REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE BLIDA 1

FACULTE DE TECHNOLOGIE

DEPARTEMENT DE MECANIQUE

Mémoire de fin d'études

En vue de l'obtention du Diplôme de
Master en Génie Mécanique

Thème :

Conception et réalisation d'un robot parallèle planaire 3 RRR

Réalisé par:

BOUDISSA SAMIA.

DAOU ARAS.

Encadré par:

Mr. CHETTIBI Taha promoteur.

Mr. LOUNICI Billel Co-promoteur.

Année universitaire 2021/2022

DÉDICACE

Pour mes parents qui m'ont soutenu et encadré jusqu'à ce jour.

Pour toute ma famille

Pour toute personne que j'ai connue lors de mon bref passage à l'université.

Pour vous tous je dédie ce modeste travail

Bien grâce à vous tous, donc un grand merci pour vous.

REMERCIEMENTS

Je remercie DIEU, le tout puissant, de nous avoir donné, le courage, la foi et surtout la patience pour la réalisation de ce travail.

Je remercie également notre promoteur Mr.CHETTIBI Taha enseignant à l'Université Saad Dahleb de Blida, pour le privilège qu'il nous a fait en acceptant de diriger ce travail. Sa gentillesse, sa modestie, sa riche expérience et l'accueil cordial qu'il nous a toujours réservé nous ont inspiré.

Mes remerciements s'adressent à notre co-promoteur Mr. LOUNICI Bilal enseignant à l'Université Saad Dahleb de Blida, pour avoir accepté de diriger ce travail. Son soutien, sa clairvoyance et ses compétences nous ont été d'une aide inestimable.

Je remercie les enseignants du département Mécanique spécialement Mr.ABADA Mourad et Mr.TEMMAR Mustapha qui ont contribué à notre cursus.

Enfin, Je remercie les membres du jury qui ont bien voulu accepter, et ce nonobstant, leur lourdes et exaltantes, responsabilités, d'évaluer ce modeste travail.

Résumé :

Dans le présent travail nous avons introduit des méthodes générale utilisées dans la robotique industriel ; pour la modélisation géométrique et cinématique des robots parallèle, par la suite nous avons appliques ces concepts sur un robot parallèle planaire trois degrés de liberté, développer un programme informatique qui simule ce type de robots.

Summary:

In this work we have introduced general methods used in industrial robotics; for the geometric and kinematic modeling of parallel robots, we then applied these concepts to a planar parallel robot 3RRR, developing a computer program that simulates this type of robot.

ملخص:

في هذا العمل قدمنا الأساليب العامة المستخدمة في الروبوتات الصناعية. من أجل النمذجة الهندسية والحركية للروبوتات المتوازية، قمنا بعد ذلك بتطبيق هذه المفاهيم على روبوت متوازي مستوٍ بثلاث درجات حرة، وقمنا بتطوير برنامج كمبيوتر يحاكي هذا النوع من الروبوتات.

Table des matières

Chapitre III

Chapitre IV

Liste des figures

La liste des symboles

$\boldsymbol{r}$: le rayon de la plate-forme $(\boldsymbol{mm})$

$\boldsymbol{l_i}$, $\boldsymbol{b_i}$: les liaisons pivot passives $(\boldsymbol{mm})$

$\boldsymbol{A_i}$: les coordonnées des articulaires

$\boldsymbol{C_i}$: les coordonnées du la plate-forme

$\boldsymbol{q}$: vecteur des articulaires $(\mathbf{deg})$

$\boldsymbol{x}$: vecteur des coordonnées opérationnelles $(\boldsymbol{mm})$

T : la matrice de transformation spatiale

$\dot{\boldsymbol{x}}$: vecteur vitesse des coordonnées opérationnelles $\boldsymbol{(mm/s)}$

$\dot{\boldsymbol{q}}$: vecteur de la vitesse des articulaires $\boldsymbol{(deg/s)}$

$\boldsymbol{K}$: vecteur unitaire

$\boldsymbol{J_D}$: matrice jacobienne

$\ddot{\boldsymbol{x}}$: vecteur accélération des coordonnées opérationnelles $\boldsymbol{(mm/s^2)}$

$\ddot{\boldsymbol{q}}$: vecteur accélération des articulaires $\boldsymbol{(deg/s^2)}$

$\boldsymbol{j}$: la drivés du la matrice jacobienne

Introduction générale

La robotique est une science d'ingénierie pluridisciplinaire qui traite de la conception, la modélisation, la commande et l'utilisation des robots. De nos jours, les machines en générale, et les robots en particulier, accompagnent les Hommes dans la vie quotidienne et les assistent dans l'achèvement de diverses fonctions. En effet, le robot est devenu aujourd'hui l'outil de travail par excellence dans différents secteurs qui touchent un large spectre d'activités humaines, allant de l'industrie manufacturière, passant par l'agriculture, le nucléaire, la médecine, jusqu'à la conquête de l'espace. Cette omniprésence du robot peut être expliquée par sa versatilité, sa dextérité et sa précision dans l'exécution de différentes tâches. Ces caractéristiques, très souvent étonnantes, sont le fruit du progrès enregistré dans plusieurs disciplines à savoir la mécanique, l'électronique, l'informatique, et l'automatique. En effet, la commande et le contrôle de telles machines par le biais de systèmes informatisés les rendent flexibles et versatiles dans leur fonctionnement et exploitation.

Les performances et les capacités d'un robot sont étroitement liées à ses fonctions de perception, de décision et d'action. La motorisation est assurée par des actionneurs, en général électriques, qui transmettent leurs mouvements aux liaisons par des mécanismes appropriés. La perception, qui permet de gérer les relations entre le robot et son environnement, est assurée par des organes de perceptions mesurant l'état interne du robot et recueillant des informations sur l'environnement. La commande synthétise les consignes d'asservissement pilotant les actionneurs à partir de la fonction de perception et des ordres de l'utilisateur.

Dans ce mémoire de master, nous nous intéressons à la conception et à la réalisation d'un robot parallèle plan à trois degrés de liberté. Pour mener à bien la rédaction du manuscrit, nous avons organiser le travail en quatre principaux chapitres. Dans le premier chapitre, nous présentons des généralités sur les robots manipulateurs, leurs constituants et leurs classifications. Le deuxième chapitre sera consacré à la modélisation analytique (géométrique, cinématique et dynamique) de notre robot. Cette modélisation sera accompagnée par une implémentation sous Matlab pour simuler le comportement du robot. Dans le troisième chapitre, nous présenterons la conception d'un modèle de ce robot sous SolidWorks ainsi que les résultats de simulation à l'aide de Motion. Le quatrième chapitre sera consacré à la réalisation notre robot ainsi que sa commande autour d'une carte Arduino. Nous terminerons ce mémoire par une conclusion et quelques perspectives.

Chapitre I

Généralités sur les robots

I.1) Introduction

Un robot industriel est un manipulateur automatique asservi en position, polyvalent, reprogrammable, capable de positionner et d'orienter des pièces ou des outils au cours de mouvements variables et programmés pour l'exécution de tâches variées. Il est composé de plusieurs segments articulés permettant de piloter un organe terminal (effecteur) avec lequel il peut interagir avec son environnement. La position et l'orientation de cet élément sont contrôlées et repérées par leurs coordonnées généralisées qui sont les coordonnées d'un point particulier de l'organe terminal exprimées dans le repère de base.

Le nombre de degrés de liberté (ddl) d'un mécanisme est le nombre de coordonnées généralisées indépendantes nécessaires pour définir la configuration d'un mécanisme. Il décrit aussi le nombre de mouvements indépendants que peut réaliser l'organe terminal d'un mécanisme. Ces mouvements sont décrits au maximum par trois translations et trois rotations selon des axes particuliers. L'analyse de l'agencement des différents segments articulés d'un robot permet de dégager deux grandes familles : les robots sériels et les robots parallèles.

La majorité des robots manipulateurs existants à l'heure actuelle présentent un caractère anthropomorphique marqué avec une forte ressemblance à un bras humain. Il s'agit d'une succession de solides, chacun étant relié à son prédécesseur et à son successeur par une articulation à un degré de liberté motorisée. Ce type d'architecture est qualifié de robot sériel **[1].** Un robot parallèle contrôle le mouvement de son effecteur au moyen d'au moins deux chaînes cinématiques allant de l'effecteur à la base. De plus, un manipulateur pleinement parallèle est un manipulateur parallèle dont le nombre de chaînes est strictement égal au nombre de ddl de l'organe terminal. Dans ce mémoire de master, nous nous intéressons à la conception et à la réalisation d'un robot plan, pleinement parallèle, à trois degrés de liberté.

I.2) Historique des robots parallèles

Les mécanismes à architecture sont connus depuis très longtemps, Léonard de Vinci en ayant déjà proposé au XVIème siècle. En outre, Merlet **[1]** rapporte que la théorie sur les mécanismes parallèles a été explorée bien avant l'apparition du terme robot et que certains problèmes théoriques concernant les robots parallèles ont donc été résolus bien avant leur apparition. Par la suite, l'histoire des robots parallèles a été marquée par plusieurs réalisations de référence notamment **[2]** :

- **1931 :** Le premier brevet déposé pour un mécanisme parallèle porte sur un mécanisme sphérique destiné à être utilisé comme plate-forme de cinéma dynamique annoncé en 1928 **(Figure. 1.1).** Cependant, cette machine n'a jamais été construite.

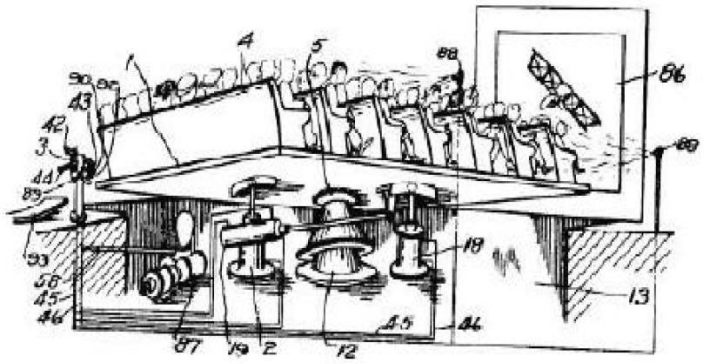


Figure 1.1 : Premier robot parallèle sphérique, annoncé en 1928 et breveté en 1931

- **1942**, Williard L.V. Pollard conçoit ce qui est considéré comme le premier robot parallèle industriel. Son fils fait breveter en 1942 cette invention qui n'a, elle non plus, jamais été produite **(Figure 1.2).**

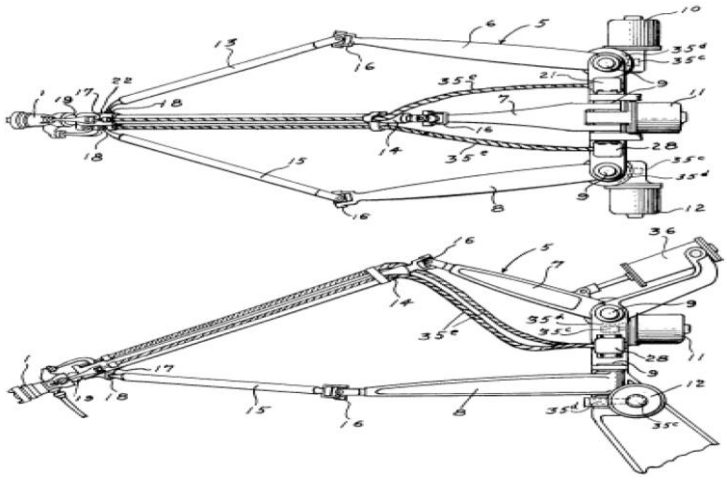


Figure1.2 : La plate-forme de Gough-Stewart pour tester des pneus

- **1947**, le robot parallèle le plus connu et le plus répliqué a été inventé, l'*hexapode octaédrique*, ou ce qu'on appelle injustement la *plate-forme de Stewar* (**Figure. 1.3a**). C'est l'écossais Dr. Eric Gough, ingénieur à Dunlop Rubber Co., en Angleterre, qui a construit le premier l'hexapode octaédrique afin de tester des pneus en appliquant des charges. Par la suite, plusieurs modèles ont été construits autour de cette architecture surtout pour les simulateurs de vol (**Figure 1.3b**).

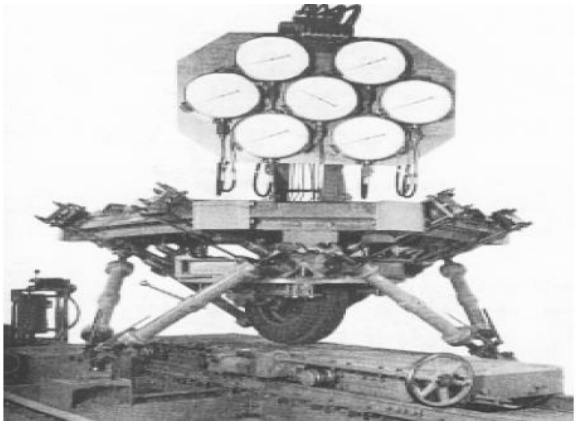


Figure 1.3a : Premier robot parallèle spatial industriel, breveté par Pollard en 1942

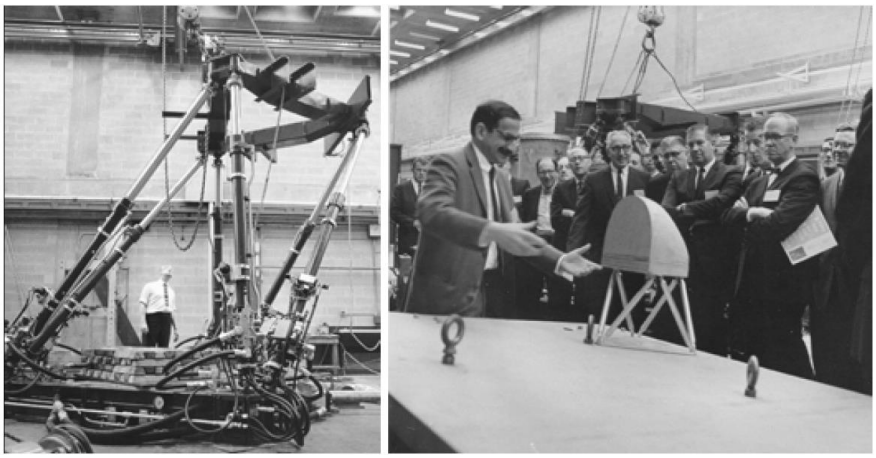


Figure 1.3 b : La plate-forme de Gough-Stewart pour la simulation de vol aérien construit autour de 1965.

- **1980** : le Pr. Reymond Clavel de l'École Polytechnique Fédérale de Lausanne invente et fait breveter l'idée d'utiliser trois parallélogrammes pour construire un robot dont la nacelle ne se déplace que selon trois translations. Ce robot, qui est à ce jour le robot parallèle le plus vendu dans le monde, marque l'arrivée des robots parallèles rapides destinés, entre autres, à des tâches de conditionnement. La **figure 1.4** illustre un modèle commercialisé par la société ABB.

Figure1.4 : FLEXPICKER modèle industriel de la société ABB du robot delta.

Au cours de ces trois dernières décennies, on a assisté à la mise en œuvre d'un grand nombre d'architectures de robots parallèles dont les applications possibles sont très nombreuses :

- Plate-forme de mouvement pour simulateurs,
- Robots manipulateurs pour processus industriels robot de prise et dépose
- Machines-outils,
- Robot médical : manipulateurs en microchirurgie,
- Positionneur de haute précision, etc…

Cet engouement vers ce type de robots s'explique par les avantages apportés par une structure parallèle, ces propriétés sont expliquées dans le paragraphe suivant.

I.3) Propriétés d'un robot parallèle

Un robot parallèle est un mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes, cette architecture lui confère des propriétés remarquables. En effet, un robot parallèle contrôle le mouvement de son effecteur au moyen d'au moins deux chaines cinématiques allant de l'effecteur à la base. Un manipulateur pleinement parallèle est un manipulateur parallèle dont le nombre de chaines est strictement égal au nombre de degrés de liberté de l'organe terminal

La mise en parallèle de plusieurs chaines cinématiques, chacune ne comportant qu'un actionneur, conduit généralement aux avantages suivants :

- Possibilité de positionner les actionneurs directement sur la base fixe ou très proche de celle-ci ; cette particularité a les conséquences positives suivantes :
  - – grand choix de moteurs et de réducteurs puisque leur masse n'a pas d'influence sur l'inertie du robot en mouvement ;
  - – simplification importante des problèmes de liaisons entre les moteurs, les capteurs et le contrôleur (câblage plus simple et plus fiable) ;
  - – facilité de refroidissement des actionneurs, donc diminution des problèmes de précision dus aux dilatations, et puissance potentielle élevée ;

  - facilité d'isolation des moteurs de l'espace de travail pour des activités en atmosphère propre, avec risque de déflagration, ou encore pour des applications en milieux humides.
- Faible masse mobile mais possibilité d'embarquer des charges très lourdes (par exemple pour les simulateurs de vol) ;
- Possibilité de mouvements à haute dynamique avec accélérations élevées : les robots les plus rapides du monde sont des robots parallèles
- Plus grande rigidité puisque la charge est reprise par l'ensemble des jambes et plus d'éléments ne sont soumis qu'à des sollicitations de traction/compression ; des machines d'usinage utilisant ce concept
- Construction mécanique modulaire, simplicité de fabrication et utilisation d'éléments standards par la présence de plusieurs composants identiques sur un robot. À noter aussi des possibilités d'actionnement très variées : au lieu de jambes rigides, on peut parfaitement utiliser des câbles qui ont permis de réaliser des grues avec une bonne précision de positionnement et des supports de maquettes pour les souffleries.
- Facilité d'intégration de capteurs ;
- Haute précision (la géométrie de la mécanique rend moins sensibles les mouvements de la plate-forme aux erreurs de mesure des capteurs qui en permettent le contrôle)
- Fonctionnement à des échelles de taille allant du très grand (les simulateurs de vol, certaines machines de parc d'attractions) au très petit (des micro-robots utilisable en microchirurgie)

Leurs inconvénients majeurs sont un espace de travail relativement réduit et une plus grande complexité de commande en raison de leur comportement hautement non linéaire.

