

# وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat  
Département: Electronique

## MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

**Intitulé :**  
**Modélisation et Commande par Microcontrôleur  
d'une Plateforme Stewart à 3 Degrés de Liberté.**

**Domaine : Sciences et Technologie**

**Filière : Electronique**

**Spécialité : Système embarqué**

**Par :**

**Mouassa Abdennour**

**DEVANT Le JURY**

Président :	K. Chaker	Dr	UBM Annaba
Directeur de mémoire :	B. Boulebtateche	Dr	UBM Annaba
Examineurs :	A. Yahi	Dr	UBM Annaba
Examineurs :	Ait izem	Dr	UBM Annaba

## Résumé

Dans ce projet nous avons mis en œuvre nos compétences pour concevoir, réaliser, modéliser et commander une Plateforme Stewart à 3 Degrés de Liberté. Cette plateforme est pilotée par Arduino. Plusieurs scénarios de commande sont utilisés, notamment via un PC en adoptant une interface graphique 'GUI' sous Matlab et manuellement à travers un joystick et par un smartphone (à travers son gyroscope et son Bluetooth). Cette plateforme a été construite avec succès et les résultats des tests effectués sont très satisfaisants et concluants

**Mots clés :** Modélisation, Commande, Arduino, Plateforme de Stewart, 3 degrés de liberté, Matlab, joystick, Bluetooth.

## Abstract

The aim of this project is to design and to implement a system of Stewart Platform for 3 Degrees of Freedom. Controlling the platform was conducted by Arduino microcontroller under different scenario: through a PC using a graphical interface GUI under Matlab, manually using a joystick and wirelessly by a smartphone (through the Bluetooth and gyroscope). The platform was constructed with success and excellent test results were obtained.

**Keywords:** Modeling, Command, Arduino, Stewart Platform, 3 degrees of freedom, Matlab, joystick, Bluetooth.

## ملخص

في هذا العمل قمنا بإنجاز وتصميم مشروع يستطيع التحكم في منصة ستيوارت ذات ثلاث درجات حرية متصلة بالآردوينو بطرق مختلفة كإستخدام الحاسوب عن طريق واجهة رسومية للمستخدم في Matlab أو يدويا عن طريق الجويستيك أو لاسلكيا عن طريق الهاتف الذكي (بواسطة البلوتوث والجيروسكوب). تم إنجاز هذه المنصة بنجاح وكانت نتائج الاختبارات دقيقة وراضية.

**كلمات مفتاحية :** التحكم، منصة ستيوارت، آردوينو، حاسوب، ثلاث درجات حرية، واجهة رسومية، Matlab، جويستيك بلوتوث، جيروسكوب.

## *Dédicace*

*A mes chers parents,  
pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien  
et leurs prières tout au long de mes études,*

*A mes chères sœurs ..... pour leurs encouragements  
permanents, et leur soutien moral,*

*A mon cher frère et sa petite famille, ....., pour leur appui et  
leur encouragement,*

*A la mémoire de ma chère tante Karima*

*A toute ma famille pour leur soutien tout au long de mon parcours  
universitaire,*

*A mes chers Amis « Billel, Fateh, Med lamine, Mehdi »  
A mes collègues*

*Merci d'être toujours là pour moi.*

*Mr Mouassa Abdennour.*

## *Remerciement*

*Je remercie en premier lieu ; Dieu le tout puissant, le seigneur de tous les temps, pour son aide et sa grande miséricorde, car c'est grâce à lui que je suis arrivé à accomplir et à présenter ce modeste travail.*

*Je tiens à exprimer mes vifs remerciements à mon encadreur « Dr. Boulebtateche »; pour les considérables efforts qu'il a fournis afin de m'aider et m'éclairer de ses précieux conseils et dont l'amabilité et la patiente exemplaire m'a aidé à mener à bien mon travail, et surtout pour son gentillesse.*

*Je remercie également ma sœur « Sarra » pour ses précieux conseils, et aussi tous les responsables, le personnel et les enseignants du département d'électronique pour tout le savoir qu'ils nous ont transmis durant tout le cycle universitaire.*

*Grand merci à toute personne ayant aidé de près ou de loin à l'accomplissement de ce modeste travail et tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.*

*Mr Mouassa Abdennour.*

Résumé.....	i
Dédicace.....	ii
Remerciement .....	iii
Sommaire.....	iv
Liste des tableaux.....	vi
Liste des figures.....	vii
Liste des abréviations.....	viii
Introduction générale .....	01
<b>Chapitre I : État de l’art</b>	
I.1 Introduction .....	02
I.2 Constituants d'un robot .....	02
I.3 La robotique parallèle.....	03
I.4 Différents robots parallèles.....	04
I.4.1 Spécifications de la plate-forme Stewart.....	04
I.4.2 Robot à deux degrés de liberté .....	04
I.4.3 Robots à Trois degrés de liberté .....	05
I.4.4 Robots à Quatre degrés de liberté .....	06
I.4.5 Robots parallèles à six degrés de liberté.....	07
I.4.6 Les robots hybrides .....	07
I.5 Avantages de la plate-forme Stewart.....	08
I.6 Champs d’application.....	08
I.7 Conclusion .....	10
<b>Chapitre II Modèle Cinématique de la Plateforme de Stewart</b>	
II.1 Introduction .....	11
II.2 Cinématique de la plateforme de Stewart 3 ddl .....	12
II.3 Conclusion.....	18
<b>Chapitre III Conception de la Plateforme Stewart à 3 ddl</b>	
III.1 Introduction .....	19
III.2. Description du projet .....	19
III.2.1 Brochage des éléments électroniques .....	20
III.2.2 Organigramme de fonctionnement .....	22
III.3 Partie Hardware.....	23
III.3.1.1 Les actionneurs du robot .....	23
III.3.1.2 Définition d’un servomoteur .....	23
III.3.1.3 Pourquoi un servomoteur MG996 et pas un autre type des moteurs ? .....	23
III.3.2 Le Microcontrôleur.....	24
III.3.2.1 Définition de Arduino Uno .....	24
III.3.2.2 Choix de Microcontrôleur .....	24
III.3.3 Le module Bluetooth HC-05 .....	25
III.3.4 Le Gyroscope .....	26
III.3.5 Tableau du composants utilisés.....	27
III.4 Partie mécanique.....	27
III.4.1 Conception de matériel avec SolidWorks .....	27
III.5 Partie Logiciel (Programmation).....	36
III.5.1 Description du programme avec Matlab.....	36
III.5.1.1 Présentation de l’interface graphique .....	36

III.5.1.2 Instructions interface Matlab-Arduino.....	37
III.5.2 Description de commande avec Smartphone.....	38
III.5.2.1 L'Application remoteXY .....	38
III.5.2.2 Le programme avec Arduino IDE .....	39
III.6 Conclusion .....	39
<b>Chapitre IV Expérimentation et Résultats</b>	
IV.1 Expérimentation et Résultats .....	40
IV.2 Les différentes modes de commande .....	40
IV.2.1 Commande par interface graphique sous Matlab.....	40
IV.2.2 Contrôle par la boîte de commande sous Matlab.....	41
IV.2.3 Commande par Smartphone.....	43
IV.3 conclusion.....	44
<b>Chapitre V : Conclusion et Perspectives</b>	
V.1 Conclusion.....	45
V.2 Perspectives.....	45
Bibliographie.....	46
Annexe.....	47

## Liste des tableaux

---

Tableau I.1 Champs d'application de la plate-forme de Stewart.....	09
Tableau III.01 : matériel utilisé.....	27
Tableaux III.2 Code Matlab.....	37
Tableaux III.3 Instructions Arduino utilisées .....	39
Tableau IV.1 Commande de la plateforme sous Matlab en utilisons le joystick.....	42
Tableau IV.2 Différentes orientations de la plateforme.....	44

Figure I.1 différentes structures de robot.....	02
Figure I.2 Le premier hexapode octaédrique à sa naissance en 1957.....	03
Figure I.3 Le premier h�xapode octa�drique utilis� dans un simulateur de vol, construit autour de 1965 (klaus cappel).....	04
Figure I.4 Robots parall�les � deux degr�s de libert�.....	05
Figure I.5 Exemple d'application utilisant des m�canismes � 3ddl.....	05
Figure I.6 Robot Delta � quatre degr�s de libert�.....	06
Figure I.7 Repr�sentation du Delta � 4.....	06
Figure I.8 Robots parall�les � six degr�s de libert�.....	07
Figure I.9 les robots Hybrides.....	08
Figure II.1 Servomoteur avec le bras.....	12
Figure II.2 mod�le g�om�trique de la plateforme.....	13
Figure II.3 Positions des points d'ancrage de la plateforme.....	14
Figure II.4 mouvements du bras de servo paire et impaire par rapport aux axe X,Y,Z.....	15
Figure II.5 vitesses de pulsion PWM.....	17
Figure II.6 G�n�ration des signaux PWM par ATMEGA 328 pour g�n�rer la sortie sollicit�e par des valeurs de consigne d'entr�e.....	18
Figure III.1 Le sch�ma synoptique de la plateforme.....	19
Figure III.2 Montage sur breadboard (plaque d'essai).....	20
Figure III.3 Sch�ma �lectrique de circuit r�alis�.....	21
Figure III.4 Organigramme de fonctionnement.....	22
Figure III.5 Le servo MG996R.....	24
Figure III.6 Synoptique d'une carte Arduino Uno.....	25
Figure III.7 Module Bluetooth HC-05.....	25
Figure III.8 Les axes de Gyroscope.....	26
Figure III.9 Un gyroscope MEMS.....	26
Figure III.10 Pr�sentation de la position de la plateforme pour les quatre vues.....	27
Figure III.11 Dessin d'ensemble.....	28
Figure III.12 le Servomoteur MG996r.....	29
Figure III.13 La plateforme de Base fixe .....	30
Figure III.14 Le support de la patte.....	31
Figure III.15 La patte.....	32
Figure III.16 Le Bras.....	33
Figure III.17 La plateforme mobile.....	34
Figure III.18 La patte dans le support .....	35
Figure III.12 les points de connexion de bras avec le servomoteur et la patte.....	35
Figure III.20 L'interface graphique utilis�e.....	37
Figure III.21 l'application Remotexy dans le Smartphone.....	38
Figure IV.1 Contr�le de la plateforme sous Interface Matlab (GUI avec Slider).....	40
Figure IV.2 R�ponse de la plateforme � des cordonn�es consignes sous Matlab.....	41
Figure IV.3 L'interface de commande de la plateforme sous Matlab.....	41



