb/2; ya1=-Lb/2;
xa2=Lb/2; ya2=-Lb/2;
xa3=Lb/2; ya3=Lb/2;
xa4=-Lb/2; ya4=Lb/2;
% Insérer le rayon du cercle parcouru par le point mobile:
Rmax=Lb/2;
R=input('Insérer le rayon du trajectoir cerclé par
l"organe terminal:  ');
Theta=input('Insérer L"Angle en degrée du trajectoir
cerclé par l"organe terminal:  ');
% Calcul des distaces séparant le point mobile des
sommets:
for theta=1:Theta
[x,y]=pol2cart(deg2rad(theta),R);
theta_a(theta)=atand(abs(y-ya1)/abs(x-xa1))
theta_b(theta)=180-atand(abs(y-ya2)/abs(xa2-x))
theta_c(theta)=270-atand(abs(xa3-x)/abs(y-ya3))
theta_d(theta)=360-atand(abs(y-ya4)/abs(xa4-x))
end
t=1:Theta ;
plot(t,theta_a);hold on;
plot(t,theta_b,'r');hold on;
plot(t,theta_c,'g');hold on;
plot(t,theta_d,'y');hold off

%=========== Calcules couples optimums====================
%%% données du probleme
clear all;
r=0.02;R=0.1;lb=0.32; m=0.2; J1=0.0008; J2=0.0008;
J3=0.0008;J4=0.0008; C1=0.01; C2=0.01; C3=0.01;C4=0.01;
a= 500;
```

```
t=-5/a;
for i=1:a+1
    t=t+5/a
if t<2.5

%%%%la trajectoire
x=-(0.016/2)*t^2+0.1;
y=(0.016/2)*t^2;
%%%vitesse liniaire
vx=-(0.016)*t;
vy=(0.016)*t;
%%acceleration
acx=-0.016;
acy=0.016;

 else

%%%%la trajectoire
x=(0.016/2)*t^2-0.08*t+0.2;
y=-(0.016/2)*t^2+0.08*t-0.1;
     %%%vitesse liniaire
     vx=(0.016)*t-0.08;
     vy=-(0.016)*t+0.08;
     %%acceleration
     acx=(0.016);
     acy=-(0.016);

end

%%%%l'angle theta
    A10x=-lb/2;
    A20x=lb/2;
    A30x=lb/2;
    A40x=-lb/2;
    A10y=-lb/2;
    A20y=-lb/2;
    A30y=lb/2;
    A40y=lb/2;
    theta1=atan((y-A10y)/(x-A10x));
    theta2=atan((y-A20y)/(x-A20x))+pi;
    theta3=atan((y-A30y)/(x-A30x))+pi;
    theta4=atan((y-A40y)/(x-A40x))+2*pi;

    deteta_1=(vy*(x-A10x)-vx*(y-A10y))*(cos(theta1)/(x-
A10x))^2;
    deteta_2=(vy*(x-A20x)-vx*(y-A20y))*(cos(theta2)/(x-
A20x))^2;
```

```
    deteta_3=(vy*(x-A30x)-vx*(y-A30y))*(cos(theta3)/(x-
A30x))^2;
    deteta_4=(vy*(x-A40x)-vx*(y-A40y))*(cos(theta4)/(x-
A40x))^2;
    %vitesse et acceleration angulaire des poulies
    dbeta=-1/r*[cos(theta1) sin(theta1);cos(theta2)
sin(theta2);cos(theta3) sin(theta3);cos(theta4)
sin(theta4)]*[vx;vy];
    ddbeta=1/r*([deteta_1*sin(theta1) -
deteta_1*cos(theta1);deteta_2*sin(theta2) -
deteta_2*cos(theta2);deteta_3*sin(theta3) -
deteta_3*cos(theta3);deteta_4*sin(theta4) -
deteta_4*cos(theta4)]*[vx;vy]-[cos(theta1)
sin(theta1);cos(theta2) sin(theta2);cos(theta3)
sin(theta3);cos(theta4) sin(theta4)]*[acx;acy]);
    %%%mareice s
    S=[-cos(theta1) -cos(theta2) -cos(theta3) -
cos(theta4);
        -sin(theta1) -sin(theta2) -sin(theta3) -
sin(theta4)];

   %%%%matrice masssse

M11=m*r+(J1*(cos(theta1))^2+J2*(cos(theta2))^2+J3*(cos(the
ta3))^2+J4*(cos(theta4))^2)/r;
M12=((J1*cos(theta1)*sin(theta1))+(J2*cos(theta2)*sin(thet
a2))+J3*cos(theta3)*sin(theta3)+(J4*cos(theta4)*sin(theta4
)))/r;
M21=M12;
M22=m*r+(J1*(sin(theta1))^2+J2*(sin(theta2))^2+J3*(sin(the
ta3))^2+J4*(sin(theta4))^2)/r;