I.4) Architecture des robots parallèles :

Pratiquement, on distingue deux classes des robots parallèles, les robots planaires et les robots spatiaux :

I.4.1) Les robots parallèles planaires

Un robot planaire est constitué généralement d'une plate-forme, comportant trois degrés de liberté deux translations et une rotation autour de la normale au plan de la plateforme **(figure.1.5).** Nous recherchons une structure de robot pleinement parallèle, donc possédant trois chaines cinématiques indépendantes motorisées par trois actionneurs. Chacune de ces chaînes devant être liées à la fois

au sol et à la plate-forme mobile, nous aurons donc trois points d'attache au sol et sur la plate-forme mobile. On peut donc considérer sans perte de généralités une plate-forme triangulaire **[3].**

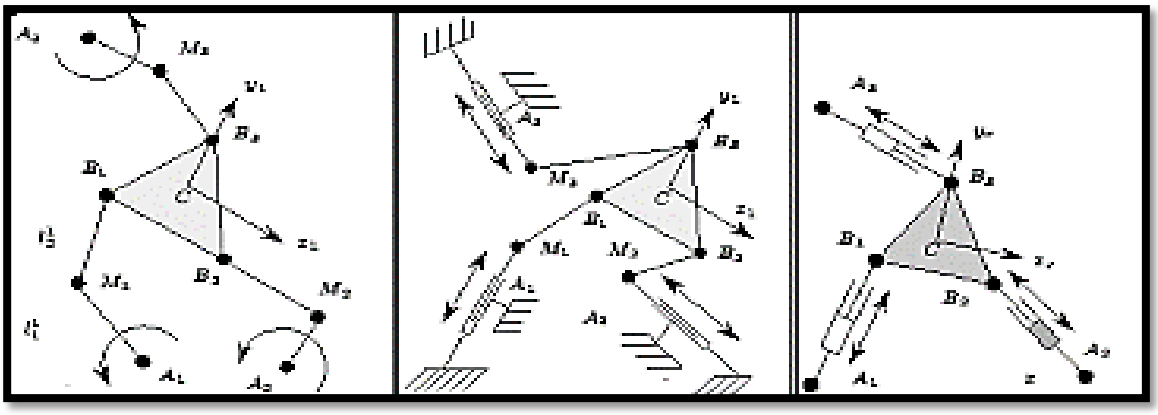


Figure 1.5 : Exemples d'architectures des robots planners.

I.4.2) Les robots parallèles spatiaux

Dans les robots parallèles spatiaux, la plate-forme et la base sont reliées entre elles par des chaines cinématiques assurant trois ou six degrés de liberté à la plate-forme. Le plus connu des robots à trois degrés de liberté en translation est le Delta **[5]** Et le plus connu pour les robots parallèles à six degrés de liberté est la plate-forme de Gough- Stewart ou plate-forme de Stewart. Merlet décrit trois concepts de structures à six degrés de liberté avec base et une plate-forme reliée entre elles par six actionneurs linéaires **(Figure. 1.6)**

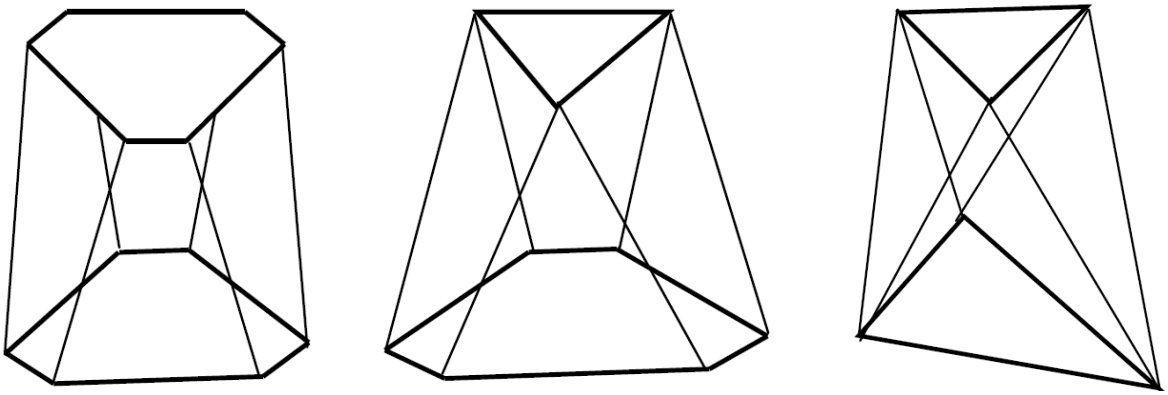


Figure1.6 : Robots parallèles de type SSM, TSSM et MSSM

I.5) Problématiques liées à l'exploitation des robots parallèle

Il s'agit d'une description sommaire qui permet de comprendre les problématiques de base liées à l'exploitation effective des robots parallèles. Les différents types de singularités sont expliqués ainsi que la notion d'espace de travail.

I.5.1) Singularités

Les singularités sont des configurations pour lesquelles le comportement du robot change soudainement. Les chercheurs ont proposé différentes classifications des types de singularités. Nous allons plutôt utiliser la classification la plus simple, notamment :

- Singularités de **type 1** ou singularités **sérielles** ;
- Singularités de **type 2** ou singularités **parallèles**.

I.5.1.1) Singularités de type 1 :

Les singularités de type 1 sont des configurations pour lesquelles l'effecteur terminal du robot perd un ou plusieurs degrés de liberté. Elles correspondent aux frontières de l'espace de travail, c'est-à-dire l'endroit où un des bras atteint sa limite intérieure ou extérieure. La **figure1.7** montre le bras gauche d'un robot parallèle à cinq barres (comme celui de **la figure 1.7**) en singularité de type 1

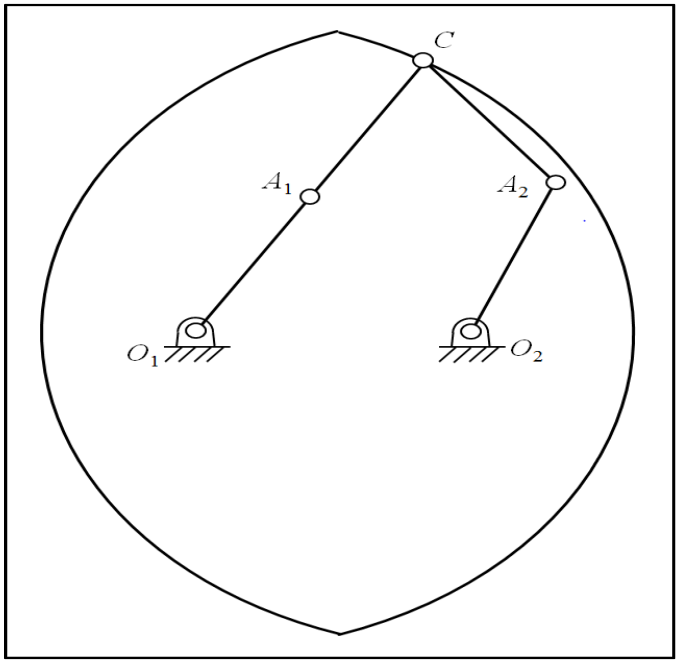


Figure 1.7: Un exemple de singularité de type1

I.5.1.2) Singularités de type 2

Les singularités de type 2 sont des configurations plus complexes où le robot perd sa raideur et gagne un ou plusieurs degrés de liberté. L'effecteur terminal devient alors mobile même si les actionneurs sont bloqués. Les actionneurs, quant à eux, ne peuvent résister à une force ou à un moment appliqué à l'effecteur terminal et le robot n'est plus contrôlable.

Les singularités de type 2 se trouvent à l'intérieur de l'espace de travail, le séparant ainsi en plusieurs zones. Le passage d'une zone à une autre peut se faire en traversant une singularité de type 2. La **figure 1.8** montre deux configurations d'un robot en singularités de type 2. À gauche, les liens A_1C et A_2C sont superposés tandis qu'à droite, ils sont alignés. Dans le casde la superposition, l'ensemble A_1CA_2 n'est pas contraint et peut tourner librement autour du point A_1. Dans le cas de l'alignement, le robot est incapable de résister à une force normale à la droite A_1A_2.

De plus, le robot ne peut pas être dégagé de ses configurations singulières de type 2 par les actionneurs et le mécanisme est complètement bloqué. Pour dégager le robot, une force externe (telle que la gravité) doit donc être utilisée.

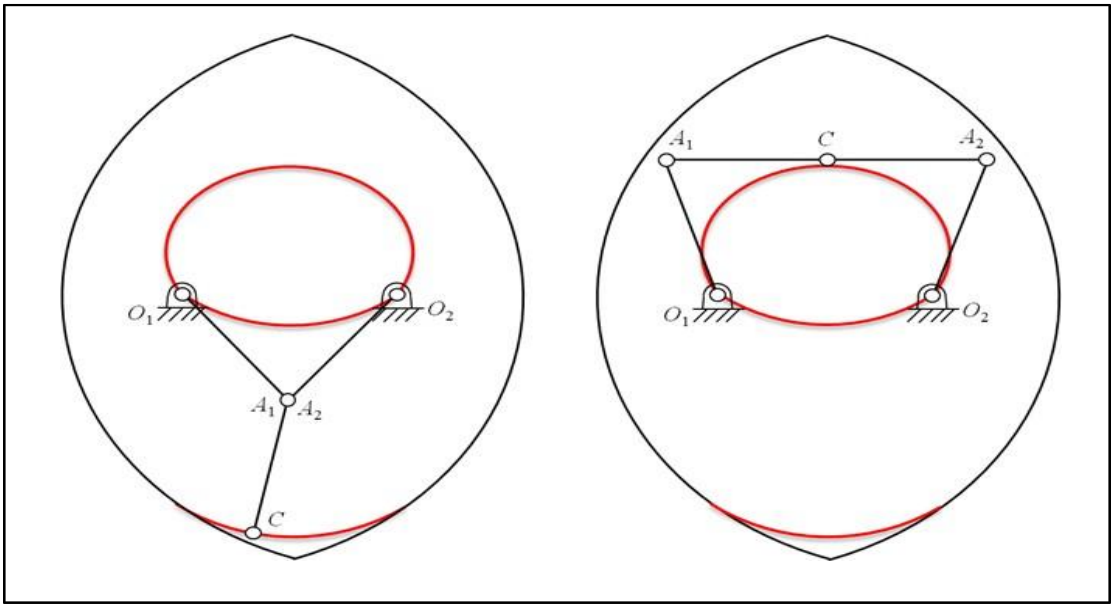


Figure1.8 : Singularités de type 2

À cause de la nature des singularités de type 2, on limite généralement les déplacements de l'effecteur dans une des zones exemptes de ces singularités. L'espace de travail est alors considérablement réduit. Si on cherche à traverser ces singularités, le robot ne pourra pas exécuter un mouvement continu et précis. Pour des opérations de « Pick-and-place », la trajectoire n'a pas besoin d'être précise et il est théoriquement possible de passer les singularités de type 2. Cependant, le contrôle du robot est momentanément perdu au passage de la singularité, ce qui peut ponctuellement entraîner un comportement imprécis. Il est aussi impossible de prendre ou de déposer un objet se trouvant proche d'une courbe de singularité de type2 **[6]**

I.5.2) Espace de travail

L'optimisation de l'espace de travail d'un robot parallèle est un enjeu important. La méthode souvent privilégiée est d'éviter des mouvements où le robot passe par une singularité detype 2 en réduisant l'espace de travail. Ainsi, le contrôle du robot est assuré dans toutl'espace réduit et le risque d'endommager le mécanisme est minimisé. Cependant, cet espace doit être encore réduit pour éviter les interférences mécaniques qui peuvent être relativement complexes pour les robots parallèles. L'espace de travail réel est donc considérable réduit comparé à celui d'un robot sériel. Afin d'utiliser l'espace de travail d'un robot parallèle à son plein potentiel, de façon à ce qu'il soit comparable à celui d'un robot sériel, il est possible d'éviter la réduction de l'espace en utilisant des méthodes qui permettent aurobot de passer d'une zone à une autre de l'espace grâce au passage par des singularités de type 1 réduit comparé à celui d'un robot sériel. Afin

d'utiliser l'espace de travail d'un robot parallèle à son plein potentiel, de façon à ce qu'il soit comparable à celui d'un robot sériel, il est possible d'éviter la réduction de l'espace en utilisant des méthodes qui permettent aurobot de passer d'une zone à une autre de l'espace grâce au passage par des singularités de type 1. **[6]**

I.6) Types de mouvements

De façon générale, il existe principalement trois types de mouvement pour accomplir une tâche industrielle :

- Mouvements suivant un chemin prédéfini. Ce sont des mouvements où l'effecteur terminal du robot doit suivre un chemin soumis à des contraintes (par exemple, lesoudage en continu ou le découpage). La trajectoire sera déterminée en tenant comptedes obstacles à éviter et de la tâche à exécuter. Le robot est constamment en interaction avec l'environnement.
- Mouvements point-à-point prédéfinis (« Pick and place »). Dans des opérations de transfert rapide, ce type de mouvement consiste à prendre un objet à un endroit prédéfini et à le déplacer à un autre endroit prédéfini sans se soucier du chemin si ce n'est quepour éviter des obstacles prédéfinis qui pourraient être dans l'espace de travail.
- Mouvements couplés à un système de vision. Ces mouvements sont utilisés lorsque les pièces arrivent sur un convoyeur de façon aléatoire. Un système de vision repère les objets et indique leur position au robot. Celui-ci doit saisir l'objet et le déposer à un autreendroit (souvent prédéfini)

Les deux premiers types de mouvements sont prédéfinis. Le chemin peut donc être calculé au préalable à l'aide d'un algorithme hors ligne. Ceci est très intéressant, car le temps de calcul n'est pas limité de sorte que des méthodes d'optimisation, souvent gourmandes en calculs, mais très puissantes, peuvent être utilisées. La troisième méthode doit utiliser des algorithmes qui ne donneront pas nécessairement des solutions optimales, faute de temps de calculsuffisamment long, mais donneront des solutions acceptables en temps réel.

Généralement, les trajectoires comportant moins de contraintes offrent plus de possibilités.La planification est alors un plus grand défi, car il y davantage place à l'optimisation. En déterminant le chemin optimal entre deux points, il est possible d'optimiser, selon plusieurs critères comme le temps de parcours, la sollicitation des actionneurs et la fluidité du mouvement. Pour un chemin prédéfini, il est possible d'optimiser le temps et la fluidité, maisles contraintes sont généralement fixées par la tâche, c'est-à-dire, par exemple, que la vitesse est contrainte par le temps nécessaire pour l'application d'un produit. **[6].**

Lors du processus de planification des trajectoires, les contraintes liées aux performances maximales du robot doivent être considérées. Aussi les modèles de comportement adaptés doivent être élaborés.

En général, il existe trois types de modélisations :

- Modélisation géométrique ;
- Modélisations cinématique ;
- Modélisation dynamique.

La modélisation cinématique se limite à contraindre la trajectoire à ne pas dépasser l'accélération maximale et la vitesse maximale. La vitesse maximale peut être calculée à partir des spécifications des actionneurs et du ratio de transmission alors que l'accélération maximale peut être approximée par le ratio du couple maximal des actionneurs par rapport à l'inertie maximale des articulations. Ces approximations engendrent souvent des bornes maximales inférieures à la réalité. Généralement, on utilise la modélisation cinématique dans des tâches où il faut éviter des obstacles. Pour ce faire, un algorithme pour trouver le chemin le plus court peut être couplé à un algorithme qui détermine le temps minimum de la trajectoire.
Le problème de planification de trajectoire est alors beaucoup plus simple à traiter puisqu'il s'exprime sous forme linéaire.
La modélisation dynamique incorpore au modèle les caractéristiques inertielles comme la masse, le moment d'inertie et le centre de gravité des composantes en mouvement. Les forces générées par les mouvements sont alors transformées en couples qui dépendent de la position, de la vitesse et de l'accélération du robot. Il s'agit donc d'une modélisation beaucoup plus réaliste du comportement du robot qui permet d'obtenir des performances supérieures. On peut utiliser un algorithme pour trouver le chemin le plus court ou utiliser un algorithme qui trouvera le chemin optimal en considérant l'énergie ou la sollicitation des actionneurs par exemple. Un autre algorithme optimisera la trajectoire selon plusieurs critères possibles. Le problème de planification de trajectoire avec un modèle dynamique est plus complexe puisqu'il faut tout d'abord bien définir ce modèle et le valider pour ensuite traiter le problème d'optimisation de trajectoire, qui sera généralement non-linéaire. **[6]**

I.7) Conclusion

Dans ce chapitre, nous avons présenté un bref historique des robots à structures parallèles avec un ordre chronologique de leur développement. Ensuite, on a donné quelques définitions utiles liées à la robotique parallèle, tout en citant quelques exemples de robots parallèles. Par la suite, on a discuté des avantages et inconvénients de ce type de robots. Enfin, nous avons présenté quelques problématiques liées à l'exploitation des robots parallèles. Le prochain chapitre sera consacré à la modélisation analytique d'un robot pleinement parallèle plan à 3ddl avant d'aborder sa réalisation par la suite.

Chapitre II

Modélisation d’un robot parallèle planaire 3RRR

II.1) Introduction

Pour pouvoir concevoir et exploiter un robot, Il est essentiel d'établir les différents modèles physiques régissant son comportement. Dans ce chapitre, nous allons développer les modèles : géométrique et cinématique du robot parallèle plan 3-RRR qu'on désire concevoir et réaliser. Pour être efficace, nous adopterons une méthode matricielle similaire à celle utilisée dans la référence **[7].**

Un modèle tridimensionnel du robot parallèle 3-RRR qu'on désire étudier est présentée à la **(fig. 2.1)**. Le schéma cénématique de ce robot est présenté à la **figure 2.2**. Les trois moteurs actionnant le robot parallèle sont disposés sur les sommets du triangle régulier de la base fixe $A_1\ A_2\ A_3$. La nacelle du robot est une plateforme mobile $C_1\ C_2\ C_3$; elle est connectée à la base fixe par trois branches identiques $A_1\ B_1\ C_1$, $A_2\ B_2\ C_2$, $A_3\ B_3\ C_3$. Comme le montre la **figure 2.2**, les trois sommets du triangle régulier sont A_1, A_2, A_3, où les articulations pivot sont actives et contrôlées. Les points de $B_1, C_1, B_2, C_2, B_3, C_3$, sont des liaisons pivots passives, où $A_1B_1 = A_2B_2 = A_3B_3 = l_i$, $B_1C_1 = B_2C_2 = B_3C_3 = b_i$. La distance des moteurs sur la plate-forme de travail fixe est R. Le rayon du cercle des joints dans la plate-forme mobile est r. Le mouvement planaire de la nacelle est à 3-ddl et peut être réalisé en utilisant les trois moteurs d'entraînement des liaisons actives**[7].**

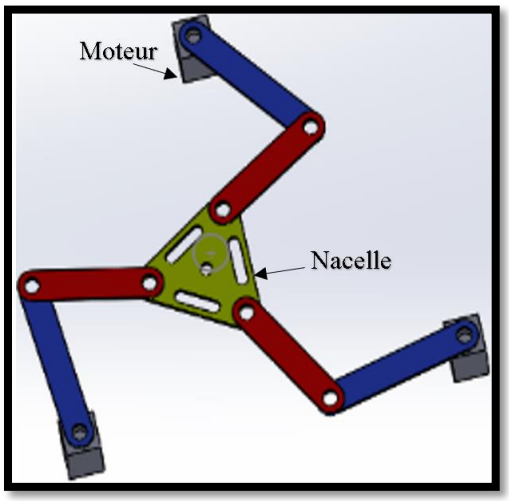


Figure 2. 1: Modèle CAO d'un robot parallèle plan (3-RRR)

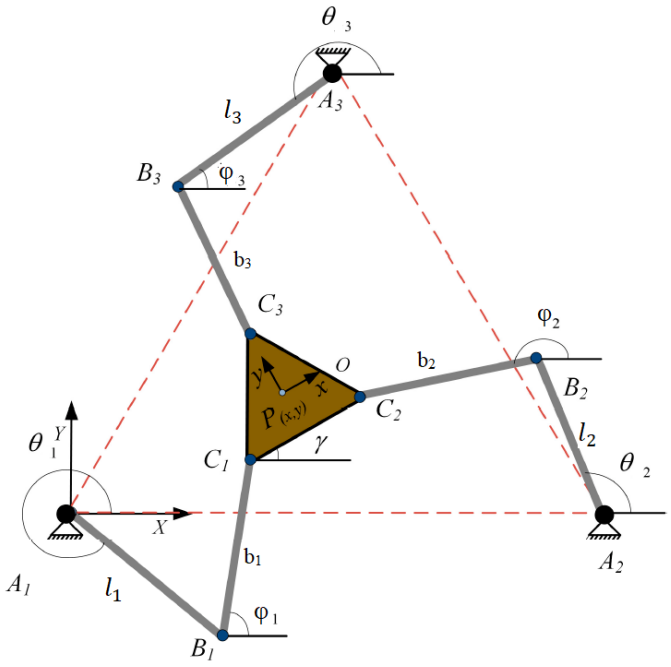


Figure 2. 2:Schéma cinématique du robot parallèle 3-RRR plan étudié.