**IDE** : environnement de développement intégré

**DDL** : degré de liberté

**SMA** : structure mécanique articulée

**PWM** : Pulse Width Modulation

**MLI** : modulation de largeur d'impulsion

**2T1R** : une rotation (1R) et deux translations (2T)

**3T** : trois translations

**GUI** : Graphical user interface

**MEMS** : système micro-électromécanique

### Introduction Générale

Les robots actuels ont le plus souvent une architecture sérielle qui impose un empilage des axes les uns sur les autres, cette architecture limite la performance dynamique de ces machines du fait de masses élevées à déplacer. Une des idées innovantes est l'architecture parallèle qui allège les masses mobiles. La naissance des premiers prototypes de ce type date du début des années 50. Ces machines cinématiques parallèles nommées PKM (Parallel Kinematics Machine) en anglais ont une masse en mouvement faible ce qui réduit l'inertie permettant d plus grandes performances dynamiques. Les chercheurs et industriels ont donc trouvé un intérêt à développer ce type de robot. Les difficultés de conception de cette architecture résident, en grande partie, dans la garantie d'une homogénéité des performances au sein d'un espace de travail restreint.

Parmi ces robots parallèles on a développé un prototype de la Plateforme de Stewart à 3 ddl commandée par Matlab, Joystick et Smartphone. Après des nombreuses efforts et tentatives qui nous ont aidés à appréhender et corriger nos erreurs, nous avons finalement atteint l'objectif de ce projet qui fonctionne parfaitement dans tous les modes de commande. La plateforme de Stewart peut être utilisée dans plusieurs domaines comme par exemple la simulation de vol, l'industrie, cinéma...etc. Le travail réalisé est multidisciplinaire c'est un ensemble d'électronique, de mécanique, d'automatique et d'informatique. Le mémoire qui résulte de ce travail est décomposé comme suit :

- Le premier chapitre présente l'historique, le type, et les champs d'application des machines parallèles.
- Le deuxième chapitre définit le modèle cinématique d'une plate-forme de Stewart à 3 ddl.
- Le troisième chapitre présente une description du projet qui explique le fonctionnement et le brochage de circuit électronique ensuite il définit le matériel utilisé dans la partie hardware et la partie mécanique qui définit les dimensions des pièces modélisées dans SolidWorks. Finalement la partie programmation montrant les différents programmes utilisés dans Matlab, Arduino IDE et remotexy.
- Le quatrième chapitre présente les résultats des différents modes de commande.
- Dans le cinquième chapitre, on termine par une conclusion sur ce travail et qui donne une vision sur la difficulté de construction et les objectifs réalisés tout en indiquant les possibilités de perspectives de développement de travaux futurs.

**Chapitre I**

**État de l'art**

## I.1 Introduction

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains.

Les premiers robots industriels apparaissent, malgré leur coût élevé, dans le début des années 1970. Ils sont destinés à exécuter certaines tâches répétitives, éprouvantes ou toxiques pour un opérateur humain : peinture ou soudage des carrosseries automobiles. Aujourd'hui, l'évolution de l'électronique et de l'informatique permet de développer des robots plus précis, plus rapides ou avec une meilleure autonomie. Industriels, militaires ou spécialistes chirurgicaux rivalisent d'inventivité pour mettre au point des robots assistants les aidant dans la réalisation de tâches délicates ou dangereuses. Dans le même temps apparaissent des robots à usages domestiques : aspirateur, tondeuses, etc. La robotique possède de nombreux domaines d'application. [1]

## I.2 Constituants d'un robot

Un robot manipulateur est constitué par deux sous-ensembles distincts, un organe terminal (dispositif destiné à manipuler des objets) et une structure mécanique articulée (SMA) (architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations). Les chaînes peuvent être soit ouvertes ou en série (tous les corps ont au plus deux liaisons), arborescentes (au moins l'un des corps a plus de deux liaisons) ou fermées (l'organe terminal est relié à la base du mécanisme par plusieurs chaînes). Ces différentes structures sont montrées dans la Figure I.1.[8]

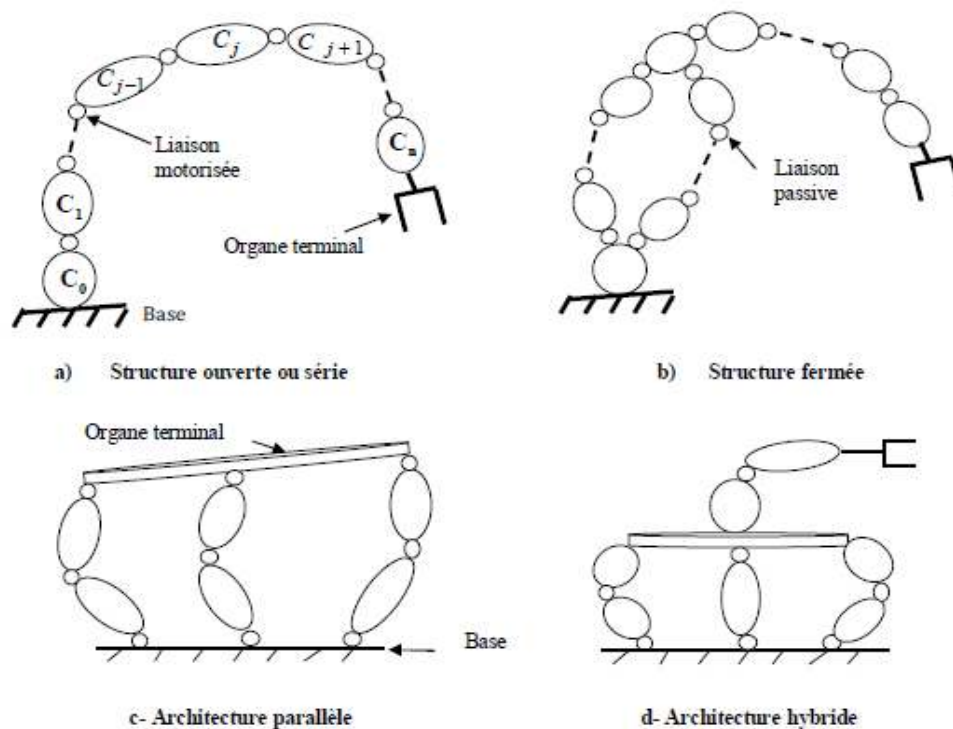


Figure I.1 différentes structures de robot

### I.3 La robotique parallèle

Un robot parallèle est un mécanisme dont l'architecture lui confère des propriétés remarquables dont la définition scientifique, est un mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes.

En quelques sortes, l'organe terminal ou effecteur partie qui agit sur l'environnement, l'outil est relié au bâti par plusieurs bras, chaque bras étant une « chaîne cinématique » (association de plusieurs pièces articulées entre elles). Sa mobilité est donc restreinte, puisqu'elle est limitée par les divers bras ; par contre, cela confère une plus grande résistance et précision, puisque les efforts sont répartis.

Le plus répandu parmi les mécanismes parallèles est connu sous des dénominations comme « hexapode », « plate-forme de Gough-Stewart », « plateforme synergétique ». Il peut se présenter sous différentes formes mais comporte en général 6 actionneurs identiques dont le couplage assure les qualités du système : 6 degrés de liberté dans les déplacements c'est-à-dire x, y, z, tangage, lacet et roulis. [2]

En 1965, le célèbre article de Stewart a été publié dans lequel est décrit essentiellement le croquis d'une variante de l'hexapode utilisée comme simulateur de vol. cet article a eu un impact majeur sur le monde académique et déclenché la recherche sur la robotique parallèle, les hexapodes octaédriques sont souvent appelés plate-forme de Stewart. C'est en essayant d'éliminer la redondance, et sans connaître les travaux du Dr. Gough, que M. Cappel réinvente l'hexapode octaédrique (Figure I.2) [1].

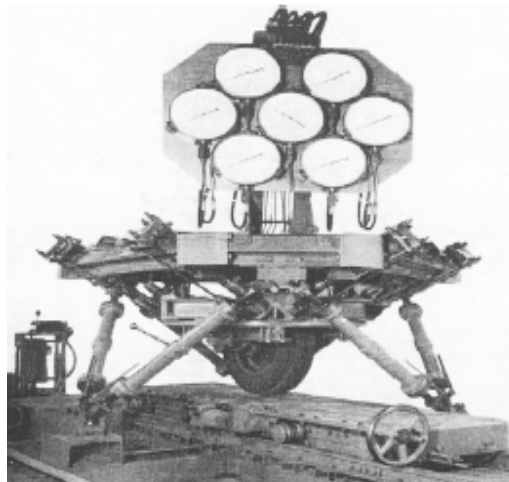


Figure I.2 Le premier hexapode octaédrique à sa naissance en 1957.

Les fabricants de simulateurs de vols ont immédiatement adopté la nouvelle architecture. En 1983, le Pr Kenneth Hunt publie un article qui propose une grande partie des architectures parallèles utilisé dans l'industrie aujourd'hui. Ce qui suit, c'est un boom exponentiel de projets de recherches en robotique parallèle.[4]

Parallèlement, plusieurs compagnies se sont lancées dans la production de robots parallèles [Figure I.3].



Figure I.3 Le premier hexaspode octaédrique utilisé dans un simulateur de vol, construit autour de 1965 (klaus cappel).