%%%%matrice inertie

N11=(cos(theta1)*(C1*cos(theta1)-
J1*deteta_1*2*sin(theta1))+cos(theta2)*(C2*cos(theta2)-
J2*deteta_2*2*sin(theta2))+cos(theta3)*(C3*cos(theta3)-
J3*deteta_3*2*sin(theta3))+cos(theta4)*(C4*cos(theta4)-
J4*deteta_4*2*sin(theta4)))/r;


N12=(C1*cos(theta1)*sin(theta1)+J1*deteta_1*(cos(theta1)^2
)+C2*cos(theta2)*sin(theta2)+J2*deteta_2*(cos(theta2)^2)+C
3*cos(theta3)*sin(theta3)+J3*deteta_3*(cos(theta3)^2)+C4*c
os(theta4)*sin(theta4)+J1*deteta_4*(cos(theta4)^2))/r;
N21=N12;
```

```
 %%%%N21=(C1*sin(theta1)*cos(theta1)-
J1*deteta_1*(cos(theta1)^2-
sin(theta1)^2)+C2*sin(theta2)*cos(theta2)-
J2*deteta_2*(cos(theta2)^2-
sin(theta2)^2)+C3*sin(theta3)*cos(theta3)-
J3*deteta_3*(cos(theta3)^2-sin(theta3)^2))/r;
N22=(sin(theta1)*(C1*sin(theta1)+J1*2*deteta_1*cos(theta1)
)+sin(theta2)*(C2*sin(theta2)+J2*2*deteta_2*cos(theta2))+s
in(theta3)*(C3*sin(theta3)+J3*2*deteta_3*cos(theta3))+sin(
theta4)*(C4*sin(theta4)+J4*2*deteta_4*cos(theta4)))/r;

    %%% la solution par la pseudo invers
    F=[M11 M12;M21 M22]*[acx;acy]+[N11 N12; N21
N22]*[vx;vy];
    taumin=max([J1 0 0 0;0 J2 0 0;0 0 J3 0;0 0 0
J4]*ddbeta+[C1 0 0 0;0 C2 0 0;0 0 C3 0;0 0 0
C4]*dbeta,0.0005);
%       figure(2);
%     hold on
%   plot(t,taumin(1),'-.g');grid
%   plot(t,taumin(2),'--k');grid
%   plot(t,taumin(3),'-.r');grid
%   plot(t,taumin(4),'-.b');grid

    taup=pinv(S)*F;
    N=null(S);
    A(:,1)=([N(1,1) N(1,2);N(2,1) N(2,2)])\[taumin(1)-
taup(1);taumin(2)-taup(2)];
    A(:,2)=[N(1,:);N(3,:)]\[taumin(1)-taup(1);taumin(3)-
taup(3)];
    A(:,3)=[N(1,:);N(4,:)]\[taumin(1)-taup(1);taumin(4)-
taup(4)];
    A(:,4)=[N(2,:);N(3,:)]\[taumin(2)-taup(2);taumin(3)-
taup(3)];
    A(:,5)=[N(2,:);N(4,:)]\[taumin(2)-taup(2);taumin(4)-
taup(4)];
    A(:,6)=[N(3,:);N(4,:)]\[taumin(4)-taup(3);taumin(4)-
taup(4)];
    tn=transpose(N);
    for k=1:6
TAU(:,k)=taup+transpose(transpose(A(:,k))*tn);
    end
   optim(TAU);
  T(i,1)=t;
  cpl1(i,1)=ans(1);
  cpl2(i,1)=ans(2);
  cpl3(i,1)=ans(3);
```

```
 cpl4(i,1)=ans(4);

  figure(2);
    hold on
  plot(t,cpl1(i,1),'-.g');grid
  plot(t,cpl2(i,1),'--k');grid
  plot(t,cpl3(i,1),'-.r');grid
  plot(t,cpl4(i,1),'-.b');grid

end
 couple1=[T cpl1]
 couple2=[T cpl2]
 couple3=[T cpl3]
 couple4=[T cpl4]


function [tauopmin]=optim(TAU)
H=TAU;
TAUop=H;
for z=6:-1:1
for j=1:4
          if   H(j,z)<0.0005-0.00001
             TAUop(:,z)=[]
             break
%            else
%               TAUop(j,z)=0;
          end
end

end

[m,n]=size(TAUop);

s=100;
for f=1:n
    if sum(TAUop(:,f))<s
        s=sum(TAUop(:,f));
        tauopmin=TAUop(:,f)
    %else tauopmin=TAUop(:,f)
    end
end
%          figure(1);
%    hold on
%  plot(t,tauopmin(1),'-.g');grid
%  plot(t,tauopmin(2),'--k');grid
%  plot(t,tauopmin(3),'-.r');grid
%  plot(t,tauopmin(4),'-.b');grid
```