II.2) La modélisations géométrique

Dans la modélisation géométrique on s'intéresse à la description géométrique du mouvement sans tenir compte des forces qui le provoquent.

II.2.1) Le modèle géométrique direct (MGD) du robot parallèle plan 3-RRR

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires **[8].**

D'une manière générale, Le modèle géométrique direct du robot peut être représenté par la relation :

$$X = f(q) \tag{2.1}$$

q : étant le vecteur des variables articulaires tel que

$$q = [\theta_1, \theta_2, \theta_3]^T$$

x : Les coordonnées opérationnelles sont définies par :

$$x = [x, y, \gamma]^T$$

Selon les notations indiquées sur la **figure 2.2**, les coordonnées du moteur dans le système de coordonnées fixe (A_1, X, Y) sont comme suit :

$$A_i = [A_1; A_2; A_3] \quad \text{avec} \quad A_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} ; A_2 = \begin{pmatrix} R \\ 0 \end{pmatrix} ; A_3 = \begin{pmatrix} \frac{1}{2}R \\ \frac{\sqrt{3}}{2}R \end{pmatrix} \tag{2.2}$$

Les coordonnées des sommets de la plate-forme mobile dans le système de coordonnées mobile sont indiquées comme suit :

$$P = [C_1; C_2; C_3] \quad \text{avec} \quad C_1 = \begin{pmatrix} -\frac{\sqrt{3}}{2}r \\ -\frac{1}{2}r \end{pmatrix} ; C_2 = \begin{pmatrix} \frac{\sqrt{3}}{2}r \\ -\frac{1}{2}r \end{pmatrix} ; C_3 = \begin{pmatrix} 0 \\ r \end{pmatrix} \tag{2.3}$$

La matrice de transformation spatiale **T** pour transformer le système de coordonnées mobile (nacelle) en système de coordonnées fixe (base) est représentée comme suit :

$$T=\begin{vmatrix}\cos\gamma & -\sin\gamma & 0 & x\\ \sin\gamma & \cos\gamma & 0 & y\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{vmatrix} \quad \textbf{(2.5)}$$

Les coordonnées du centre de la plate-forme mobile dans le système de coordonnées fixe peuvent être exprimées comme suit :

$$\binom{C}{1} = T \times \binom{P}{1} \quad \textbf{(2.6)}$$

En combinant les équations **(2.2)** et **(2.6)**, et compte tenue de la géométrie de nacelle de la **figure 2.3**, les coordonnées C= [C_1 ; C_2 ; C_3] des sommets de la plate-forme mobile dans le système de coordonnées fixe sont comme suit :

$$C_1=\begin{pmatrix}x+r\cos(\gamma+\vartheta_1)\\ y+r\sin(\gamma+\vartheta_1)\end{pmatrix} ; C_2=\begin{pmatrix}x+r\cos(\gamma+\vartheta_2)\\ y+r\sin(\gamma+\vartheta_2)\end{pmatrix} ; \ C_3=\begin{pmatrix}x+r\cos(\gamma+\vartheta_3)\\ y+r\sin(\gamma+\vartheta_3)\end{pmatrix} \quad \textbf{(2.7)}$$

$$\vartheta_1 = \frac{7\pi}{6}$$

$$\vartheta_2 = -\frac{\pi}{6}$$

$$\vartheta_3 = \frac{\pi}{2}$$

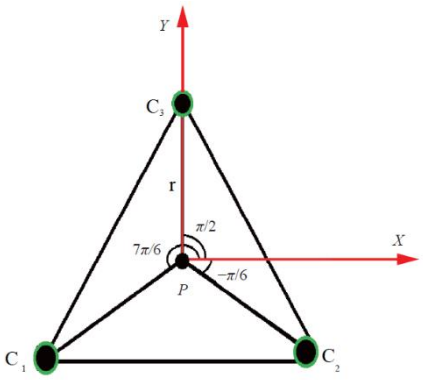


Figure 2. 3: Géométrie de la nacelle du robot en 2D

La **figure 2.4** est une capture du programme Matlab qu'on a réalisé pour simuler la géométrie de notre robot.

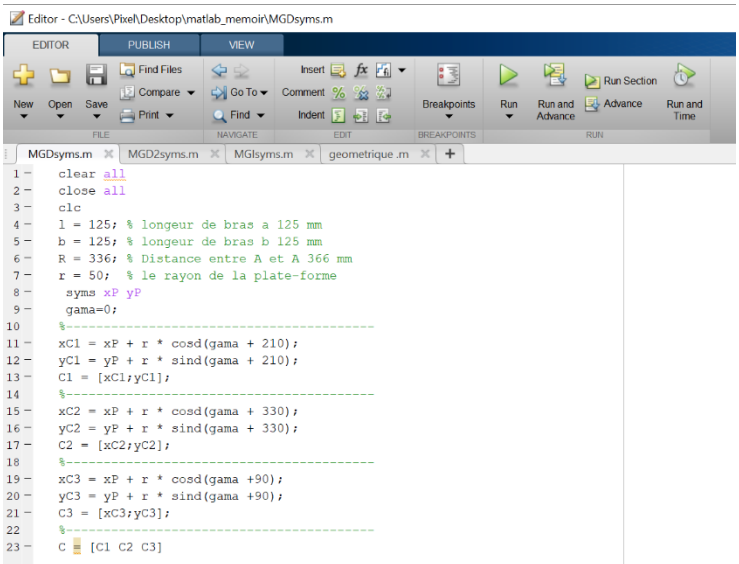


Figure 2. 4: programme Matlab de la modélisation géométrique directe

Pour une branche quelconque A_i B_i C_i, le robot parallèle satisfait à la contrainte **(2.8)** de mouvement suivant:

$$\overrightarrow{OC_i}=\overrightarrow{OA_i}+\overrightarrow{A_iB_i}+\overrightarrow{B_iC_i} \qquad \text{i=1 ,2 et 3} \qquad \textbf{(2.8)}$$

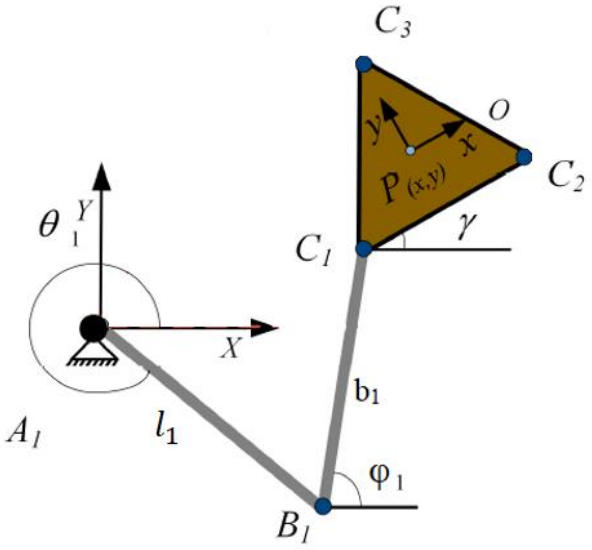


Figure 2. 5: Angles associés à une branche du manipulateur parallèle 3RRR

Combiné à l'équation **(2.7)**, φ_i est l'angle passif entre la liaison $\mathbf{B_iC_i}$ et la direction positive de l'axe X horizontal, alors :

$$\begin{cases} x_{Ci} = x_{Ai} + l_i \cos\theta_i + b_i \cos\varphi_i \\ y_{Ci} = y_{Ai} + l_i \sin\theta_i + b_i \sin\varphi_i \end{cases} \qquad \textbf{(2.10)}$$

Puisque φ est une liaison passive, cet angle doit être éliminé des équations précédentes, Pour pouvoir écrire le MGD de notre robot.

II.2.2) Le modèle géométrique inverse MGI du robot plan 3-RRR

Le modèle géométrique inverse MGI permet de déterminer le vecteur des variables articulaires à partir du vecteur de coordonnées opérationnelles**. [8]**

$$q = g(x) \qquad \textbf{(2.11)}$$

L'équation **(2.10)** peut être exprimée comme suit :

$$\begin{cases} x_{Ci}^2 + x_{Ai}^2 - 2x_{Ci}x_{Ai} = l_i^2 \cos^2\theta_i + b_i^2 \cos^2\varphi_i + 2l_i\, b_i \cos\theta_i \cos\varphi_i \\ y_{Ci}^2 + y_{Ai}^2 - 2y_{Ci}y_{Ai} = l_i^2 \sin^2\theta_i + b_i^2 \sin^2\varphi_i + 2\, l_i b_i \sin\theta_i \sin\varphi_i \end{cases} \qquad \textbf{(2.12)}$$

D'après l'Eq **(2.12)** on a par sommation des deux côtés:

$$x_{Ci}^2 + x_{Ai}^2 + y_{Ci}^2 + y_{Ai}^2 - 2x_{Ci}x_{Ai} - 2y_{Ci}y_{Ai} = l_i^2 + b_i^2 + 2l_i b_i \cos(\theta_i - \varphi_i) \qquad \textbf{(2.13)}$$

Et l'équation **(2.13)** peut être exprimée comme suit :

$$e_{i1} \sin\theta_i + e_{i2} \cos\theta_i + e_{i3} = 0 \qquad \textbf{(2.14)}$$

soient **:** $\qquad e_{i1} = 2l_i(y_{Ai} - y_{Ci}),\ e_{i2} = 2l_i(x_{Ai} - x_{Ci})$

et $e_{i3} = {x_{Ci}}^2 + {y_{Ci}}^2 + x_{Ai}^2 + y_{Ai}^2 - 2x_{Ci}x_{Ai} - 2y_{Ci}y_{Ai} + {l_1}^2 - {b_1}^2$

Par substitution des identités trigonométriques suivantes

$\sin\theta_i = \frac{2t_i}{1+{t_i}^2}$ Et $\cos\theta_i = \frac{1-{t_i}^2}{1+{t_i}^2}$ avec $t_i = \tan\frac{\theta_i}{2}$ et $\tan\theta = \frac{2t}{1-t^2}$

dans l'éq **(2 .15)**, nous obtenons :

$$(e_3 - e_2){t_1}^2 + 2e_1t_1 + (e_3 + e_2) = 0 \qquad \textbf{(2.15)}$$

La résolution de l'équation **(2.15)** pour t_i nous donne :

$$\begin{cases} \theta_{i1} = 2\tan^{-1}\frac{-e_1+\sqrt{{e_1}^2+{e_2}^2-{e_3}^2}}{e_3-e_2} \\ \theta_{i2} = 2\tan^{-1}\frac{-e_1-\sqrt{{e_1}^2+{e_2}^2-{e_3}^2}}{e_3-e_2} \end{cases} \qquad \textbf{(2.16)}$$

La **figure 2.6** illustre le programme Matlab implémentant les relations du MGI développées.

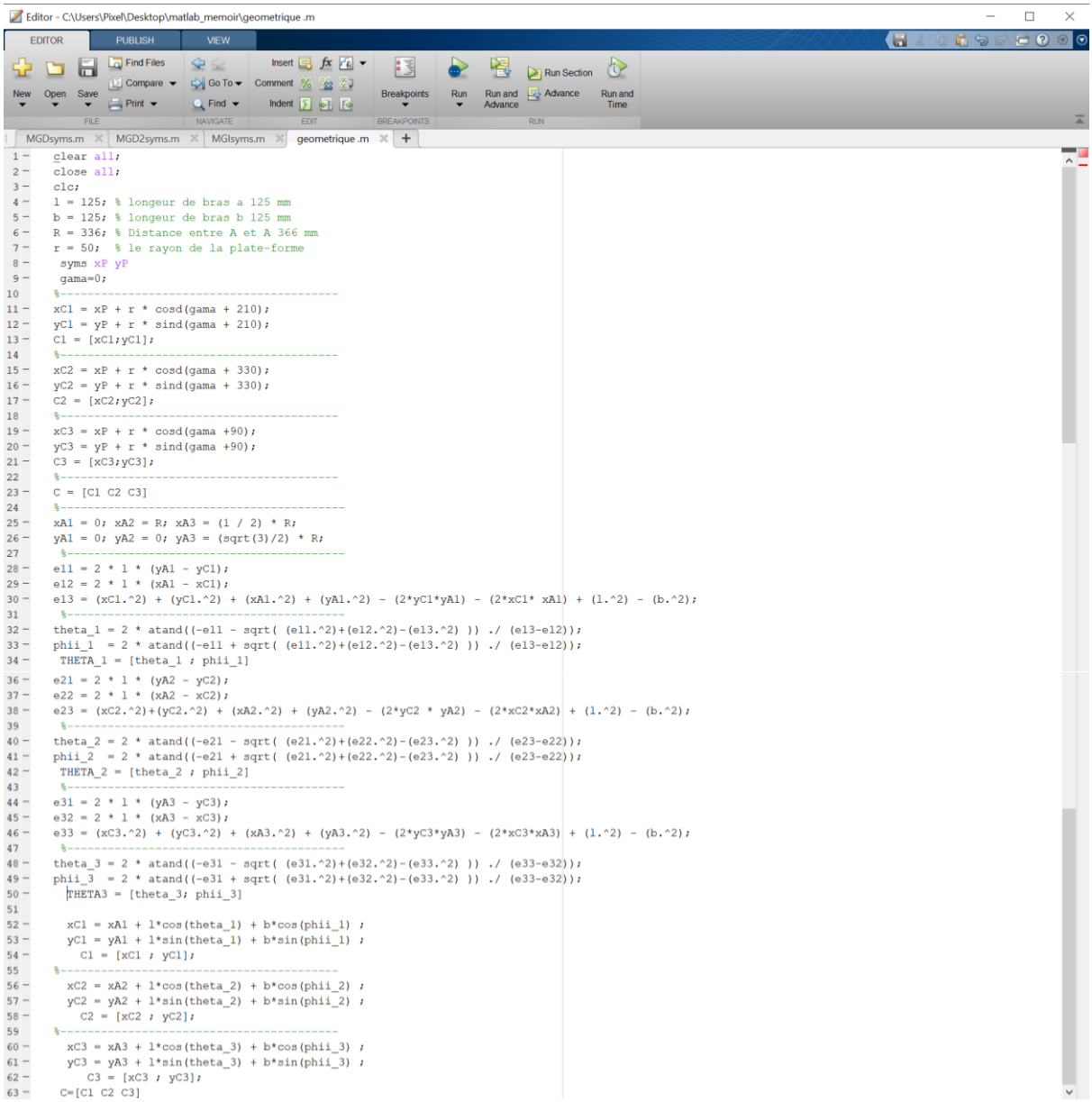


Figure 2. 6:Programme Matlab de la modélisation géométrique inverse

La **Figure 2.7** illustre un programme que nous avons réalisé en exploitant le MGI pour accomplir une trajectoire rectangulaire du point p situe au centre de la plate-forme **(Figure 2.8).**

rectangle.m

```
clear all;close all;clc;
r=50;
gama=0;
l=125;
b=125;
R=340;
xo=[110 210 210 110 110];
yo=[88 88 188 188 88];
o=[xo;yo]
for i=1:length(o)
    Xp=xo(i)
    Yp=yo(i)
    Xa1=0;
    Ya1=0;
    Xa2 = R;
    Ya2 = 0;
    Xa3 = (1 / 2) * R;
    Ya3 = (sqrt(3)/2) * R;
     Xc1=xo(i)+r*cosd(gama+210);
     Yc1=yo(i)+r*sind(gama+210);

     Xc2=Xp+r*cosd(gama+330);
     Yc2=Yp+r*sind(gama+330);

     Xc3=Xp+r*cosd(gama+90);
     Yc3=Yp+r*sind(gama+90);

e21 = 2 * l * (Ya2 - Yc2);
e22 = 2 * l * (Xa2 - Xc2);
e23 = (Xc2.^2)+(Yc2.^2) + (Xa2.^2) + (Ya2.^2) - (2*Xc2*Xa2) -(2*Yc2 * Ya2) + (l.^2) - (b.^2);
 %---------------------------------------
thita21 = 2 * atand((-e21 - sqrt( (e21.^2)+(e22.^2)-(e23.^2) )) ./ (e23-e22))
thita22 = 2 * atand((-e21 + sqrt( (e21.^2)+(e22.^2)-(e23.^2) )) ./ (e23-e22));
 Xb2=Xc2-b*cosd(thita22);
 Yb2=Yc2-b*sind(thita22);

e31 = 2 * l * (Ya3 - Yc3);
e32 = 2 * l * (Xa3 - Xc3);
e33 = (Xc3.^2) + (Yc3.^2) + (Xa3.^2) + (Ya3.^2) - (2*Yc3*Ya3) - (2*Xc3*Xa3) + (l.^2) - (b.^2);
 %---------------------------------------
thita31 = 2 * atand((-e31 - sqrt( (e31.^2)+(e32.^2)-(e33.^2) )) ./ (e33-e32));
thita32 = 2 * atand((-e31 + sqrt( (e31.^2)+(e32.^2)-(e33.^2) )) ./ (e33-e32));

 Xb3=Xc3-b*cosd(thita32);
 Yb3=Yc3-b*sind(thita32);

 e11 = 2 * l * (Ya1 - Yc1);
e12 = 2 * l * (Xa1 - Xc1);
e13 = (Xc1.^2) + (Yc1.^2) + (Xa1.^2) + (Ya1.^2) - (2.*Yc1.*Ya1) - (2.*Xc1.* Xa1) + (l.^2) - (b.^2);
 %---------------------------------------
thita11 = 2 * atand((-e11 - sqrt( (e11.^2)+(e12.^2)-(e13.^2) )) ./ (e13-e12));
thita12 = 2 * atand((-e11 + sqrt( (e11.^2)+(e12.^2)-(e13.^2) )) ./ (e13-e12));

 Xb1=Xc1-b*cosd(thita12);
 Yb1=Yc1-b*sind(thita12);

    plot(xo,yo,[Xc1 Xc2],[Yc1 Yc2],[Xc2 Xc3],[Yc2 Yc3],[Xc3 Xc1],[Yc3 Yc1],...
        [Xc2 Xb2],[Yc2 Yb2],[Xb2 Xa2],[Yb2 Ya2],[Xc3 Xb3],[Yc3 Yb3],...
        [Xb3 Xa3],[Yb3 Ya3],[Xc1 Xb1],[Yc1 Yb1],[Xb1 Xa1],[Yb1 Ya1],'linewidth',2)

    grid on;
    axis equal;
    axis([-100 500 -100 400]);
    drawnow;
    pause (0.5)
end
```