### I.4 Différents robots parallèles

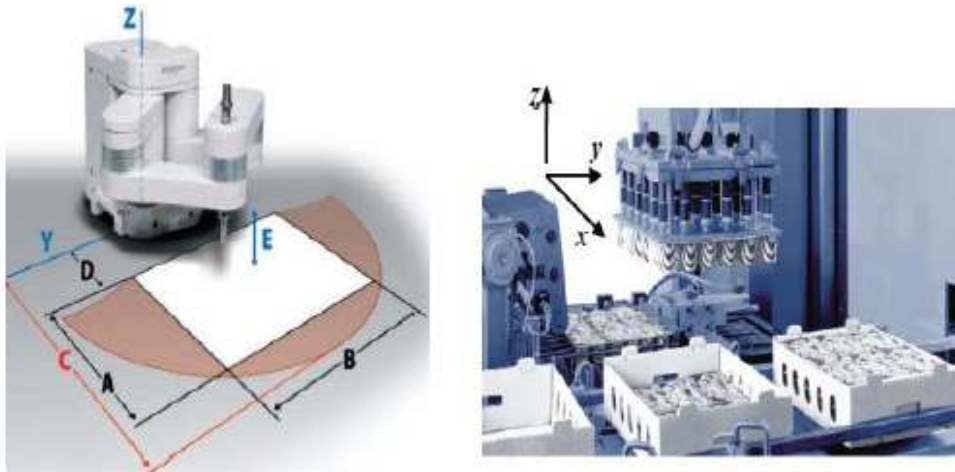
#### I.4.1 Spécifications de la plate-forme Stewart

La plate-forme Stewart est un exemple classique de conception mécanique qui est utilisé pour le contrôle de position. C'est un mécanisme parallèle qui consiste en une plaque supérieure de corps rigide ou plaque mobile reliée à une plaque de base fixe et est défini par au moins trois points fixes sur la base fixée à la terre et relié à trois pattes cinématiques indépendantes. En règle générale, les trois pattes sont relié à la fois la plaque de base et la plaque supérieure par des joints universels situé aux deux extrémités de chaque patte. Les pattes sont conçues avec des extrémités ajustables, permettant à chaque patte d'être de longueur variable. [3]

La position et l'orientation de la plate-forme mobile varient en fonction des longueurs auxquelles les trois pattes sont ajustées. La plate-forme Stewart peut être utilisé pour positionner la plate-forme selon trois degrés de liberté (trois degrés de liberté en rotation, ainsi que trois degrés de translation liberté). En général, la plaque supérieure est de forme triangulaire et tourne 60 degrés autour de la plaque inférieure, permettant à toutes les pattes d'être équidistantes par rapport des autres et à chaque patte pour se déplacer indépendamment des autres.

#### I.4.2 Robot à deux degrés de liberté

Ce sont des robots simples utilisés dans les lignes de production lentes ou intermittentes, seuls deux ddl sont nécessaires : les translations suivant x et z (architecture 2T). Dans l'exemple présenté à la Figure I.4 [5].



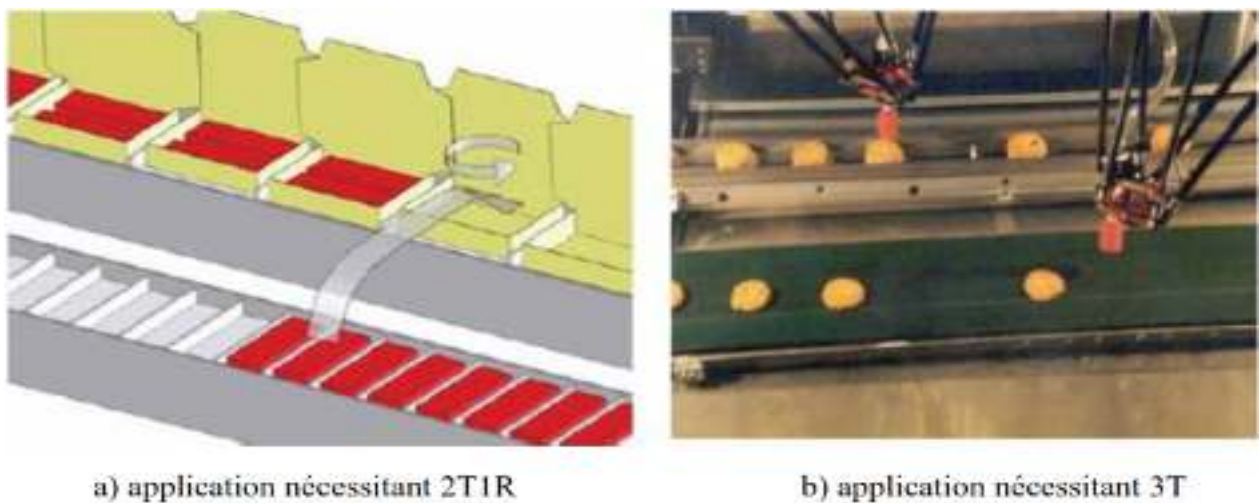
a) Robot Mitsubishi RP1-AH      b) Exemple d'application pour un mécanisme 2T

Figure I.4 Robots parallèles à deux degrés de liberté.

### I.4.3 Robots à Trois degrés de liberté

Il y a différents types des robots à Trois degrés de liberté :

- Les mécanismes 2T1R sont utilisés dans des applications dont les caractéristiques sont similaires au cas précédent (ligne très lente ou intermittente), mais l'objet à manipuler demande à être orienté. Les ddl de ces mécanismes doivent donc être deux translations en  $x$  et  $z$  et une rotation autour de  $z$ . Un exemple de ce type d'application est présenté à la (Figure I.5a).
- Le mécanisme 3T dans le cas d'applications rapides pour les quelles un suivi de convoyeur (appelé « tracking ») est indispensable, l'organe terminal du robot doit pouvoir se déplacer suivant les trois translations  $x$ ,  $y$ ,  $z$ . Un exemple de ce type d'application est présenté à la Figure I.5b . [7]



a) application nécessitant 2T1R

b) application nécessitant 3T

Figure I.5 Exemple d'application utilisant des mécanismes à 3ddl.

Les robots parallèles présentent de nombreux avantages pour les tâches de pick-and-place (prise et pose d'un objet), notamment grâce à leur surprenante vélocité. Ainsi le célèbre robot Delta (Figure.6), avec ses trois degrés de liberté en translation, permet de manipuler un objet dans les trois directions de l'espace avec des cadences de plusieurs Hertz sur de distances de quelques centimètres voire dizaines de millimètres.



Figure I.6 Robot Delta à trois degrés de liberté.

### I.4.4 Robots à Quatre degrés de liberté

Le Delta à quatre degrés de liberté ; afin de répondre aux besoins des applications de manipulation d'objets, l'architecture Delta, originalement pourvue de trois ddl, fut modifiée afin d'y ajouter une quatrième mobilité. Ainsi, la rotation est obtenue en ajoutant une liaison rotoïde à la plateforme dont la rotation est commandée à l'aide d'une chaîne cinématique de type RUPU (Figure I.7) [3].



Figure I.7 Représentation du Delta à 4 degrés de liberté.



### I.4.5 Robots parallèles à six degrés de liberté

Plusieurs formes d'architectures parallèles à six degrés de liberté sont possibles et la plus répandue est la plate-forme de Gough-Stewart (Figure 1.8A). Il s'agit d'un manipulateur à six degrés de liberté, dont la plate-forme est déplacée par l'élongation de six actionneurs linéaires. Son application la plus connue est le simulateur de vol (Figure 1.8B) en raison, entre autres, de la masse élevée de la cabine (jusqu'à 15 000 kg) à laquelle on doit faire subir des mouvements. Les simulateurs de vol sont utilisés pour l'entraînement des pilotes au sol. [2]



A-Plate-forme de Gough-Stewart



B-Simulateur de vol

Figure I.8 Robots parallèles à six degrés de liberté.

### I.4.6 Les robots hybrides

Les manipulateurs à structure ouverte avec des corps montés en série ont des avantages et des inconvénients ; leur avantage est d'avoir un espace de travail assez large, mais ces systèmes ont l'inconvénient d'être relativement lents dans leur réponse et peuvent manquer de rigidité.

Par contre, les manipulateurs parallèles ont un espace de travail réduit mais leur rigidité, la grande vitesse et la précision est plus importante. L'idée des structures hybrides série-parallèle c'est de combiner les avantages des deux structures parallèles et sérielles (Figure I.9A), la structure hybride consiste à disposer en série des modules de structure parallèle (Figure I.9B). Parmi les réalisations de ce genre de structures, on trouve le robot hybride de [Zhang] qui est constitué de deux manipulateurs parallèles mis en série (Figure I.9C, chaque manipulateur a 3 ddl, la plateforme inférieure et supérieure contrôlent respectivement la position et l'orientation de l'effecteur. Ce type de manipulateur permet de découpler la position et l'orientation de l'organe terminal. On trouve une autre structure similaire dans (Figure I.9) [6].

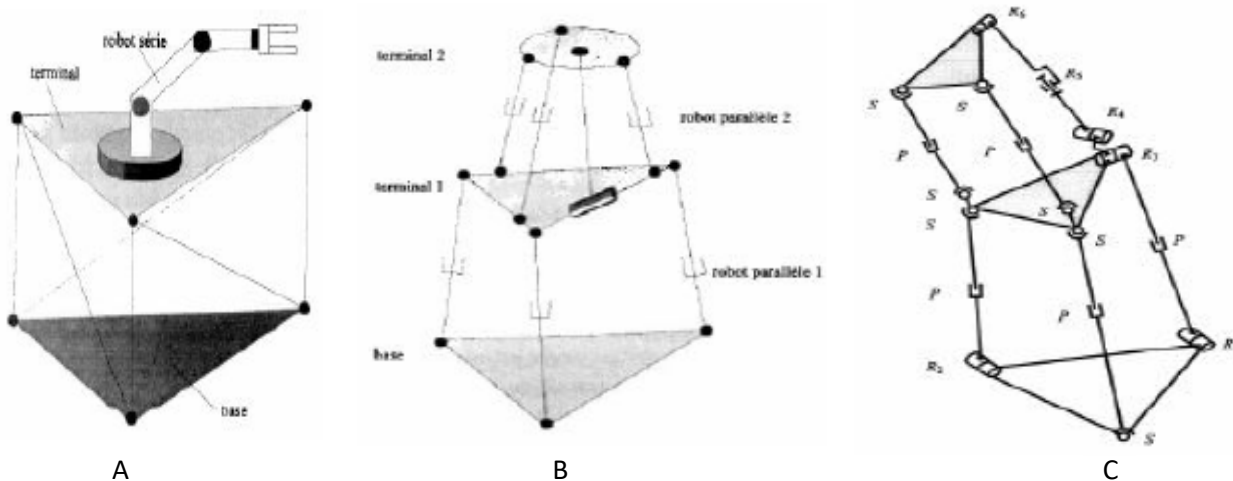


Figure I.9 les robots Hybrides.

### I.5 Avantages de la plate-forme Stewart

La plupart des variantes ont trois pattes actionnées linéairement avec différentes combinaisons de connexions patte-plateforme. C'est en grande partie en raison de la grande amplitude de mouvement du système et de son aptitude positionnement précis. Il fournit une grande quantité de rigidité, permettant au système de la plate-forme Stewart atteindre une position avec très grande certitude. La plate-forme Stewart prend en charge une capacité de charge élevée.

En raison de cette conception, les pattes supportent des forces de compression et de tension, et ne succombera pas à la force de flexion indésirable que l'on retrouve dans d'autres conceptions.

Les trois pieds sont espacés autour de la plaque supérieure et partagent la charge sur l'assiette. Cela diffère des conceptions série, telles que les bras de robot, où la charge est supportée par le moment d'un long bras[9].

### I.6 Champs d'application

La plate-forme Stewart a été conçue en 1965 en tant que simulateur de vol, et il est encore couramment utilisé à cette fin, une grande variété d'applications ont bénéficié de cette conception. Quelques industries utilisent la conception de la plate-forme Stewart dans l'aérospatiale et la défense, l'automobile, le transport et la technologie des machines-outils, la plate-forme Stewart est également utilisée pour le positionnement des antennes paraboliques et des télescopes et dans applications telles que la construction navale, la construction de ponts, transport, et comme plate-forme de forage sur le véhicule rover lunaire.


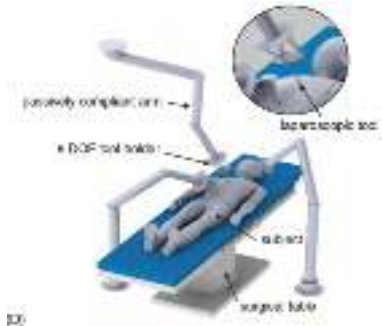






<p>– la technologie des machines-outils.</p> 	<p>- les robots de l'opération médicale.</p> 
<p>– Cinéma 5D.</p> 	<p>– Positionnement antenne parabolique.</p> 
<p>– Test de roues d'avion dans la phase d'atterrissage.</p> 	<p>– Simulateur d'équitation.</p> 
<p>– Simulation de conduite et d'avion [6].</p> 	<p>Transporteurs de conteneur maritimes de navires [11].</p> 

Tableau I.1 : Champs d'application de la plateforme de Stewart.

### **I.7 Conclusion**

Dans ce chapitre nous avons présenté comment les robots parallèles sont développés depuis le premier hexapode octaédrique de Gough-Stewart en 1947 pour tester les pneus en appliquant des charges. En 1965 le célèbre article de Stewart décrit l'hexapode et propose qu'il soit utilisé comme simulateur de vol. Nous avons aussi cité les robots parallèles utilisés dans les différents domaines en les classant selon le nombre de degré de liberté et leurs applications.

**Chapitre II**

**Modèle Cinématique de la  
Plateforme de Stewart**

### II.1 Introduction

Avant de commencer à envoyer des signaux à tous les servos, nous avons pensé qu'il serait intéressant de se familiariser avec les mathématiques de la plate-forme Stewart.

Contrairement aux bras robotiques articulés, la cinématique inverse de la plate-forme Stewart est plus simple que sa cinématique directe. Cela signifie qu'il est plus facile de calculer la longueur des pattes et les paramètres du moteur en fonction de la position souhaitée pour la plate-forme que de calculer l'attitude de la plate-forme pour un ensemble donné de paramètres du moteur. De toute façon, nous voulons vraiment la cinématique inverse, et nous évitons ainsi de résoudre un système de plusieurs équations non linéaires avec plusieurs solutions possibles.

Le problème de la cinématique inverse d'une plate-forme de Stewart peut être divisé en deux étapes:

- 1- Étant donné la position et l'orientation souhaitées pour la plate-forme, à quelle distance se trouve chaque joint sur la plate-forme par rapport à l'articulation de base correspondante.
- 2- choix des angles des servo-moteur, le cas échéant, pour placer chaque joint de plate-forme dans les positions calculées à l'étape précédente.

Le premier problème est facile à résoudre, une fois que vous avez défini les points et les systèmes de coordonnées appropriés, comme dans l'image ci-dessus, les distances entre les joints de base et les joints de plate-forme peuvent être calculées à l'aide d'opérations matricielles simples pour la rotation et la translation.

À savoir, la longueur de chaque patte est calculée en utilisant l'équation de  $l_i$ .  $T$  est le vecteur de translation entre le système de coordonnées de base et le système de coordonnées de la plate-forme (où vous souhaitez déplacer la plate-forme),  $b_i$  et  $p_i$  sont les emplacements des liaisons dans les systèmes de coordonnées de base et de la plate-forme, respectivement, et  $PRB$  est une matrice de rotation décrivant l'angle que la plate-forme tourne.  $PRB$  est également décrit en détail dans l'une des images ci-dessus.

La deuxième partie du problème de la cinématique inverse est un peu plus délicate. Pour chaque servo, étant donné une position de joint de plate-forme  $P$ , par rapport à la position de joint de base  $B$ , et des longueurs fixes pour le servomoteur  $a$  et le pied de support  $s$ , quel est l'angle du bras d'asservissement qui satisfait à la distance  $l$  calculée à l'étape précédente.

Puisque l'angle augmente à mesure qu'on fait varier l'angle de servo-bras de  $-90^\circ$  à  $+90^\circ$  (par rapport au plan de base), on peut résoudre cet angle en effectuant une recherche sur les valeurs d'angle et en recherchant celle qui est plus proche à satisfaire toutes les contraintes de distance.



Figure II.1 Servomoteur avec le bras.

## II.2 Cinématique de la plateforme de Stewart 3 ddl :

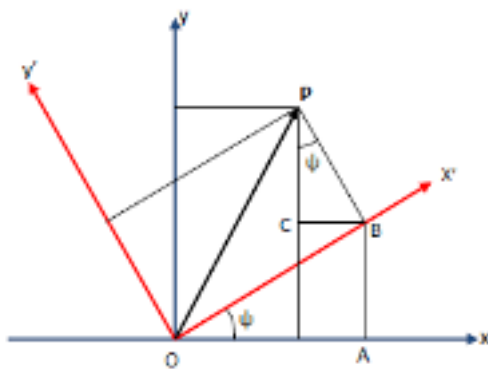
La plate-forme Stewart se compose de 2 cadres rigides reliés par 3 pattes de longueur variable. La base est considérée comme le cadre de référence fixe, avec des axes orthogonaux  $x, y, z$ . La plate-forme a ses propres coordonnées orthogonales  $x', y', z'$ .

La plate-forme dispose de 3 degrés de liberté par rapport à la base. L'origine des coordonnées de la plate-forme peut être définie par 3 déplacements translationnels par rapport à la base, un pour chaque axe [10]. Trois déplacements angulaires définissent ensuite l'orientation de la plate-forme par rapport à la base. Un ensemble d'angles d'Euler sont utilisés dans la séquence suivante :

1. Rotation d'un angle autour de l'axe  $z$  (lacet).
2. Rotation d'un angle  $\theta$  autour de l'axe  $y$  (Pitch).
3. Rotation d'un angle autour de l'axe  $x$  (roulis).

Si nous considérons la première rotation (lacet) autour de l'axe  $z$  :

$$\mathbf{P} = \mathbf{i}'x' + \mathbf{j}'y' + \mathbf{k}'z' = \mathbf{i}x + \mathbf{j}y + \mathbf{k}z$$



$$\begin{aligned} x &= OA - BC \\ &= x' \cos \psi - y' \sin \psi \\ y &= AB + PC \\ &= x' \sin \psi + y' \cos \psi \\ z &= z' \end{aligned}$$

Nous définissons la matrice de rotation  $\mathbf{R}_z(\psi)$  où

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}_z(\psi) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad \text{and} \quad \mathbf{R}_z(\psi) = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

De même, si nous considérons la deuxième rotation  $\theta$  (Pitch) autour de l'axe nous y pouvons montrer :

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix},$$

Et pour la troisième rotation (roulis) autour de l'axe x :

$$\mathbf{R}_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix}$$

La matrice de rotation complète de la plateforme par rapport à la base est ensuite donnée par :

$$\begin{aligned} {}^P\mathbf{R}_B &= \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\varphi) \\ &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \\ &= \begin{pmatrix} \cos \psi \cos \theta & -\sin \psi & \cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & \sin \psi \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \\ &= \begin{pmatrix} \cos \psi \cos \theta & -\sin \psi \cos \varphi + \cos \psi \sin \theta \sin \varphi & \sin \psi \sin \varphi + \cos \psi \sin \theta \cos \varphi \\ \sin \psi \cos \theta & \cos \psi \cos \varphi + \sin \psi \sin \theta \sin \varphi & -\cos \psi \sin \varphi + \sin \psi \sin \theta \cos \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix} \end{aligned} \quad - \quad (1)$$