Figure 2. 7: Programme Matlab d'une trajectoire rectangulaire

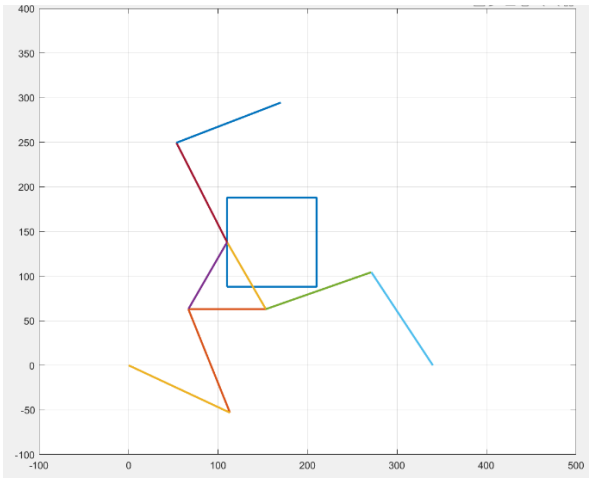


Figure 2. 8: Un chemin rectangulaire généré sous Matlab en utilisant le MGI

II.2.2.1) les solutions possibles du MGI du robot

Pour la même posture de la nacelle, le MGI dispose de 8 solutions possibles donnant lieu à huit modes de fonctionnement différents du manipulateur à l'étude **(Figure 2.9)**, que nous appelons maintenant (a), (b), ..., (h).

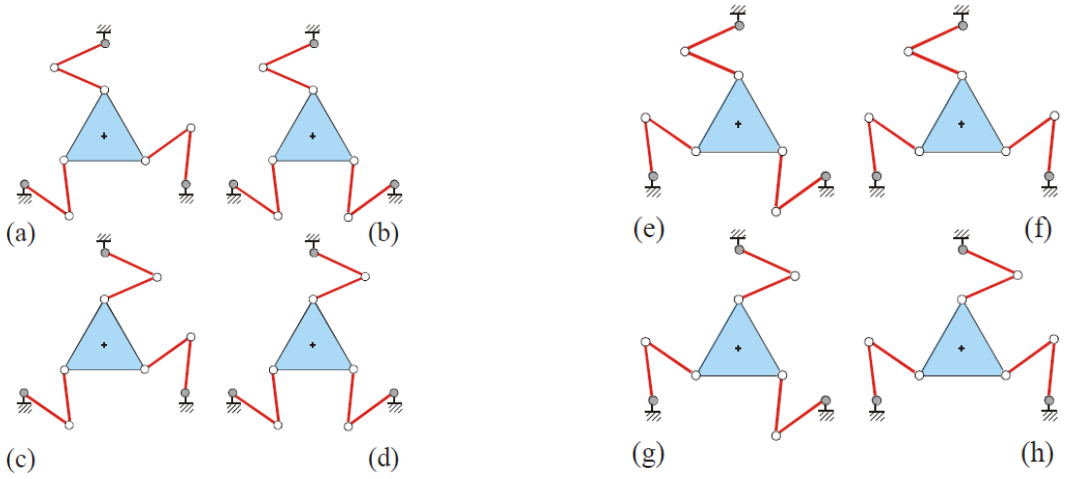


Figure 2. 9: Les huit positions des bras du robot

La capacité d'un manipulateur parallèle à modifier sa solution cinématique inverse dépend des limites des joints passifs et actionnés. Ce problème n'est pas pris en compte dans notre étude puisque des articulations sont supposées illimitées **[9].**

II.3) Modélisation cinématique

Nous examinons d'abord les matrices jacobiennes puis les positions singulières du manipulateur planaire 3RRR décrit précédemment. On se basera sur le schéma du manipulateur montré à la **figure 2.1**. La plate-forme mobile est définie par les pivots mobiles C_1, C_2 et C_3, et la plate-forme fixe est définie par les pivots fixes A_1, A_2 et A_3. Le point P est un point dans la plate-forme mobile, l_i, et b_i Indique les vecteurs des premier et deuxième maillons mobiles de la i -ème branche, tandis que $\boldsymbol{\theta_i}$ et $\boldsymbol{\varphi_i}$ indique les angles d'orientation de $\mathbf{l_i}$ et $\mathbf{b_i}$, par rapport à l'axe des x. Pour ce manipulateur, le vecteur d'entrée est $q = [\theta_1, \theta_2, \theta_3]^T$ et le vecteur de sortie est $x = [x_P, y_P, \gamma]^T$. pour une boucle cinématique, on a une équation de fermeture qui peut être écrite pour chaque membre. Par exemple, pour le premier membre nous avons **[10] :**

$$\overrightarrow{AP} + \overrightarrow{PC_1} = \overrightarrow{AB_1} + \overrightarrow{B_1C_1} \qquad \textbf{(2.17)}$$

II.3.1) Le modèle cinématique directe du robot parallèle plan 3RRR-(MCD)

Comme il s'agit d'un mécanisme planaire, les vecteurs de vitesse angulaire de tous les maillons pointent dans la direction z. Un vecteur de vitesse pour l'équation de boucle est obtenu en prenant la dérivée d'Eq. **(2.17)** par rapport au temps **[10]**, alors on obtient :

$$\boldsymbol{v_P} + \dot{\gamma}(\boldsymbol{K} \times r_i) = \dot{\theta}_i(\boldsymbol{K} \times l_i) + \dot{\varphi}_i(\boldsymbol{K} \times b_i) \qquad \text{i=1 ;2 ;3,} \qquad \textbf{(2.18)}$$

Où v_P est la vitesse du point P et k est un vecteur unitaire pointant dans la direction positive de l'axe z. Comme il s'agit d'une variable φ_i passive, elle devrait être éliminée de l'équation **(2.18)**. Pour atteindre cet objectif, nous multiplions les deux côtés de l'équation **(2.18)** par b_i. Cela mène à **[10] :**

$$b_i . v_P + \dot{\gamma} K(r_i \times b_i) = \dot{\theta}_i K(l_i \times b_i) \qquad \textbf{(2.19)}$$

Écrire l'équation. **(2.19)** trois fois, pour i = 1, 2, et 3, nous donne trois équations scalaires, qui peuvent être disposées sous forme de matrice :

$$J_x \dot{x} = J_q \dot{q} \qquad \textbf{(2.20)}$$

$$\text{Avec } J_x = \begin{bmatrix} b_{1x} & b_{1y} & r_{1x}b_{1y} - r_{1y}b_{1x} \\ b_{2x} & b_{2y} & r_{2x}b_{2y} - r_{2y}b_{2x} \\ b_{3x} & b_{3y} & r_{3x}b_{3y} - r_{3y}b_{3x} \end{bmatrix}$$

$$J_q = \begin{bmatrix} l_{1x}b_{1x} - l_{1y}b_{1y} & 0 & 0 \\ 0 & l_{2x}b_{2x} - l_{2y}b_{2y} & 0 \\ 0 & 0 & l_{3x}b_{3x} - l_{3y}b_{3y} \end{bmatrix}$$

$$\text{Où} \quad \dot{x} = [v_{Px}, v_{Py}, \dot{\gamma}]^T \quad \text{et} \quad \dot{q} = [\dot{\theta_1}; \dot{\theta}_2; \dot{\theta}_3]^T.$$

Les deux matrices J_x et J_q dans l'équation **(2.20)** ce sont deux matrices jacobiennes distinctes. Ainsi, l'ensemble de La matrice jacobienne J_D peut être obtenue comme suite

$$J_D = J_x^{-1} J_q \qquad \textbf{(2.21)}$$

La **figure 2.10** montre un programme utilisant les matrices jacobiennes du robot traité. Et la **figure 2.11** illustre le modèle cinématique direct développé.

Le jacobien du manipulateur est utilisé en fait pour transférer les vitesses de l'espace opérationnelles à l'espace cartésien, i.e. :

$$\dot{x} = J_D \dot{q} \qquad \textbf{(2.22)}$$

JACD.m

```
function J=JACD(lx1,bx1,rx1,ly1,by1,ry1,lx2,bx2,rx2,ly2,by2,ry2,lx3,bx3,rx3,ly3,by3,ry3)

Jx=[bx1 by1 ((rx1.*by1)-(ry1.*bx1)) ; bx2 by2 ((rx2.*by2)-(ry2.*bx2)) ; bx3 by3 ((rx3.*by3)-(ry3.*bx3))];
Jq=[((lx1.*bx1)-(ly1.*by1)) 0 0 ; 0 ((lx2.*bx2)-(ly2.*by2)) 0 ; 0 0 ((lx3.*bx3)-(ly3.*by3))];
if det(Jx)==0
    disp('configuration singuliere')
    J=[0 0 0;0 0 0;0 0 0];
else
    J=inv(Jx)*Jq
end
end
```

Figure 2. 10: programme Matlab de la matrice jacobienne directe

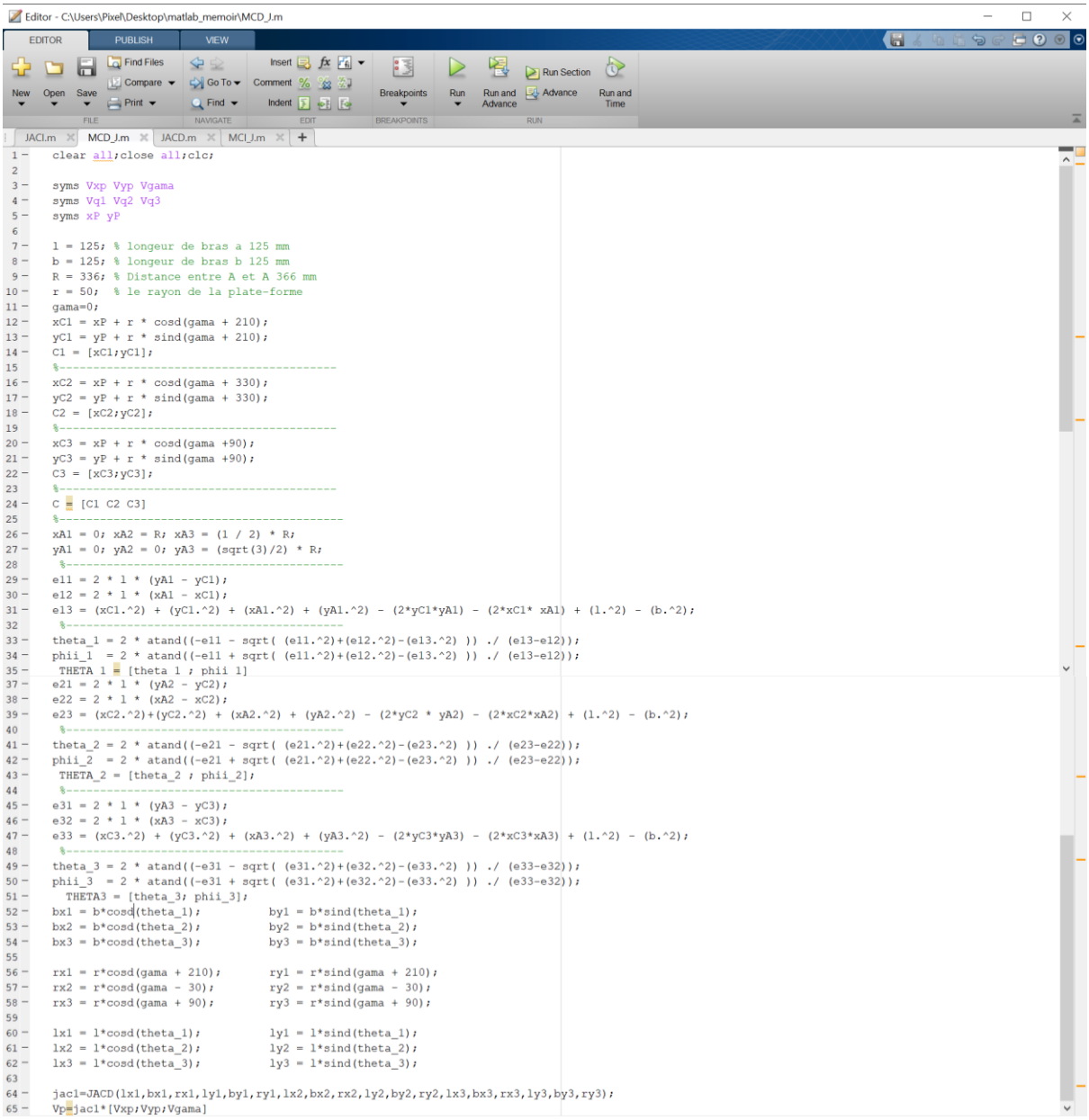


Figure 2. 11: programme Matlab de la modélisation cinématiques direct (MCD)

II.3.2) Le modèle cinématique inverse MCI du robot

Le MCI sera écrit à partir des deux matrices J_x et J_q de l'équation **(2.20)** qui sont deux matrices jacobiennes distinctes. Ainsi, l'ensemble de La matrice jacobienne du MCI peut être obtenue comme suit

$$J = J_q^{-1} J_x \qquad \textbf{(2.23)}$$

$$J = \begin{bmatrix} \dfrac{b_{1x}}{l_{1x}b_{1x} - l_{1y}b_{1y}} & \dfrac{b_{1y}}{l_{1x}b_{1x} - l_{1y}b_{1y}} & \dfrac{r_{1x}b_{1y} - r_{1y}b_{1x}}{l_{1x}b_{1x} - l_{1y}b_{1y}} \\ \dfrac{b_{2x}}{l_{2x}b_{2x} - l_{2y}b_{2y}} & \dfrac{b_{2y}}{l_{2x}b_{2x} - l_{2y}b_{2y}} & \dfrac{r_{2x}b_{2y} - r_{2y}b_{2x}}{l_{2x}b_{2x} - l_{2y}b_{2y}} \\ \dfrac{b_{3x}}{l_{3x}b_{3x} - l_{3y}b_{3y}} & \dfrac{b_{3y}}{l_{3x}b_{3x} - l_{3y}b_{3y}} & \dfrac{r_{3x}b_{3y} - r_{3y}b_{3x}}{l_{3x}b_{3x} - l_{3y}b_{3y}} \end{bmatrix}$$

Cette matrice jacobienne est utilisée pour transférer les vitesses de l'espace articulaire à l'espace cartésien

$$\dot{q} = J\dot{x} \tag{2.24}$$

Où $\dot{x} = [v_{Px}; v_{Py}; \dot{\gamma}]^{\mathrm{T}}$ et $\dot{q} = [\dot{\theta}_1; \dot{\theta}_2; \dot{\theta}_3]^{\mathrm{T}}$ sont des vecteurs de vitesse dans les espaces cartésiens et articulaires, respectivement.

La **figure 2.12** illustre ce modèle MCI.