Considérons maintenant une plate-forme Stewart pour la  $i$ -ème patte :

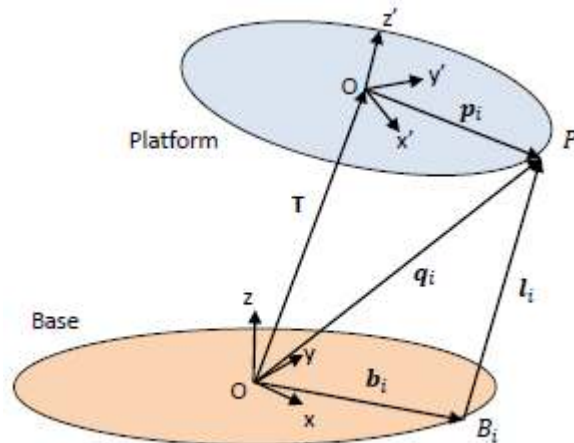


Figure II.2 modèle géométrique de la plateforme

Les coordonnées  $q_i$  du point  $P_i$  d'ancrage par rapport au cadre de référence de base sont données par l'équation :



$$q_i = \mathbf{T} + {}^P\mathbf{R}_B \cdot \mathbf{p}_i \quad - (2)$$

Où  $\mathbf{T}$  est le vecteur de translation, donnant le déplacement linéaire positionnel de l'origine du cadre de plate-forme par rapport au cadre de référence de base, et  $\mathbf{p}_i$  est le vecteur définissant les coordonnées du point d'ancrage en ce qui concerne le cadre de la plateforme.

De même la longueur de la  $i$ -ème patte est donnée par

$$l_i = \|\mathbf{T} + {}^P\mathbf{R}_B \cdot \mathbf{p}_i - \mathbf{b}_i\| \quad - (3)$$

Où  $\mathbf{b}_i$  est le vecteur des coordonnées du point d'ancrage inférieur. Ces 3 équations donnent les longueurs des 3 pattes pour atteindre la position désirée et l'attitude de la plate-forme.

Cette expression représente des équations non linéaires simultanées dont les 3 inconnues représentant la position et l'attitude de la plate-forme [12]. Beaucoup de travail a été fait pour trouver les solutions à ces équations.

Si les longueurs de pattes sont variées par des servos rotatifs, plutôt que des servos linéaires, un autre calcul est nécessaire pour déterminer l'angle de rotation du servo. Chaque combinaison servo/patte peut être représentée comme suit :

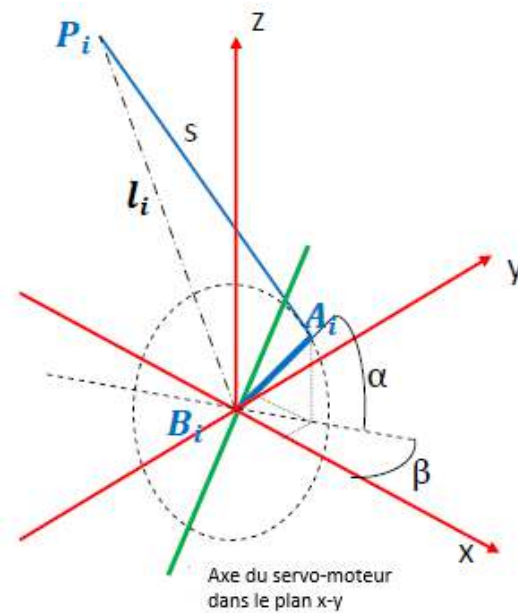


Figure II.3 Positions des points d'ancrage de la plateforme [12]

Où :  $a$  = longueur du bras de commande du servo.

$A_i$  sont les points de l'articulation bras/patte sur le servo avec des coordonnées  $\mathbf{a} = [x_a \ y_a \ z_a]^T$  dans le cadre de base .

$B_i$  sont les points de rotation des bras de servo avec les coordonnées  $\mathbf{b} = [x_b \ y_b \ z_b]^T$  dans le cadre de base .

$P_i$  sont les points les articulations entre les tiges d'opération et la plate-forme, avec des coordonnées  $p = [x_p \ y_p \ z_p]^T$  dans le cadre de la plateforme.

$S$  = longueur de la patte articulée.

$l_i$  = longueur de la patte calculée à partir de

$$l_i = \tau + {}^P R_B \cdot p_i - b_i$$

$\alpha$  = angle du bras d'asservissement horizontal

$\beta$  = angle du plan du bras de servo par rapport à l'axe des abscisses. Notez que l'axe de l'arbre se trouve dans le plan x-y où  $z = 0$

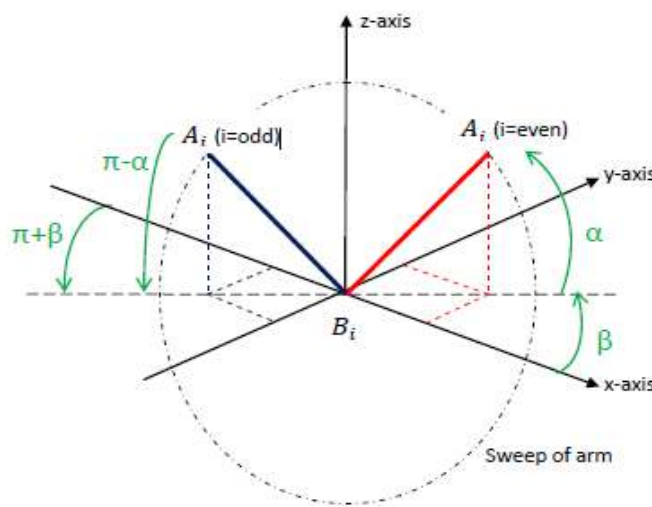


Figure II.4 coordonnées du bras de servo par rapport aux axe X, Y, Z

Un point est contraint d'être sur le bras de servo, donc pour les mêmes pattes, on a :

$$x_a = a \cos \alpha \cos \beta + x_b, \text{ and} \quad - (4)$$

$$y_a = a \cos \alpha \sin \beta + y_b, \text{ and} \quad - (5)$$

$$z_a = a \sin \alpha + z_b \quad - (6)$$

En utilisant le théorème de Pythagore, on obtient l'équation suivante :

$$\begin{aligned} l^2 - (s^2 - a^2) &= 2a \sin \alpha (z_p - z_b) + 2a \cos \alpha \cos \beta (x_p - x_b) + 2a \cos \alpha \sin \beta (y_p - y_b) \\ &= 2a \sin \alpha (z_p - z_b) + 2a \cos \alpha [\cos \beta (x_p - x_b) + \sin \beta (y_p - y_b)] \end{aligned}$$

Qui est une équation de la forme

$$L = M \sin \alpha + N \cos \alpha$$

Nous disposons maintenant de suffisamment d'informations pour calculer les longueurs des "pattes" effectives, et l'angle associé des bras de servo, pour la cinématique inverse de la plate-forme. Mais pour concevoir et mettre en œuvre la plate-forme tripode, nous devons définir quelques constantes afin de définir la gamme de mouvement.

- 1) Nous devons définir la position relative de la plate-forme. Par définition, la plate-forme est à une hauteur au-dessus du cadre de base, et il n'y a aucun autre mouvement de translation ou de rotation. c-à-d:

$$\begin{aligned} \mathbf{q}_i &= \mathbf{T} + {}^P\mathbf{R}_B \cdot \mathbf{p}_i \\ \mathbf{q}_0 &= \begin{bmatrix} 0 \\ 0 \\ h_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \\ &= \begin{bmatrix} x_p \\ y_p \\ h_0 + z_p \end{bmatrix} \end{aligned}$$

De même, nous définirons la position relative où les bras et les tiges de servo sont à angle droit les uns aux autres.

C'est-à-dire

$$\begin{aligned} l^2 &= s^2 + a^2 \\ &= (x_p - x_b)^2 + (y_p - y_b)^2 + (h_0 + z_p - 0)^2 \end{aligned}$$

Ce qui donne

$$h_0 = \sqrt{s^2 + a^2 - (x_p - x_b)^2 - (y_p - y_b)^2} - z_p \quad (10)$$

et puisque la plate-forme est construite symétriquement autour de l'axe z, cette équation donnera le même résultat pour n'importe quelle patte.

- 2) Nous pouvons également calculer l'angle du bras de servo à la position par rapport à la base.

À l'aide de l'équation  $\mathbf{l}_i = \mathbf{T} + {}^P\mathbf{R}_B \cdot \mathbf{p}_i - \mathbf{b}_i$

Et se souvenir de la construction symétrique de la plate-forme. La longueur des pattes dans la position de base est donnée par

$$l_0 = \begin{bmatrix} x_p \\ y_p \\ h_0 + z_p \end{bmatrix} - \begin{bmatrix} x_b \\ y_b \\ 0 \end{bmatrix}$$

$$l_0^2 = (x_p - x_b)^2 + (y_p - y_b)^2 + (h_0 + z_p)^2 \quad - (11)$$

$$\alpha_0 = \sin^{-1} \frac{L_0}{\sqrt{M_0^2 + N_0^2}} - \tan^{-1} \frac{M_0}{N_0} \quad - (12)$$

$$\text{where } L_0 = l^2 - (s^2 - a^2)$$

$$= s^2 + a^2 - (s^2 - a^2) = 2a^2$$

$$M_0 = 2a[\cos \beta (x_p - x_b) + \sin \beta (y_p - y_b)]$$

$$= 2a(x_p - x_b)$$

$$N_0 = 2a(h_0 + z_p)$$

Et l'angle du bras du servomoteur dans le référentiel de base peut être donné par l'équation (9). Par symétrie, pour la patte 2, où  $\beta = 0^\circ$  on a :

Les servos sont montés de sorte que leur point médian soit proche de la position de base, et nous limiterons leur mouvement à  $\pm 90^\circ$ . Les servos sont contrôlés par une impulsion dont la durée définit l'angle du bras.