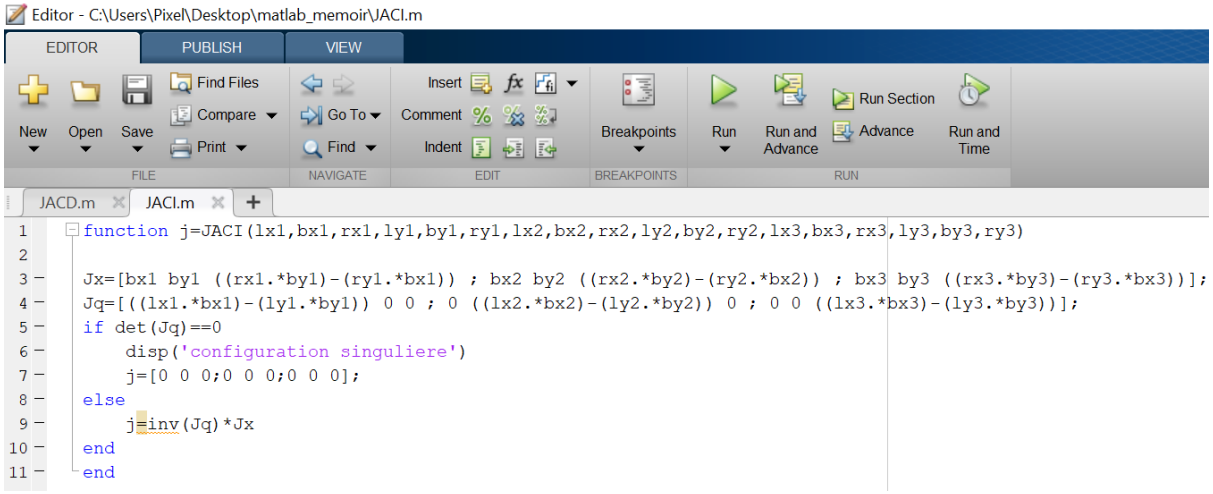


Figure 2. 12: programme Matlab de la matrice jacobienne inverse

Pour calculer la dynamique inverse du manipulateur, l'accélération de l'effecteur final est utilisée comme information d'entrée. Par conséquent, la relation entre l'articulation cartésienne et les accélérations peuvent être extraites par différenciation de l'équation **(2.22)** par rapport au temps **[11].**

$$\ddot{q} = J\ddot{x} + \dot{J}\dot{x} \tag{2.25}$$

Où $\ddot{x} = [\dot{v}_{px}; \dot{v}_{py}; \ddot{\gamma}]^{\mathrm{T}}$ et $\ddot{q} = [\ddot{\theta}_1; \ddot{\theta}_2; \ddot{\theta}_3]^{\mathrm{T}}$ sont des vecteurs d'accélération dans les espaces cartésiens et articulaires, respectivement. Dans l'équation **(2.25)**, on suppose que les autres quantités sont connues à partir de l'inversion de la vitesse et la seule matrice qui n'a pas encore été définie est la dérivée temporelle de la matrice jacobienne. La différenciation de l'équation **(2.23)** donne lieu à **[11]**

$$\dot{J} = \begin{bmatrix} K_1 & L_1 & R_1 \\ K_2 & L_2 & R_2 \\ K_3 & L_3 & R_3 \end{bmatrix} \tag{2.26}$$

K_i, L_i et R_i dans l'équation **(2.26)** peuvent s'écrire comme suite :

$$K_i = \frac{\dot{b}_{ix}(l_{ix}b_{ix} - l_{iy}b_{iy}) - b_{ix}(\dot{l}_{ix}\dot{b}_{ix} - \dot{l}_{iy}\dot{b}_{iy})}{(l_{ix}b_{ix} - l_{iy}b_{iy})^2} \quad \textbf{(2.27)}$$

$$L_i = \frac{\dot{b}_{iy}(l_{ix}b_{ix} - l_{iy}b_{iy}) - b_{iy}(\dot{l}_{ix}\dot{b}_{ix} - \dot{l}_{iy}\dot{b}_{iy})}{(l_{ix}b_{ix} - l_{iy}b_{iy})^2} \quad \textbf{(2.28)}$$

$$R_i = \frac{(\dot{r}_{ix}\dot{b}_{iy} - \dot{r}_{iy}\dot{b}_{ix})(l_{ix}b_{ix} - l_{iy}b_{iy}) - (r_{ix}b_{iy} - r_{iy}b_{ix})(\dot{l}_{ix}\dot{b}_{ix} - \dot{l}_{iy}\dot{b}_{iy})}{(l_{ix}b_{ix} - l_{iy}b_{iy})^2} \quad \textbf{(2.29)}$$

Editor - C:\Users\Pixel\Desktop\matlab_memoir\MCI_J.m

EDITOR | PUBLISH | VIEW

JACI.m | MCD_J.m | JACD.m | MCI_J.m

```
clear all;close all;clc;

syms Vxp Vyp Vgama axp ayp agama
syms Vq1 Vq2 Vq3
syms xP yP

l = 125; % longeur de bras a 125 mm
b = 125; % longeur de bras b 125 mm
R = 336; % Distance entre A et A 366 mm
r = 50;  % le rayon de la plate-forme
gama=0;
xC1 = xP + r * cosd(gama + 210);
yC1 = yP + r * sind(gama + 210);
C1 = [xC1;yC1];
%----------------------------------------
xC2 = xP + r * cosd(gama + 330);
yC2 = yP + r * sind(gama + 330);
C2 = [xC2;yC2];
%----------------------------------------
xC3 = xP + r * cosd(gama +90);
yC3 = yP + r * sind(gama +90);
C3 = [xC3;yC3];
%----------------------------------------
C = [C1 C2 C3]
%----------------------------------------
xA1 = 0; xA2 = R; xA3 = (1 / 2) * R;
yA1 = 0; yA2 = 0; yA3 = (sqrt(3)/2) * R;
 %----------------------------------------
e11 = 2 * l * (yA1 - yC1);
e12 = 2 * l * (xA1 - xC1);
e13 = (xC1.^2) + (yC1.^2) + (xA1.^2) + (yA1.^2) - (2*yC1*yA1) - (2*xC1* xA1) + (l.^2) - (b.^2);
 %----------------------------------------
theta_1 = 2 * atand((-e11 - sqrt( (e11.^2)+(e12.^2)-(e13.^2) )) ./ (e13-e12));
phii_1  = 2 * atand((-e11 + sqrt( (e11.^2)+(e12.^2)-(e13.^2) )) ./ (e13-e12));
 THETA 1 = [theta 1 ; phii 1]
e21 = 2 * l * (yA2 - yC2);
e22 = 2 * l * (xA2 - xC2);
e23 = (xC2.^2)+(yC2.^2) + (xA2.^2) + (yA2.^2) - (2*yC2 * yA2) - (2*xC2*xA2) + (l.^2) - (b.^2);
 %----------------------------------------
theta_2 = 2 * atand((-e21 - sqrt( (e21.^2)+(e22.^2)-(e23.^2) )) ./ (e23-e22));
phii_2  = 2 * atand((-e21 + sqrt( (e21.^2)+(e22.^2)-(e23.^2) )) ./ (e23-e22));
 THETA_2 = [theta_2 ; phii_2];
 %----------------------------------------
e31 = 2 * l * (yA3 - yC3);
e32 = 2 * l * (xA3 - xC3);
e33 = (xC3.^2) + (yC3.^2) + (xA3.^2) + (yA3.^2) - (2*yC3*yA3) - (2*xC3*xA3) + (l.^2) - (b.^2);
 %----------------------------------------
theta_3 = 2 * atand((-e31 - sqrt( (e31.^2)+(e32.^2)-(e33.^2) )) ./ (e33-e32));
phii_3  = 2 * atand((-e31 + sqrt( (e31.^2)+(e32.^2)-(e33.^2) )) ./ (e33-e32));
  THETA3 = [theta_3; phii_3];
bx1 = b*cosd(theta_1);          by1 = b*sind(theta_1);
bx2 = b*cosd(theta_2);          by2 = b*sind(theta_2);
bx3 = b*cosd(theta_3);          by3 = b*sind(theta_3);

rx1 = r*cosd(gama + 210);       ry1 = r*sind(gama + 210);
rx2 = r*cosd(gama - 30);        ry2 = r*sind(gama - 30);
rx3 = r*cosd(gama + 90);        ry3 = r*sind(gama + 90);

lx1 = l*cosd(theta_1);          ly1 = l*sind(theta_1);
lx2 = l*cosd(theta_2);          ly2 = l*sind(theta_2);
lx3 = l*cosd(theta_3);          ly3 = l*sind(theta_3);

jac2=JACI(lx1,bx1,rx1,ly1,by1,ry1,lx2,bx2,rx2,ly2,by2,ry2,lx3,bx3,rx3,ly3,by3,ry3);
Vq=jac2*[Vq1;Vq2;Vq3]

aj=diff(jac2)
aq=jac2*[axp;ayp;agama]+aj*[Vxp;Vyp;Vgama]
```

Figure 2. 13: programme Matlab de la modélisation cinématiques inverse

II.4) Conclusion

Dans ce chapitre, le problème de modélisation du robot parallèle plan 3-RRR est traité en utilisant les outils de la mécanique rationnelle. Tout d'abord, le modèle géométrique a été élaboré, par la suite c'est le modèle cinématique qui a été écrit en calculant les matrices jacobiennes correspondantes. Ces différents modèles ont été élaborés en tenant compte des particularités cinématiques du robot. Le prochain chapitre sera consacré à la conception et simulation d'un modèle de ce robot sous SolidWorks.

Chapitre III

Conception et analyse des mouvements du robot sous SolidWorks

III.1 Introduction

Notre projet de fin d'étude a pour objectif la mise en œuvre d'un robot parallèle planaire 3RRR. Dans ce chapitre, nous allons traiter la conception du robot ou on détaillera donc la conception des organes les plus importants du robot et l'étude des mouvements du robot en utilisant SolidWorks.

Le logiciel SOLIDWORKS est un modeleur volumique permettant de créer des pièces complexes en 3 dimensions. Ces pièces peuvent être ensuite utilisées pour créer des mises en plan en 2D et des assemblages de plusieurs pièces entre elles.

SOLIDWORKS est un système à cotation piloté. On peut spécifier des côtes et rapports géométriques entre les éléments. Un changement de côte entraîne un changement de taille de la pièce, tout en préservant l'intention de conception.

Un modèle SolidWorks est constitué de pièces, d'assemblages et de mise en plan. Les pièces, les assemblages et les mises en plan affichent le même modèle dans des documents différents. Les changements opérés sur le modèle dans l'un des documents se propagent aux autres documents contenant ce modèle. Le logiciel SolidWorks comprend 3 modules élémentaires : Pièce, Assemblage, Mise en plan **(Figure. 3.1),** on les a utilisés pour réaliser le modèle CAO de notre robot.

Figure 3. 1: Les différents modules élémentaires de SolidWorks

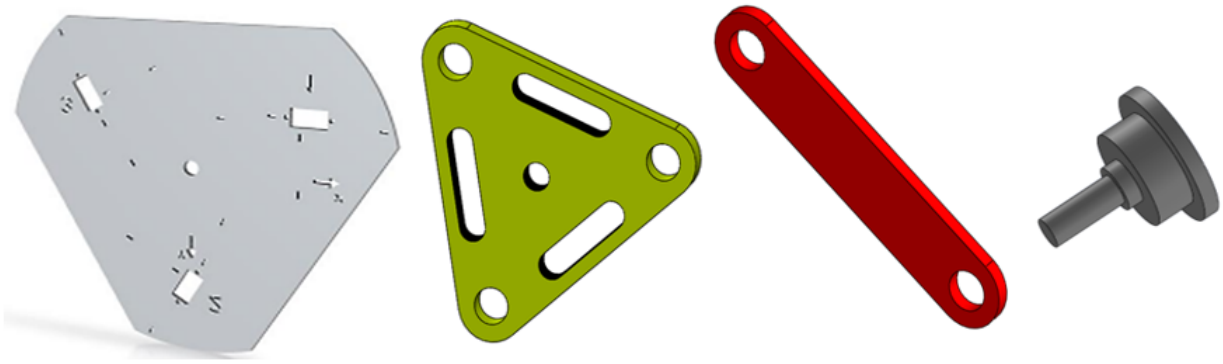
III.2) Conception du robot

Dans cette partie, on présente les organes les plus importants pour réaliser notre robot parallèle planaire 3RRR. La **figure 3.2** illustre les pièces nécessaires pour le robot et les noms auxquels elles seront référencées dans ce chapitre:

- **La table de base (fig3.2a):** Dans notre cas nous avons une table fixe selon l'axe « z », elle est la base de fixation des trois moteurs du robot, elle est donc liées

indirectement avec les bras auxquels on transmet le mouvement. Elle peut prendre différentes formes

- **La nacelle (fig3.2a):** Pièce centrale du robot car elle déplace l'outil selon des mouvements circulaires ou triangulaires selon l'axe « z », la tête de l'outil est en mouvement par rapport à la table fixe.
- **Les bras (fig3.2c): le robot comporte six bras identiques,** Ils ont un rôle important dans le déplacement du mouvement des axes vers la nacelle et aident également à déterminer le domaine du travail et de l'équilibre de la nacelle porte-outil, car la forme du robot parallèle aide à la maintenir en position droite pour la table.
- **Les axes (fig3.2d):** le robot comporte six liaisons passives identiques, elles sont assurées par des axes de pivotement assurant un mouvement avec frottement minimal.



(a) La base (b) la nacelle (c) bras (d) axe

Figure 3. 2: Principales Pièces du robot

L'assemblage des pièces (b,c,d) de la **figure 3.2,** nous permet d'avoir la chaine cinématique principale de notre robot illustrée par la **figure 3.3**.

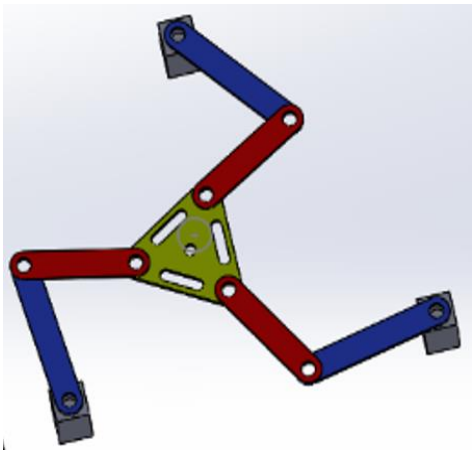


Figure 3. 3: Assemblage des principales Pièces du robot

Au fait pour créer l'assemblage du robot, nous avons suivi Les étapes suivantes :

- *Ouverture de SolidWorks :*
- Cliquez sur nouveau, après assemblage, cliquez sur ok :
- Cliquez sur parcourir on choisir la première pièce correspondent – ouvrir après on fixe la pièce avec double clique.
- Pour insérer la deuxième pièce cliquez sur insérés composants (**étape 1 de la figure 3.4**)–parcourir (**étape 2 de la figure 3.4**) et ainsi de suite.
- pour lier entre les éléments de robot on doit imposer des contraintes en cliquant sur le symbole des contrainte sur le menu (**étape 3 de la figure 3.4)** :

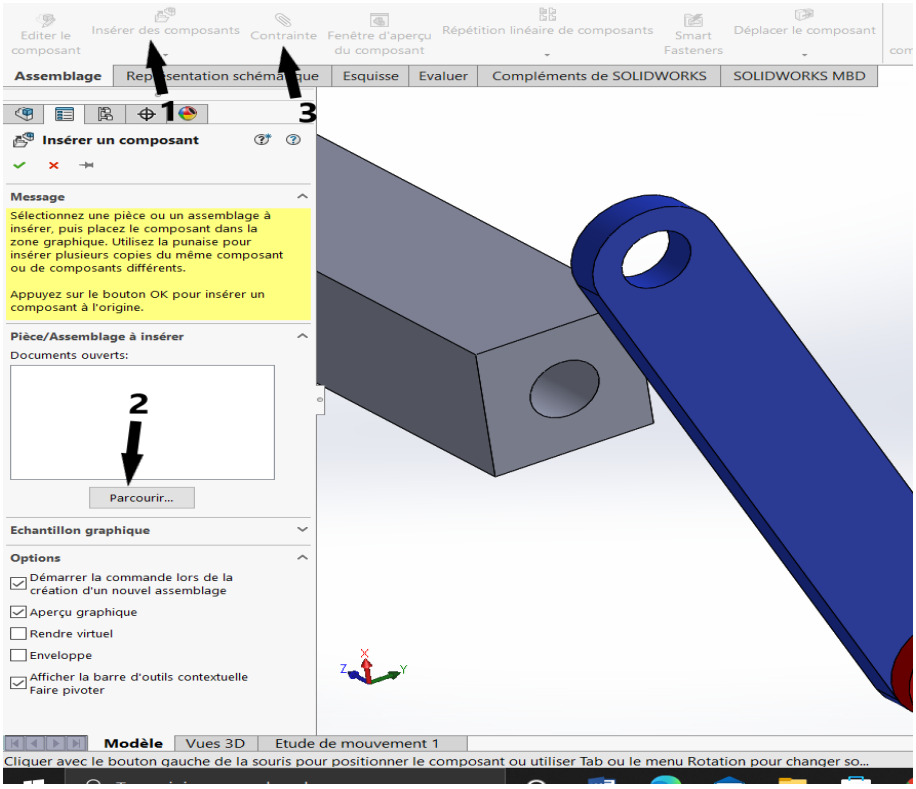


Figure 3. 4:Différentes étapes suivies pour effectuer l'assemblage

Attention, les composants fixes et mobiles dans SolidWorks Motion sont déterminés par leur état **Fixe/Libéré** dans le modèle SolidWorks. Dans notre cas, le composant **Base (les moteurs)** est fixe; tandis que les 7 autres éléments se déplacent.

- Sous **sélectionner des contraintes choisir** la face cylindrique de éléments 1 connectée par la face cylindrique de éléments 2 (**étape 4 de la figure 3.5**)
- Cliquez sur **coïncidente (étape 5 de la figure 3.5).**

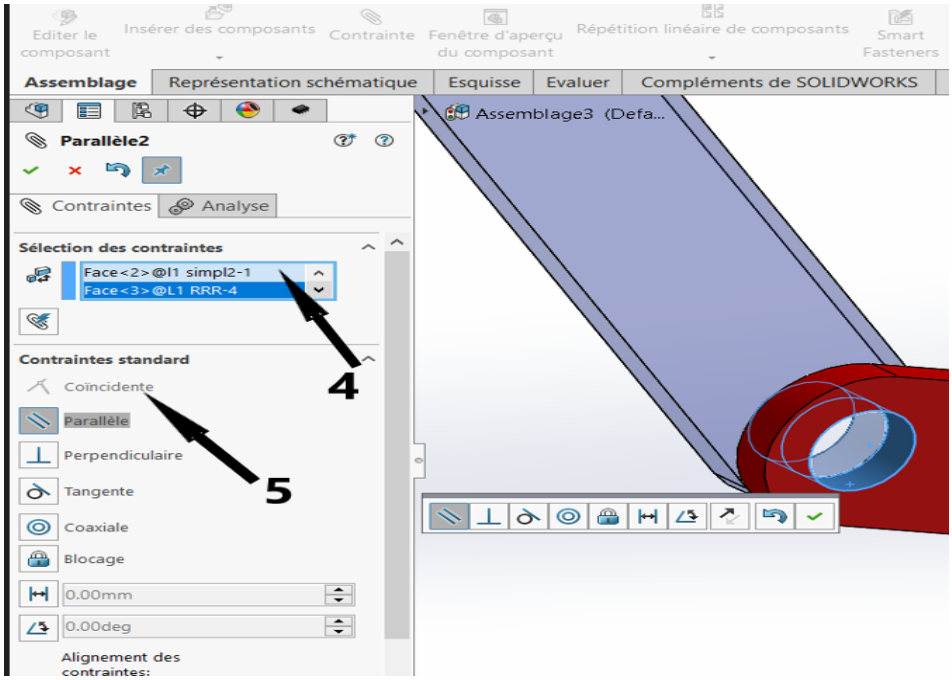


Figure 3. 5: Étapes suivies pour imposer les contraintes aux éléments du robot

Les étapes précédentes doivent être réalisées pour toutes liaisons du robot. Enfin, le mouvement du mécanisme peut être étudié.

III.3) Analyse des mouvements du robot à l'aide de SolidWorks *motion*

Les études de mouvement sont des simulations du mouvement des modèles d'assemblage qui se fait sous SolidWorks à l'aide de l'outil ***Motion***. C'est un outil de prototypage virtuel qui prend en charge l'animation, l'analyse et la conception de mécanismes. Au lieu de construire et de tester physiquement des prototypes de mécanismes, on peut utiliser *SOLIDWORKS Motion* (*SW-Motion*) pour évaluer et affiner le mécanisme avant de finaliser la conception et réaliser le prototypage fonctionnel. Cet outil a été utilisé d'une façon intensive dans notre projet pour

valider les différentes solutions mécaniques.

Les fonctionnalités disponibles dans *SW-Motion* aident également à rechercher de meilleures alternatives de conception. Un meilleur l'alternative de conception dépend beaucoup du problème. Il est essentiel qu'un objectif de conception soit clairement défini par le concepteur avant de rechercher de meilleures alternatives de conception.
A partir d'une étude de mouvement, vous pouvez utiliser le Motion Manager, une interface sous forme de chronogramme, qui comprend les outils d'étude de mouvement suivants:

Animation (disponible dans SOLIDWORKS)	Vous pouvez utiliser Animation pour animer le mouvement d'assemblages. • Ajoutez des moteurs pour piloter le mouvement d'une ou de plusieurs pièces dans un assemblage. • Spécifiez la position des composants de l'assemblage à divers moments en utilisant des clés. Animation utilise l'interpolation pour définir le mouvement des composants entre les clés.
Mouvement standard (disponible dans SOLIDWORKS)	Mouvement standard vous permet de simuler les effets de moteurs, de ressorts, de contacts et de la gravité sur des assemblages. Mouvement standard prend en compte la masse dans le calcul du mouvement. L'analyse étant relativement rapide, Mouvement standard peut être utilisé pour créer des animations de qualité professionnelle basées sur les lois de la physique.
Analyse de mouvement (disponible avec le complément SOLIDWORKS Motion™ dans SOLIDWORKS Premium)	Vous pouvez utiliser Analyse de mouvement pour simuler et analyser les effets d'éléments de mouvement (forces, ressorts, amortisseurs et frottement) sur un assemblage. Analyse de mouvement utilise des solveurs de cinématique puissants et tient compte des propriétés du matériau, ainsi que de la masse et de l'inertie, dans l'analyse. Analyse de mouvement peut aussi être utilisé pour tracer les résultats de la simulation en vue d'une analyse plus poussée.

Dans ce qui suit, nous allons aborder l'étude des mouvements du robot en utilisant SolidWorks motion qui sera présente brièvement dans la suite. On imposera le suivi d'un cercle par le centre de la nacelle. Les Étapes suivies pour réaliser l'étude de mouvement du robot sont indiquées sur la **figure 3.6.**

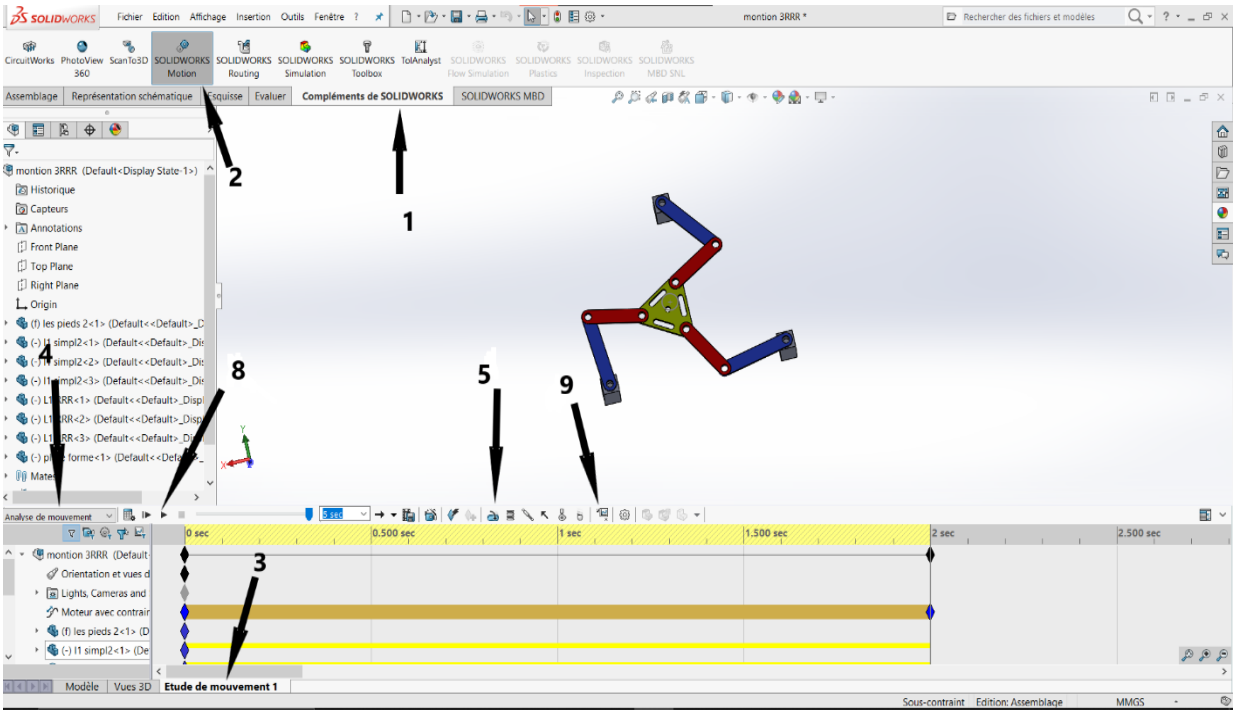


Figure 3. 6: Étapes suivies pour réaliser l'étude de mouvement du robot

Nous suivons les étapes suivantes **:**

1 / Cliquez sur **compléments de SolidWorks**
2/ Activer **SolidWorks motion**

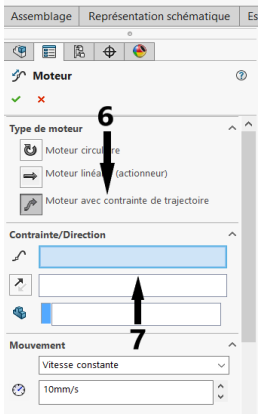
3/ Cliquez sur **étude de mouvement 1**

4/ activez **animation sur analyse de mouvements**

5/ Sous Type de **moteur**,

6/ sélectionnez **moteur avec contrainte de trajectoire** .

7/ Sous Composant/Direction, **sélectionnez la face cylindrique**



8/ Cliquez sur Lecture du début ▶ (barre d'outils Motion Manager) pour lire la simulation du début.

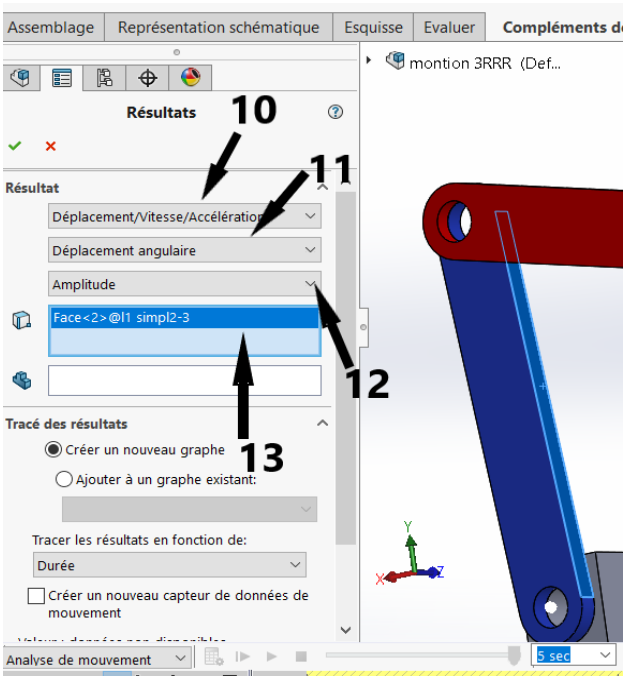
Ensuite, après la fin des calculs, pour afficher les résultats de différents graphes ; nous suivrons les étapes suivantes :

9/ Cliquez sur l'icône **Résultats et graphes pour ouvrir la boîte de dialogue Résultats.**

10 / sous Résultats, **sélectionnez Déplacement/ Vitesse/Accélération,**

11/ Accélération angulaire et Composant Z.

12 / cliquez **sur amplitude**



13/ Toujours sous Résultats, **sélectionnez** la face de bras 1. Le Composant pour définir les directions XYZ (facultatif) sert à référencer nos tracés de résultats par rapport au système de coordonnées local d'un autre composant mobile. Pour tracer les résultats du système de coordonnées par défaut indiqué sur la figure, n'effectuez aucune entrée dans ce champ. Cliquez sur OK pour montrer le tracé**. [12]**

Les figures suivantes illustrent les courbes de déplacements angulaires et les vitesses obtenues pour le chemin circulaire imposé à la nacelle.