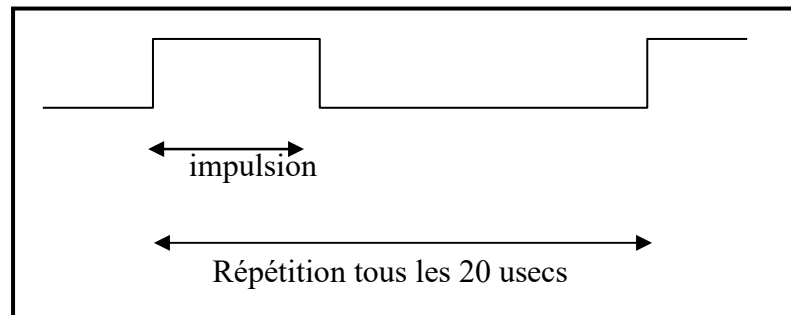


Figure II.5 signaux de commande PWM appliqué aux Servomoteurs

De ce qui précède, les largeurs d'impulsion pour chacun des servos peuvent être obtenues par les équations :

$$\text{Pour } i \text{ pair : } W_i = W_i^0 + (\alpha_i - \alpha_0)r \quad - (13)$$

$$\text{Pour } i \text{ impair : } W_i = W_i^0 - (\alpha_i - \alpha_0)r \quad - (14)$$

Le circuit utilisé pour contrôler la plate-forme est basé sur un microcontrôleur Atmega328, pour contrôler les servomoteurs attachés.

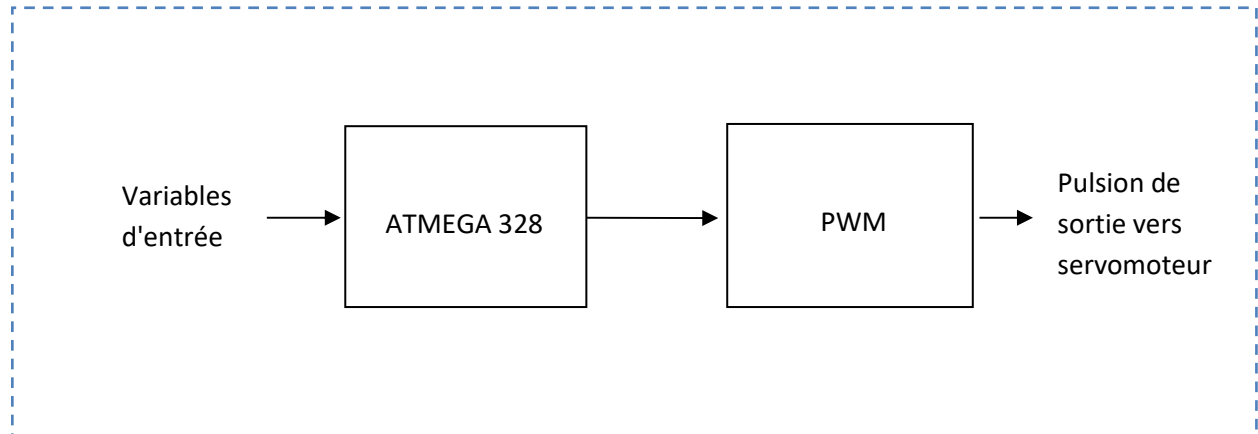


Figure II.6 Génération des signaux PWM par ATMEGA 328 pour générer la sortie sollicitée par des valeurs de consigne d'entrée

La séquence des événements est la suivante :

- 1) Entrez les informations positionnelles pour la plate-forme,  $b_i$ ,  $\pi_i$ ,  $s$ ,  $a$ ,  $C_e$  sont toutes les constantes de la construction de la plateforme.
- 2) Entrez les constantes des servomoteurs  $W_i$  et  $r$
- 3) Calculez les valeurs de l'équation (10) et  $\alpha_0$  de l'équation (12)
- 4) Entrez les variables pour  $(x, y, z, e, \theta, \varphi)$  pour la position de la plate-forme requise
- 5) Calculer la matrice  ${}^P\mathbf{R}_b$  de rotation à partir de l'équation (1)
- 6) Calculez les longueurs effectives des pattes  $l_i$  à partir des équations (3)
- 7) Calculez les angles  $\alpha_i$  requis pour chaque servo de l'équation (9)
- 8) Calculez les largeurs d'impulsion requises pour chaque servo des équations (13) & (14)
- 9) Sortie des valeurs de  $W_i$  la PWM pour piloter les servos
- 10) Revenez à l'étape 4) pour répéter le processus.

### II.3 Conclusion

Dans ce chapitre, le modèle cinématique de la plateforme de Stewart a été présenté. On a calculé la cinématique inverse et les différentes matrices pour déterminer la rotation de la plate-forme selon la position et l'orientation souhaitées, et générer aussi les angles des servomoteurs.

**Chapitre III**

**Conception de la Plateforme**

**Stewart à 3 ddl**

### III.1 Introduction

Les robots parallèles présentent intrinsèquement des potentialités intéressantes, mais choisir une architecture, une géométrie pour les différents éléments mécanique du manipulateur dans le but de satisfaire un cahier des charges définissant les contraintes d'exploitation n'est pas une chose simple, par exemple, l'espace de travail, la précision, la vitesse d'exécution et le poids ...etc. sont difficiles à réaliser ensemble [13].

Dans ce chapitre nous proposons une structure d'un robot parallèle à 3 degrés de liberté. Une description générale de cette structure y est détaillée. L'implémentation de la partie software d'interface, communication et de pilotage de la plateforme est aussi présentée.

Notre système permet de piloter de façon simple et confortable l'ensemble de configuration de cette plate-forme en exploitant les technologies open-hardware Arduino, qui est sans doute le moyen le plus bas prix existant sur le marché. Ce système sera accessible et contrôlable en temps réel de plus l'utilisateur a la possibilité de piloter cette plate-forme de plusieurs manières notamment par Smartphone, par un joystick ou bien à travers un PC relié par liaison série sous interface Matlab.

### III.2. Description du projet

Notre projet est construit autour de la carte Arduino Uno et consiste à faire varier la rotation et l'inclinaison d'une plate-forme Stewart à 3ddl avec un joystick et une interface Matlab ou par le Bluetooth d'un Smartphone équipé d'une application qui utilise le gyroscope.

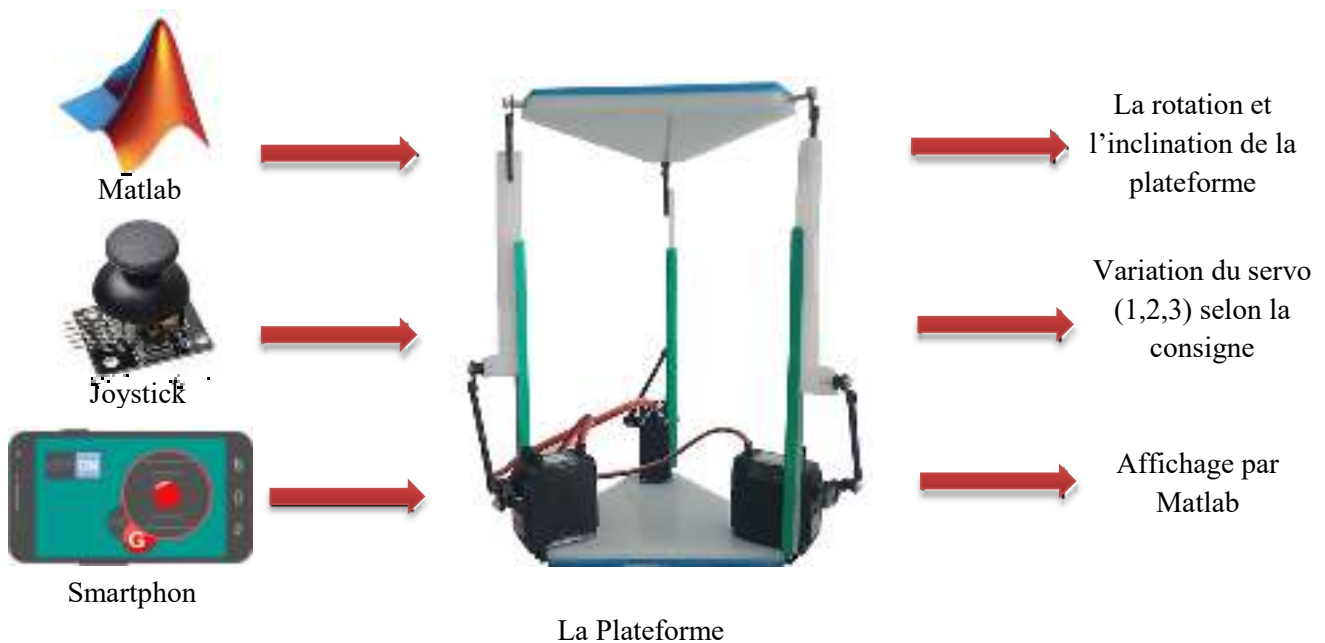


Figure III.1 Le schéma synoptique de la plateforme.

### III.2.1 Brochage des éléments électroniques

La figure suivante montre le circuit de brochages des éléments électroniques qui utilisé par commander et communique avec la plateforme Pour ce faire, on a connecté aux pins 9,10,11 qui sont des pins MLI (PWM) les 3 servomoteur MG996 et les pins Rx et Tx de réception et de transmission sont connecté avec Tx et Rx de module Bluetooth HC-05, les pins VRy et VRx sont connecté avec les ports Analogique A0 et A1, pour les deux boutons poussoirs il sont connecté sur les pins 2 et 3, tous les composent sont alimenté par la carte Arduino.