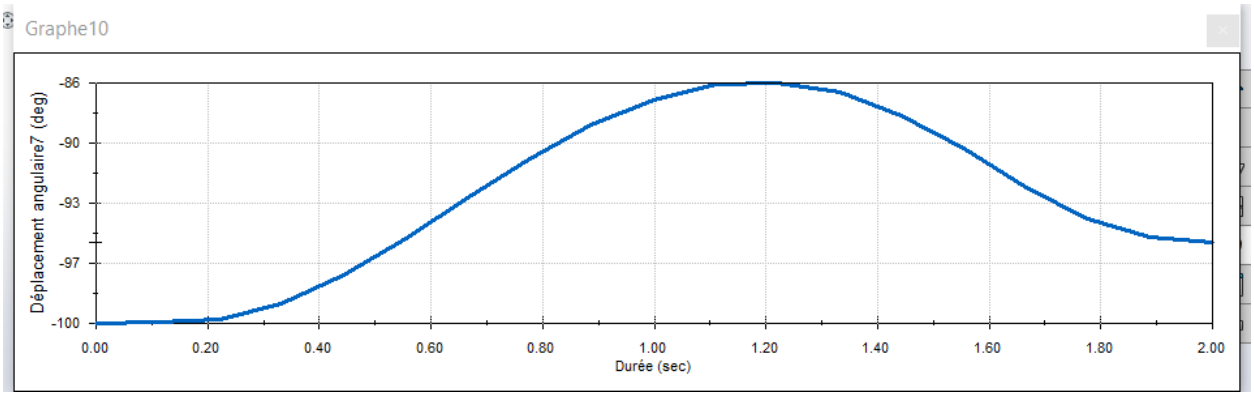


Figure 3. 7: résultat de déplacement de premier bras L1

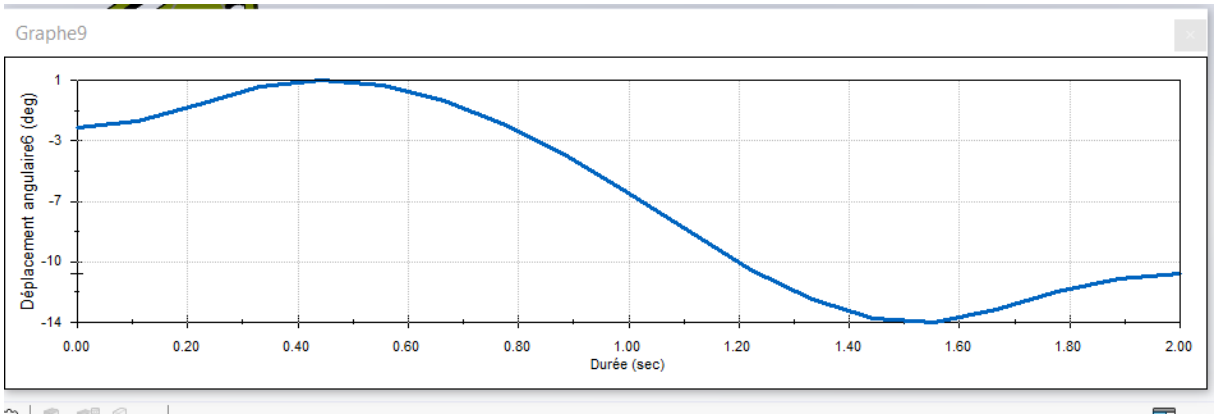


Figure 3. 8: résultat de déplacement du deuxième bras L2

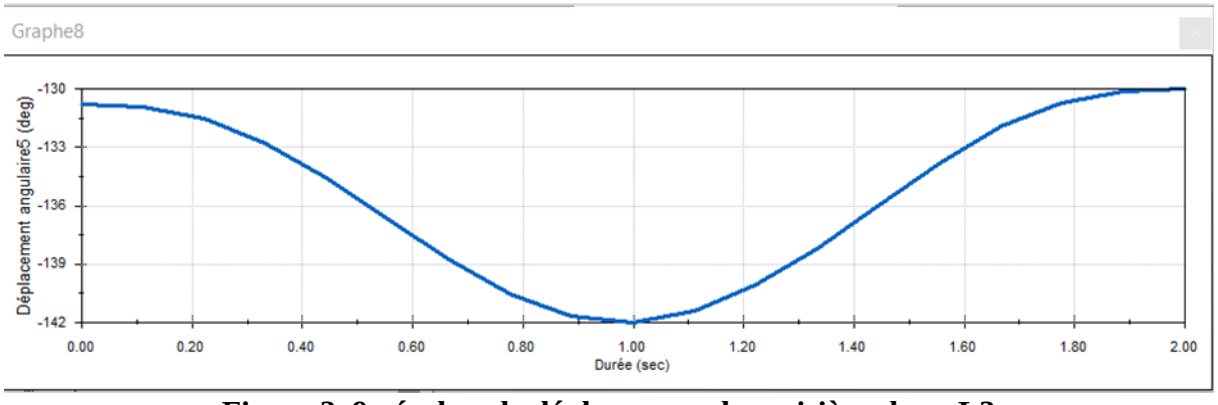


Figure 3. 9:résultat de déplacement du troisième bras L3

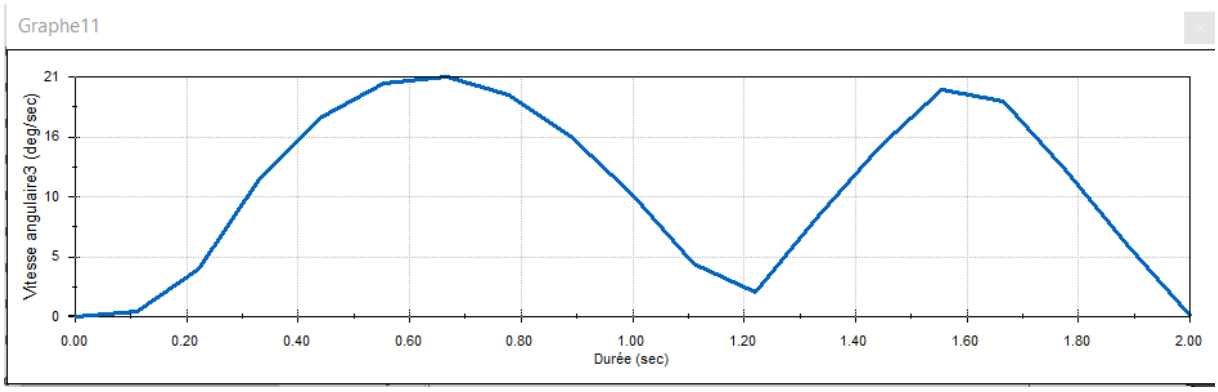


Figure 3. 10: résultat de vitesse de premier moteur L1

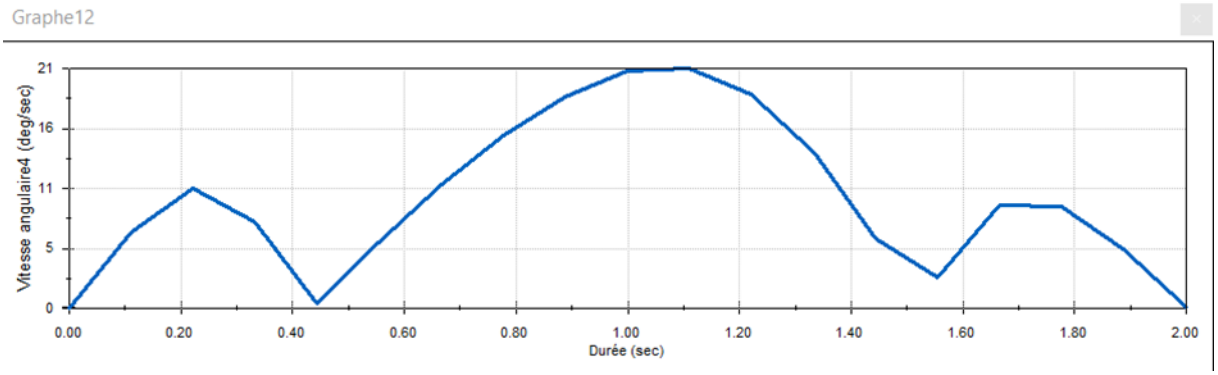


Figure 3. 11: résultat de vitesse de deuxième moteur L2

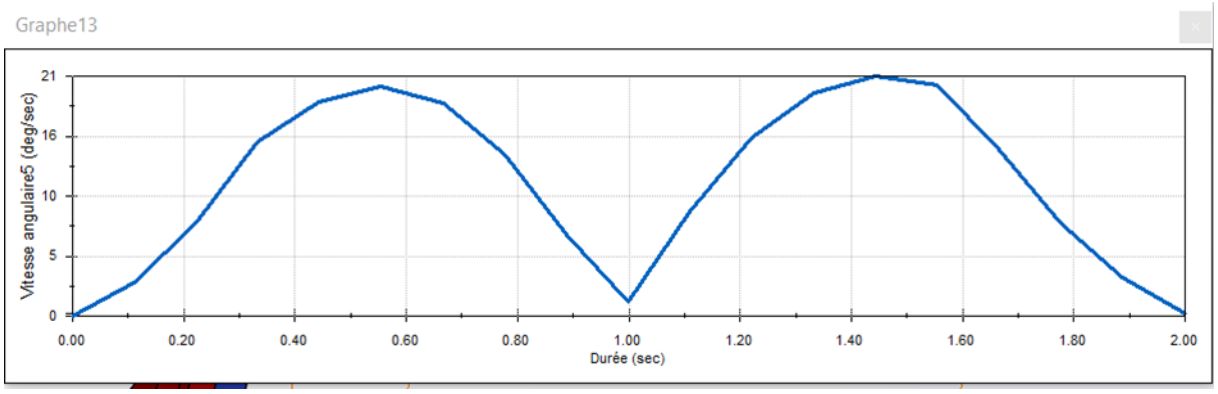


Figure 3. 12: résultat de vitesse de troisième moteur L3

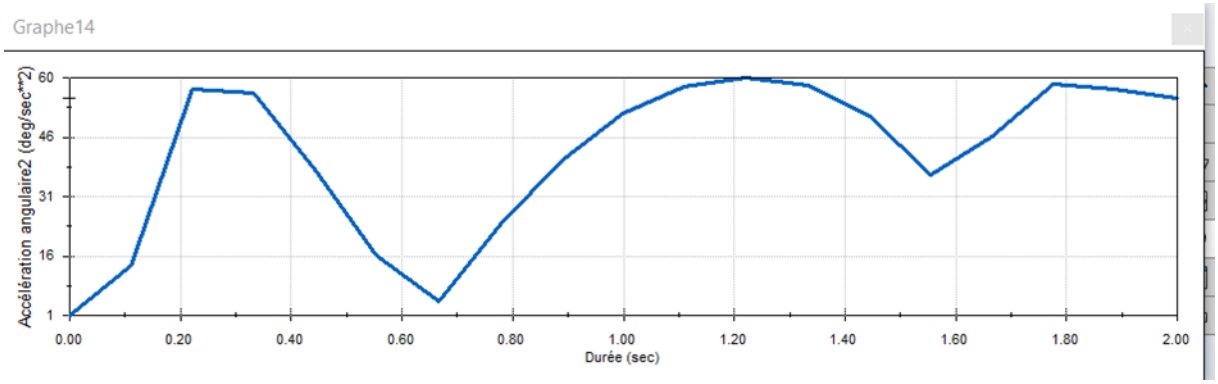


Figure 3. 13: résultat de l'accélération de premier moteur L1

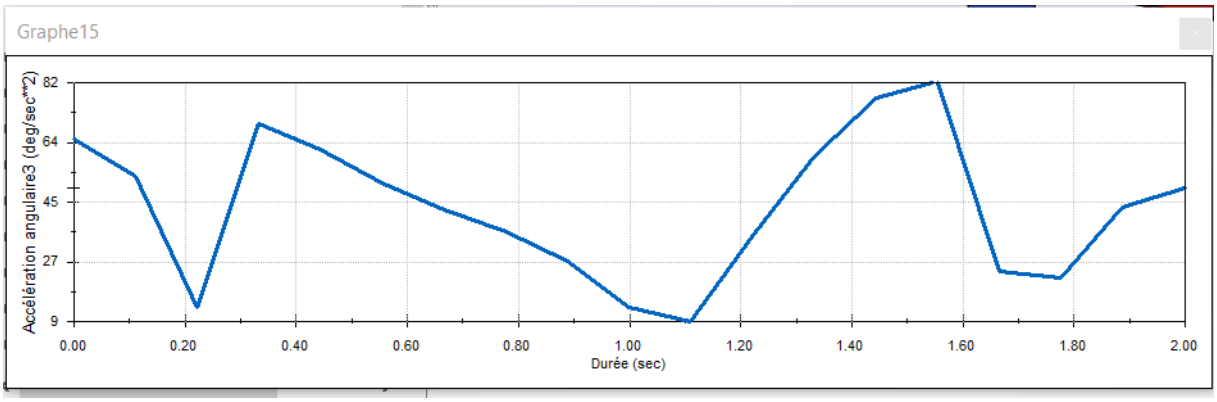


Figure 3. 14: résultat de l'accélération de deuxième moteur L2

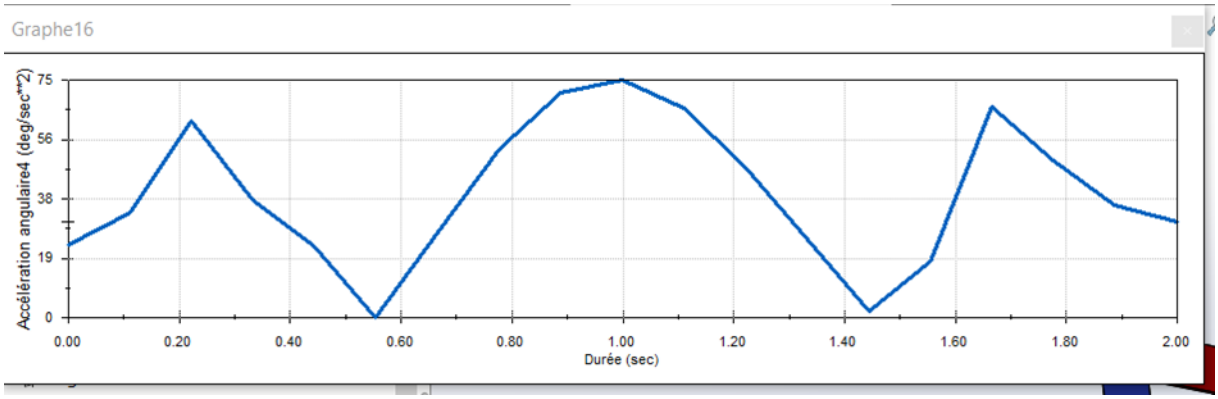


Figure 3. 15: résultat de l'accélération de troisième moteur L3

III.4) Conclusion

L'Objectif de ce chapitre était de présenter l'analyse de mouvement de la structure de base de notre robot parallèle 3RRR à l'aide de l'outil complémentaire de l'analyse des mouvements de SolidWorks. Nous avons dans un premier temps passé en revue les éléments de base de cet outil par la suite, nous avons présenté son application à notre cas. Dans le prochain chapitre nous présenterons la partie électrique et informatique de notre robot.

Chapitre IV

Présentation de la partie Commande du robot

IV.1) Introduction

Ce chapitre sera consacré à la description du développement de la partie commande du robot qui porte sur les parties électrique et informatique de notre robot. Nos choix se sont naturellement tournés vers les solutions disponibles sur le marché local. Nous utiliserons particulièrement l'environnement de programmation Matlab pour réaliser l'**I**nterface **G**raphique **U**tilisateur (Graphic User Interface, GUI) pour contrôler les servomoteurs MG996R du robot via une carte Arduino UNO, cela en utilisant d'autres composants électroniques et électriques qui seront présentés par la suite.

IV.2) Description des principaux éléments utilisés

IV.2.1) Description de la carte Arduino UNO

Pour l'électronique, on va se baser sur une carte Arduino qui est simple à prendre en main. La carte Arduino est un circuit imprimé en matériel libre (les plans de la carte elle-même sont publiés en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage…), le pilotage d'un robot, etc. une carte Arduino est donc carte électronique équipée d'un micro-contrôleur programmable. La carte Arduino la plus utilisée est la carte Arduino Uno (figure 4.1), elle sera employée pour commander notre robot.

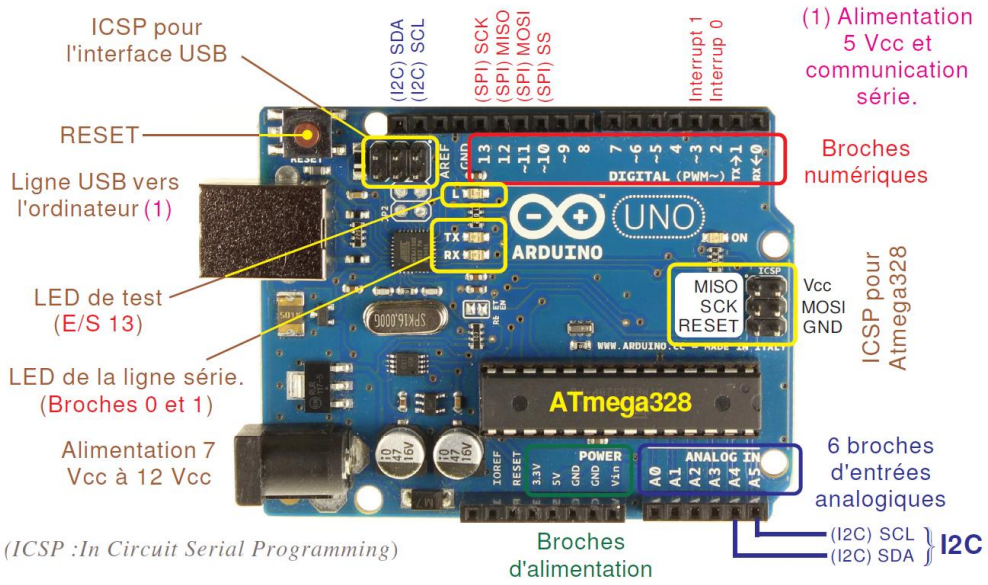


Figure 4. 1: la carte Arduino UNO

Les principales Caractéristiques techniques de la carte Arduino UNO sont:

- Micro-contrôleur : ATmega328.

- Fréquence horloge : 16 MHz.
- Tension d'alimentation interne : 5Vcc.
- Tension d'alimentation externe recommandée : 7Vcc à 12Vcc. *(Limites : 6Vcc à 20 Vcc)*
- Courant max sur la sortie 3,3V généré par le régulateur interne : 50mA.
- Entrées/sorties binaires : 14 broches.
- Courant MAX par broches en sortie : 40 mA. *(85 mA en court-circuit)*
- Courant MAX cumulé par les broches en sorties : 200 mA. *(Soit 14 mA en moyenne)*
- Les E/S binaires 0 et 1 sont mobilisées par le dialogue sur la ligne série.
- S0 pour RX et S1 pour TX. Chaque broche est reliée à une LED via une résistance de 1kΩ.
- Les E/S binaires 3, 5, 6, 9, 10, et 11 sont dédiées au mode PWM.
- L'E/S 13 est reliée sur la carte à la LED de test via une résistance de 1kΩ.
- Entrées analogiques : 6. Le niveau logique maximal doit être de +5Vcc.
- Mémoire Flash 32 KB dont 0.5 KB utilisée par le Bootloader.
- Mémoire SRAM 2 KB. Mémoire EEPROM 1 KB.
- La carte s'interface au PC par l'intermédiaire de sa prise USB.
- La carte s'alimente par le jack d'alimentation. *(Utilisation autonome)*

IV.2.2) Description des servomoteurs utilisés MG 996R

Pour la motorisation, nous allons utilises les servomoteurs MG 996R (**figure 4.2**), qui sont des moteurs a courant continu asservis en position par la technique PWM (pulse Widh modulation) avec un débattement angulaire de 180 degrés, il s'alimente d'un courant continu avec un tension 4.8V-6V et délivre un couple max de 11kg/cm. Le servo peut convertir de l'énergie électrique a l'énergie mécanique sous forme de mouvement de rotation. Le MG996R est un servomoteur numérique robuste et dispose de pignons et roulements métalliques, il est conçu pour produire des mouvements précis d'un élément mécanique selon un commande externe qui agit sur la largeur des impulsions PWM.

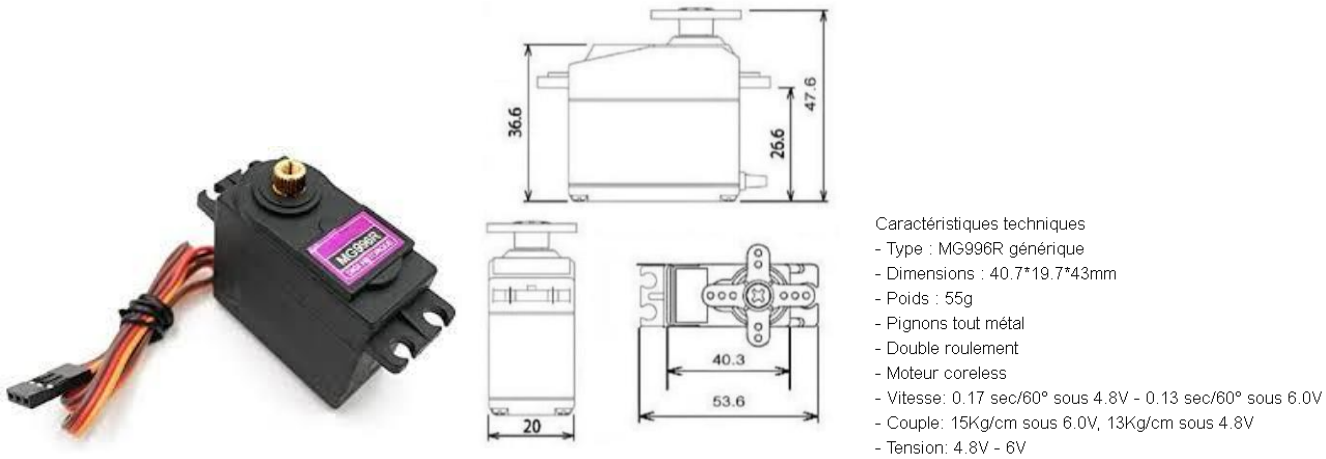


Figure 4. 2: Le servo moteur MG996R et ses caractéristiques.

IV.2.3) Description de l'Alimentation utilisée

Pour l'alimentation électrique, nous allons utiliser un transformateur électrique (AC/DC ADAPTER) illustré par la **figure 4.3** pour alimenter les 3 servo-moteurs. Les trois servo-moteurs ont besoin d'une tension a 5V-3A ce qui est en fait assuré par notre adapteur électrique qui a à l'entrée : 100-240V AC 50/60Hz et à la sortie :5V DC 3A.

Figure 4. 3: Transformateur électrique AC /DC ADAPTER

IV.2.4) l'assemblage

Pour assembler les composants il faut savoir que les servo-moteurs ont 3 fils, et chaque fil a une couleur, le noir pour le négatif, le rouge pour le positif et le jaune pour le signal. On commence l'assemblage en attachant les fils noirs (négatif) et les fils rouges (positif) de chaque servo-moteur avec les bornes électroniques de la plaque d'essai. On raccorde un fil négative avec le GND de la carte Arduino UNO, après on choisit 3 broches numérique des 14 existants sur la carte Arduino UNO et on les relier avec les fils de signal. On attache aussi les fils de

l'alimentation (transformateur électrique ou piles) avec la plaque, le nigatif avec le côte nigatif de la plaque,et le positif avec cote positif de la plaque comme indiquer la **figure 4.4.**

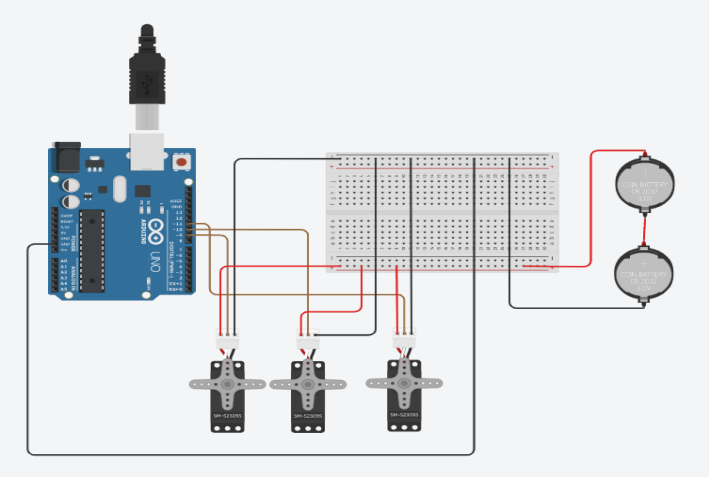


Figure 4. 4: Assemblage des composants

IV.3) Développement de l'interface informatique

Le développement de l'interface informatique s'est fait à l'aide de MATLAB. Matlab est un langage de script émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société **The MathWorks**, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran.

Nous avons utilisé MATLAB pour écrire un programme pour contrôler des servomoteurs connectés à une carte Arduino. Pour cela, il fallait installer les packages de prise en charge MATLAB de Arduino **(Figure.4.5)** :

- démarrez MATLAB et cliquez sur Modules Add_Ons>Get Hardware Support Packages
- Sélectionnez-le package que vous souhaitez installer et suivez les instructions d'installation dans la fenêtre du programme d'installation.

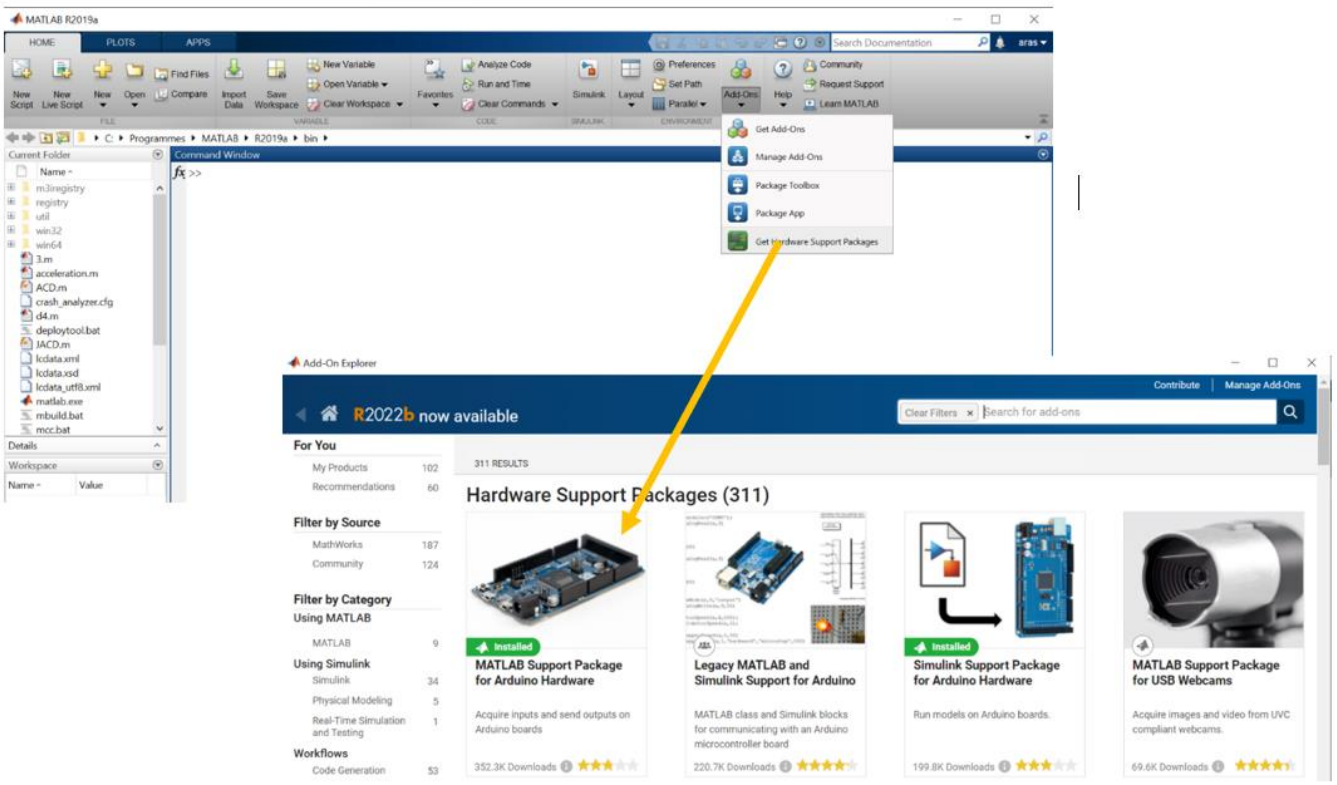


Figure 4. 5: environnement Matlab et Ajout du support de Arduino.

Une fois les packages installés, on teste la communication Matlab-Arduino : on a connecté notre Arduino au PC avec un câble USB et exécuté la commande suivante :

```
>> a = arduino ()
```

À ce stade, MATLAB tentera de communiquer avec l'Arduino. Si cela réussit, l'écran **(figure 4.6)** affichera quelque chose comme ceci :

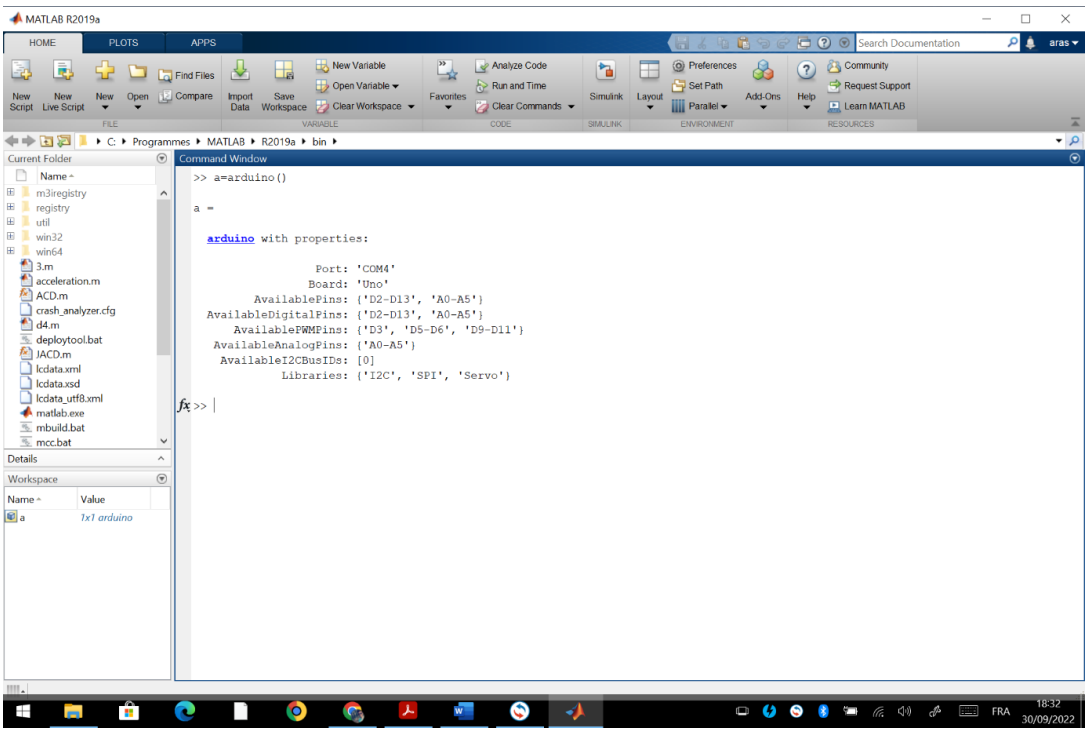


Figure 4. 6: message de confirmation de connectivité *Matlab-Arduino*

Cet écran indique que mon Arduino UNO est connecté à COM4. Si MATLAB n'a pas pu se connecter à l'Arduino, un message d'erreur s'affichera. Une fois le la liaison configurée, on va passer au développement de l'interface informatique GUI. Pour cela on va utiliser l'outil **guide** de Matlab **(Figure 4.7)**.

Pour le développement de l'interface graphique **(GUI)** de commande du robot nous avons utilisé principalement les fonctions du Matlab suivantes :

- **Servo () :** Un objet servo représente une connexion à un servomoteur sur Arduino hardware.
- **writePosition () :** Cette fonction MATLAB écrit la valeur spécifiée au servo spécifié sur le Arduino hardware.
- **readPosition () :** Cette fonction MATLAB lit la position de l'arbre du moteur et renvoie la valeur de la position en rapport avec l'angle maximum du servomoteur spécifié.

A la fin nous avons établi l'interface de la figure 4.8. Pour contrôler notre robot nous avons le choix entre les boutons de l'interface suivants

- **Se connecter au robot :** permet de connecter Matlab avec l'Arduino pour contrôler les servomoteurs
- **Home position** : les servomoteurs reviennent à la position initiale.
- **Se déconnecter** : déconnecter la carte Arduino avec Matlab
- **Sens horaire** et **Sens antihoraire :** des boutons pour contrôler les angles des servomoteurs
- **Position actuelle du robot :** permet d'afficher les positions et les angles des servomoteurs
- **Save position :** permet de sauvegarder les potions et les angles en fonction d'une tache dans un dossier (current folder)
- **Visiter les configurations sauvegardées :** autoriser de visiter les positions qu'on a sauvegardées
- **Exécuter la tâche :** nous offre de retourner à la tâche qui nous avons sauvegardé dans un dossier d'après avoir écrit le nom de la tache

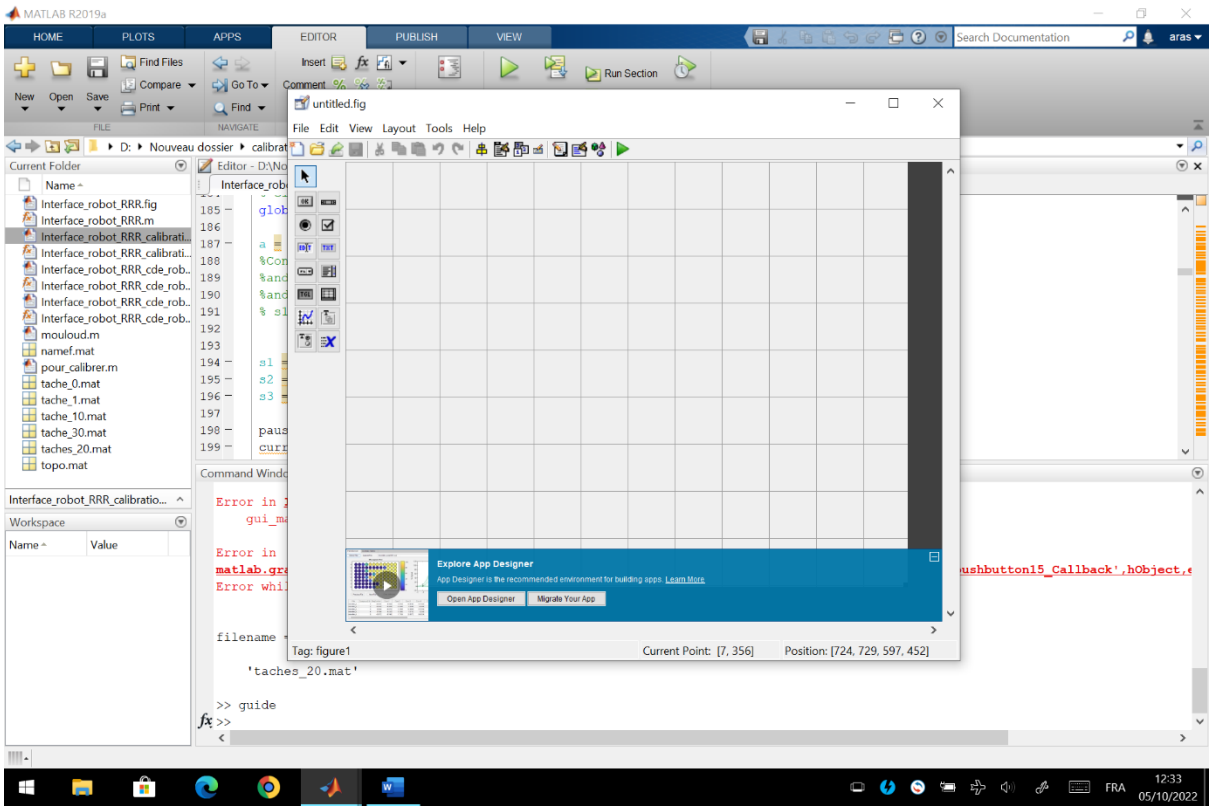


Figure 4. 7: Outil guide de MATLAB pour développer les GUI

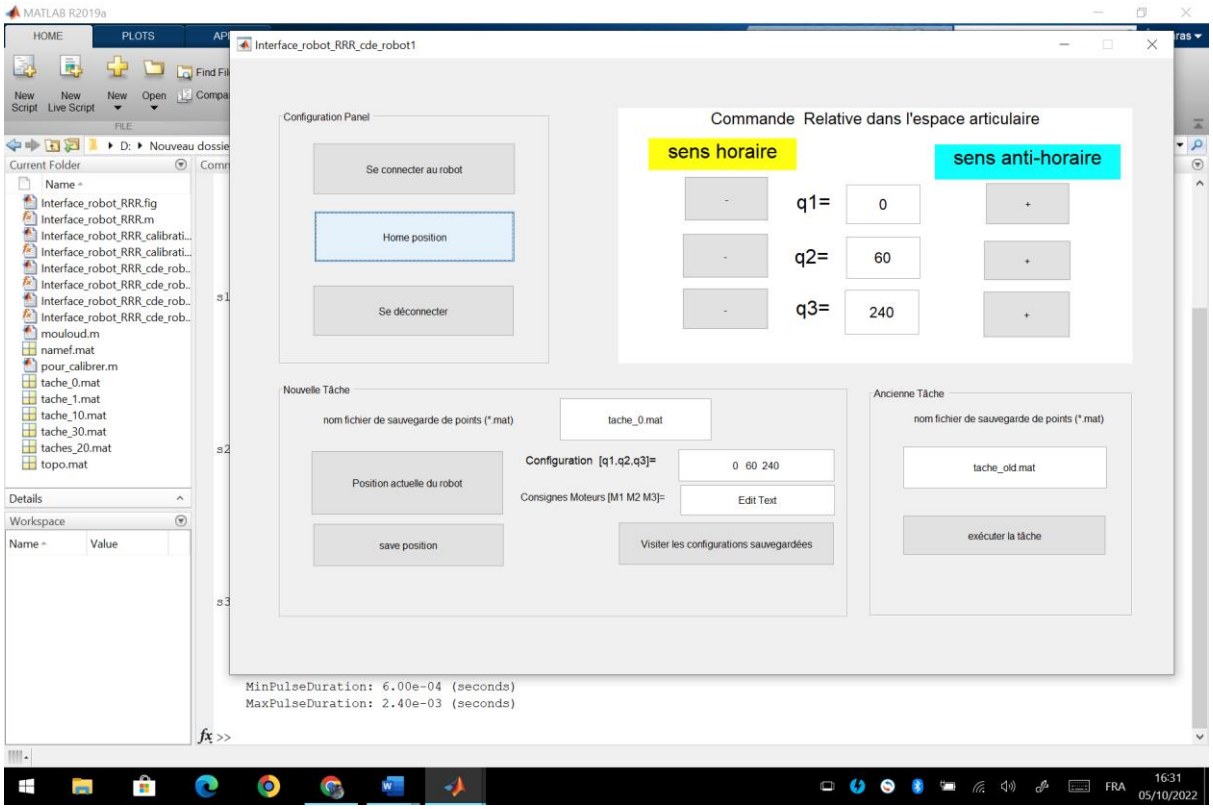


Figure 4. 8: interface pour contrôler le robot

IV.4) Fabrication et de réalisation des composants du robot

D'abord on a commencé par la conception de la gamme d'usinage des composants du robot (voir les composants dans le chapitre III)

- **La nacelle :** matière en plastique (**téflon**), on l'a passé à la fraiseuse pour effectuer l'épaisseur étudiée, et pour diminuer le poids de la nacelle (plateforme) on a percé 3 trous perpendiculairement étudié.
- **Les bras :** matière en plastique (**plexiglass**), on a tracé sur la plaque des formes rectangulaires après on les a coupé sous dimensions : (149mmx24mm) avec des corners arrondis, puis on a percé deux trous dans les deux extrémités du bras pour les connecter avec les servomoteurs en utilisant les axes.
- **Les axes :** matière en (**aluminium**), sous forme arrondi avec un épaulement pour éviter le frottement des bras, fabriqué par la machine Tour.
- **La base :** matière en (**bois bakélisé**), on a tracé et percé la table pour fixer l'ensemble du robot.

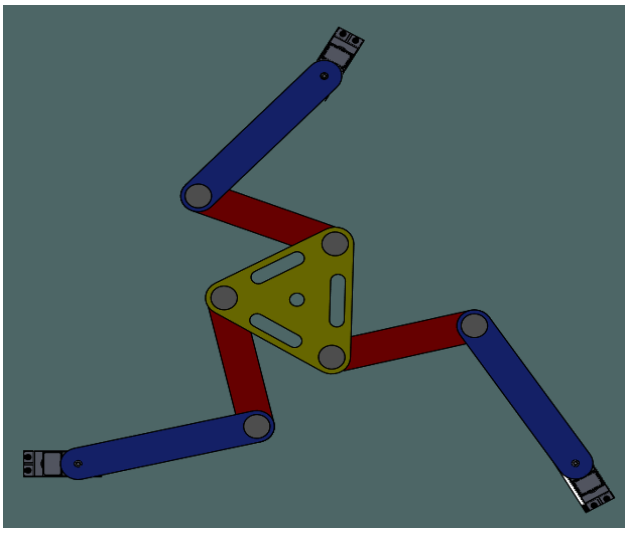


Figure 4. 9:Assemblage Final du robot

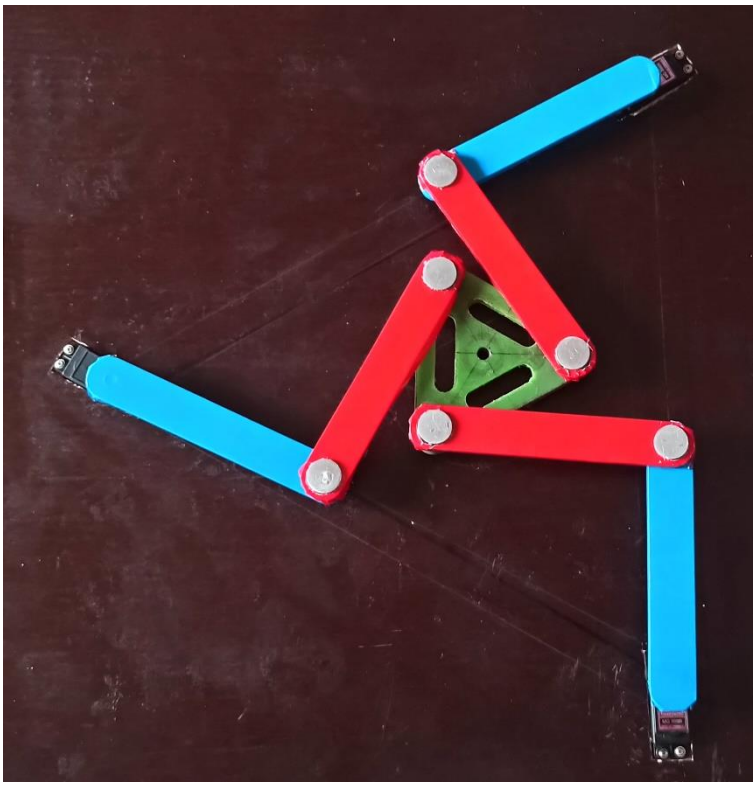


Figure 4. 10:Photo réelle du robot

IV.5) Conclusion

Dans ce chapitre, nous avons présenté les principaux composants électriques utilisés dans notre projet ainsi que leur câblage. En outre nous avons décrit la partie informatique du robot. Par ailleurs, Nous avons réussi à contrôler les servomoteurs en utilisons la carte Arduino UNO, le montage final et satisfaisant et nous avons pu réaliser des mouvements avec la structure réalisée.

Conclusion générale

Le travail présenté dans ce mémoire visait la Conception et réalisation d'un robot parallèle planaire 3 RRR à architecture parallèle pouvant réaliser diverses tâches de haute précision. La partie mécanique responsable du positionnement de l'outil dans le plan est basée sur la structure d'un robot parallèle planaire trois rotations de liberté 3RRR qui assure aussi bien la précision de positionnement.

En conclue , notre travail a permis d'aboutir aux résultats suivants :

- La mise au point de différents modèles analytiques du robot qui ont servis de base pour générer les trajectoires d'usinage du robot.
- La simulation cinématique et dynamique du comportement du robot après l'assemblage sous SolidWorks pour apprécier son comportement, dimensionner ses actionneurs et évaluer éventuellement ses performances.
- L'adaptation de solutions informatiques d'usage commun dans le domaine des amateurs (Makers) des mini-CNC ou imprimantes 3D pour proposer une chaine numérique de CFAO utilisable avec le robot.

A travers notre étude, nous avons pu présenter une méthodologie générale pour l'intégration et l'adaptation des robots aux opérations.

Comme complément à ce travail, nous proposons les points suivants :

- Régler les paramètres des commandes visant à assurer le comportement correct du système en termes de précision et d'efficacité.
- Réaliser d'une structure mécanique plus rigide afin de supporter des efforts plus importants générés lors des opérations (traçage ; soudage ; perçage ; découpage …), en tenant compte des aspects ergonomiques et économiques. Il faut garder en tête le fait que l'architecture robot de base lui permet d'effectuer des opérations à Grande Vitesse si on utilise les bons actionneurs.
- S'attaquer au problème de la calibration, de manière à minimiser les sources d'erreurs susceptibles de diminuer la précision du robot.
- Ce robot il peut faire plusieurs taches dans des différents domaines (médicales, impression 3D, fabrication textile, maintenance et réparation électronique, design…), on gardant la même conception de fonctionnement et les paramètres des commandes variables.

Enfin, nous espérons que ce modeste travail pourrait servir de référence aux Projets futurs des prochaines promotions et les inciter ainsi à s'intéresser d'avantage au côté pratique de mécanique en l'occurrence l'application de la robotique à la fabrication.

BIBLIOGRAPHIE

[1] : memoir saadi ramezy-salhi nessredine

[2] : brevet us-no-1789.680

[3] : memoir amori ammare univ oum el bouaghi

[4] : merlet j-p les robots paralleles hermes paris 1997

[5] : clavel.r « delta fast root with parallèle geometry » pro int symposimon industruiel robot .avril 1988.p91-100

[6] : bourbounnia-francis pdf

[7]: (Optimization Design by Genetic Algorithm Controller for Trajectory Control of a 3-RRR Parallel Robot. (2018, January 15). School of Mechatronic Engineering, China University of Mining and Technology, china)

[8]: Abdelouahab, C. (2018-2019). commande des Robots Manipulateurs. Faculté des Sciences et de la Technologie Département d'Automatique, Jijel.

[9]: Wenger, D. C. (s.d.). The Kinematic Analysis of a Symmetrical Three-Degree-Freedom Planar Parallel Manipulator. rue de la Noë, 44321 Nantes, France, Institut de Recherche en Communications et Cybernétique de Nantes1.

[10]: John Wiley AND Sons, I. .. (1999, Février). RABOT ANALYSIS : The Mechanics of Serial and Parallel Manipulators. Mechanical Engineering and Institute for systems Resezrch University of Maryland, canada

[11]: Küçük, S. (2012, Mars). Serial and Parallel Robot Manipulators – Kinematics, Dynamics, Control and Optimization. Janeza Trdine 9, 51000 Rijeka, Croatia, Croatia.

[12]: (introduction aux applications d'analyse de mouvement avec SolidWorks Motion, Livret de travail de l'étudiant)