On a monté les différents composants sur une plaque d'essai dont les connexions sont illustrées par la figure suivante, obtenue en utilisant le logiciel de prototypage Fritzing :

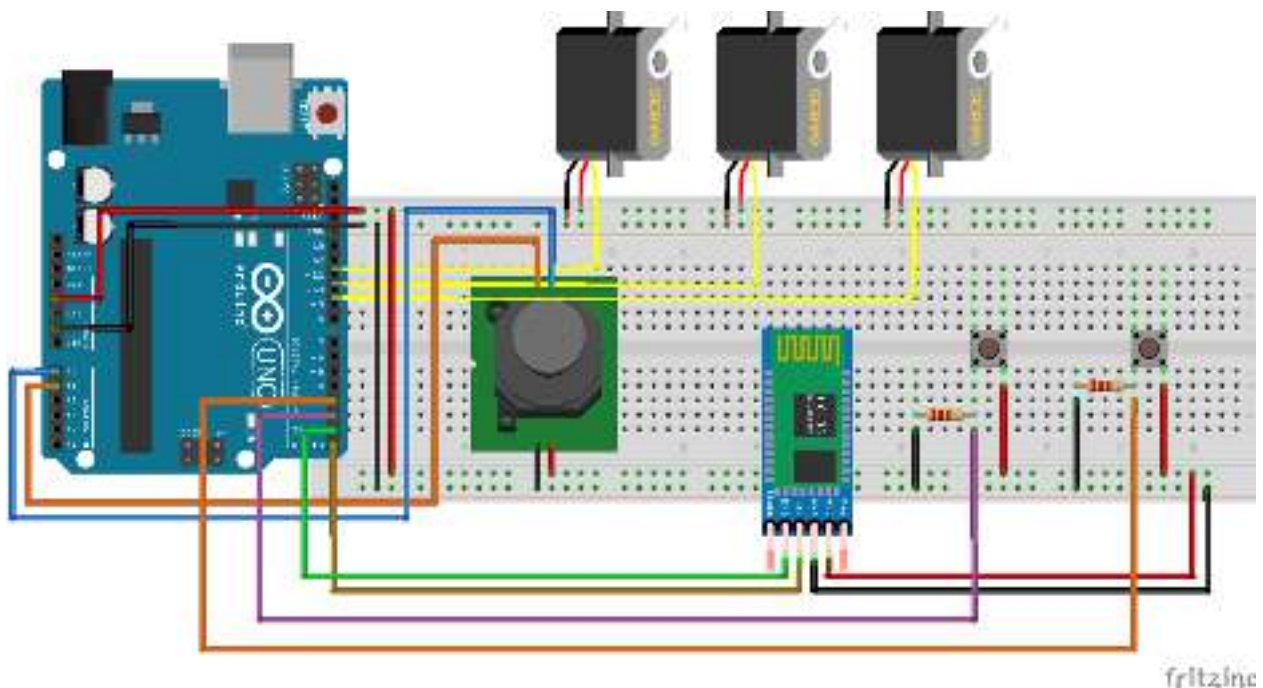


Figure III.2 Montage sur breadboard (plaque d'essai).



La figure suivante montre le schéma électronique utilisé pour la simulation dans le logiciel Proteus. Le joystick et la liaison Bluetooth sont remplacés par 2 potentiomètres car il n'y a ni le joystick ni le Bluetooth dans la bibliothèque de Proteus. Le programme de ce schéma a été implanté sur la carte Arduino Uno et la simulation s'est déroulée sans difficultés en obtenant des résultats comme escompté pour le mouvement de la plate-forme.

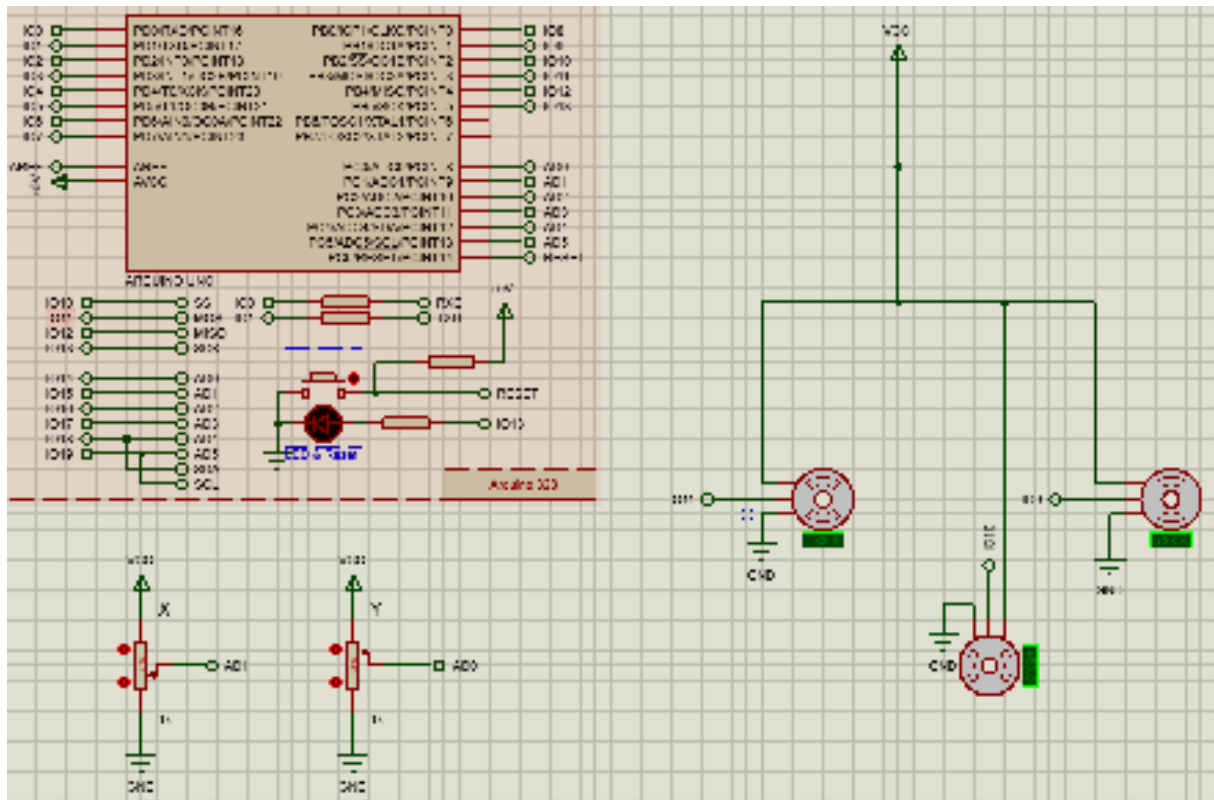


Figure III.3 Schéma électrique de circuit réalisé.

### III.2.2 Organigramme de fonctionnement

L'organigramme suivant montre le choix des tests et les étapes de fonctionnement de cette plateforme.

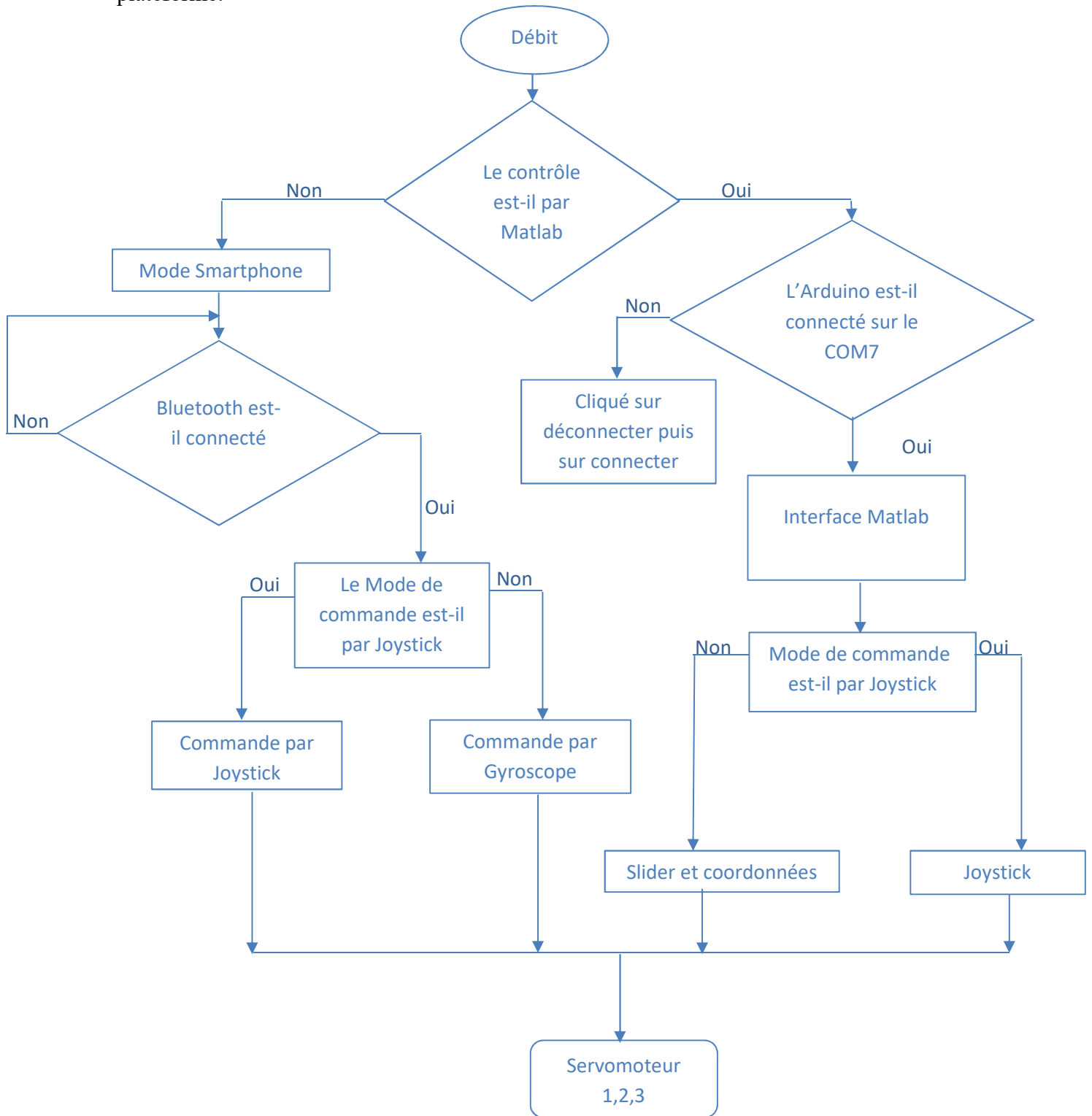


Figure III.4 Organigramme de fonctionnement.

### III.3 Partie Hardware

#### III.3.1.1 Les actionneurs du robot

L'actionnement de mouvements d'articulation est confié à des moteurs qui permettent la réalisation d'un mouvement souhaité pour le système mécanique. Concernant le type de puissance d'entrée, elle est choisie en fonction de la faible puissance du prototype, et en fonction des critères suivants :

- Coût.
- La puissance consommée.
- Le couple délivré.
- La précision de positionnement.

#### III.3.1.2 Définition d'un servomoteur

Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, C'est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique.

#### III.3.1.3 Pourquoi un servomoteur MG996 et pas un autre type des moteurs?

Un servomoteur est capable d'atteindre des positions prédéterminées dans les instructions qui lui on était donné, puis de les maintenir. Le servomoteur a l'avantage d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur (palonnier) respectera la consigne d'instruction que vous lui avez envoyée en son entrée.

Pour un ajustement précis de la position, le moteur et son réglage sont équipés d'un système de mesures qui détermine la position courante.

Le servo numérique MG996R à couple élevé est doté d'un engrenage en métal qui génère un couple de décrochage extrêmement élevé de 10 kg dans un boîtier minuscule. Le MG996R est essentiellement une version améliorée du célèbre servo MG995. Il est doté d'une protection antichoc améliorée et d'un système de commande à circuit intégré et de contrôle IC repensé qui le rend bien plus précis que son prédécesseur. Ce servo standard à couple élevé peut pivoter à environ 180 degrés (90 dans chaque direction). Vous pouvez utiliser n'importe quel code de servo, matériel ou bibliothèque pour contrôler ces servos. D'autant plus qu'il va prendre une petite place.



Figure III.5 Le servo MG996R.

### III.3.2 Le Microcontrôleur

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte et mémoire vive), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, Le microcontrôleur nous permet de créer un système embarqué en tant que composant d'un système plus large, interagissant avec des processus externes en récupérant de l'information via des senseurs et en agissant dessus via des actuateurs. Il peut également être en liaison directe avec l'utilisateur [14].

#### III.3.2.1 Définition de Arduino Uno

C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR.), et un logiciel, véritable environnement de développement intégré (IDE) pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer avec ses logiciels.

#### III.3.2.2 Choix de Microcontrôleur :

Pour les essais pratiques et le prototypage, nous avons choisi d'utiliser la carte Arduino Uno qui contient un Microcontrôleur ATmega328 pour trois raisons :

- Sa capacité de communication avec Simulink/Matlab.
- Programmable avec L'environnement de programmation et Développement (EDI).
- Alimenté directement par PC ou par batterie.
- Ses performances, 32kB de mémoire flash, 14 entrées/sorties dont 6 PWM, vitesse d'horloge 16Mhz et un port de communication USB. Ces performances sont largement suffisantes pour l'application que nous comptons réaliser.

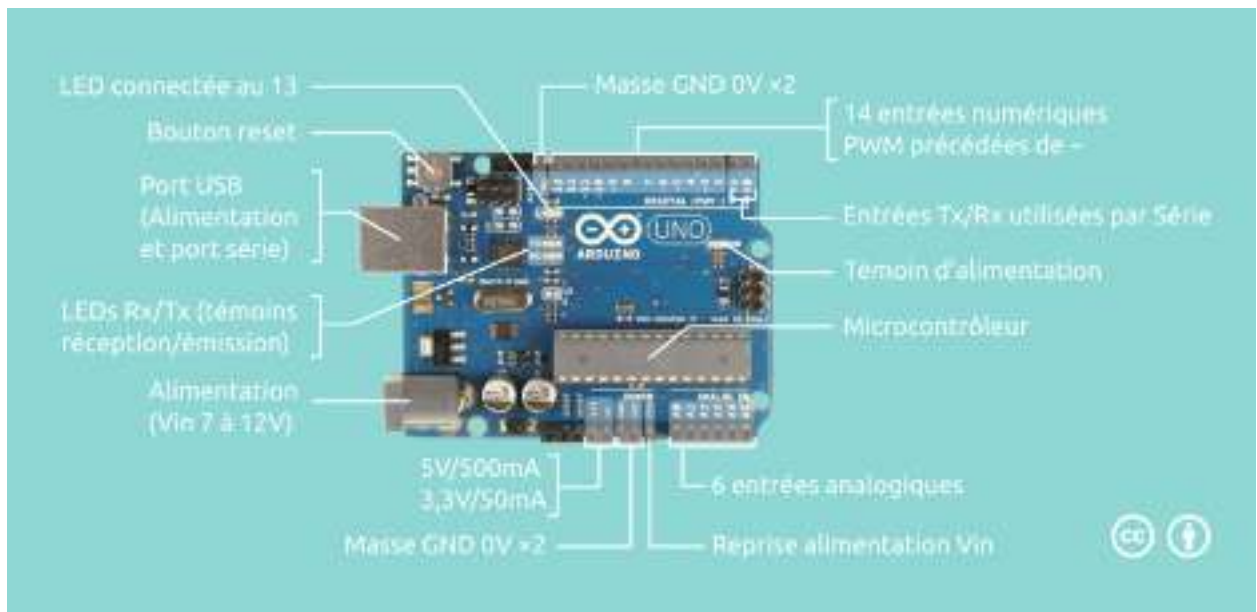


Figure III.6 Synoptique d'une carte Arduino Uno.

### III.3.3 Le module Bluetooth HC-05

Le Bluetooth est un protocole de communication sans fil, Il a subi de nombreuses révisions et évolutions pour atteindre aujourd'hui, Ce protocole est un cousin du Wi-Fi. En effet, ils respectent tous deux une même spécification IEEE et utilisent la même gamme de fréquences : 2.4 GHz (tout comme les téléphones portables et le zigbee par exemple). C'est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est "maître/esclave". Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. le module HC-05 contient un ensemble de broches (TE,RX,TX,GND,VCC,EN) VCC et GND pour l'alimentation (5V), Rx/Tx pour la communication [11].

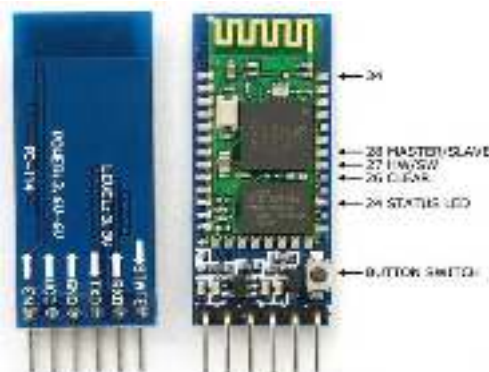


Figure III.7 Module Bluetooth HC-05

### III.3.4 Le Gyroscopie

Un gyroscopie est un appareil utilisé pour mesurer ou conserver l'orientation et la vitesse angulaire. C'est une roue ou un disque dans lequel l'axe de rotation (axe de rotation) est libre d'assumer toute orientation par lui-même. Lors de la rotation, l'orientation de cet axe n'est pas affectée par l'inclinaison ou la rotation du support, en fonction de la conservation du moment cinétique.

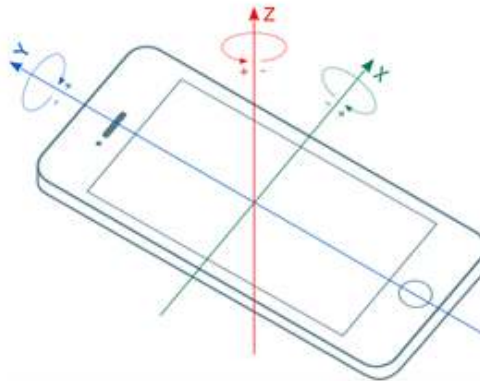


Figure III.8 Les axes de Gyroscopie.

### Les gyroscopes MEMS

Un gyroscopie à système micro-électromécanique (MEMS) est un gyroscopie miniaturisé trouvé dans des dispositifs électroniques. Il reprend l'idée du pendule de Foucault et utilise un élément vibrant. La version MEMS des gyroscopes est extrêmement petite et légère. Elle est intégrée dans plusieurs applications de Smartphone [12].

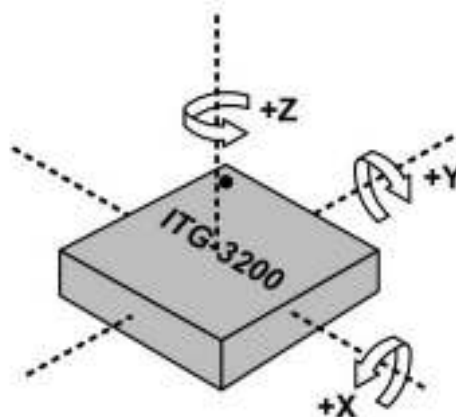


Figure III.9 Un gyroscopie MEMS.

### III.3.5 Tableau du composants utilisés

Composent	Quantité	Prix
Arduino Uno rev 3	1	2900 DA
Servo moteur	3	1800 DA *3
Joystick	1	500 DA
Pushbutton	2	--
Bluetooth HC-05	1	1200 DA
Résistance 1KΩ	2	--
Prix total		10000 DA

Tableau III.01 : matériel utilisé.

### III.4 Partie mécanique

#### III.4.1 Conception de matériel avec SolidWorks

Pour la conception mécanique de la plate-forme et la simulation des différentes pièces on a utilisé SolidWorks afin de modéliser l'assemblage de différentes parties mécaniques de cette structure comme il est montré dans la figure ci-dessous; les dimensions physiques réelles de la structure y sont aussi représentées.

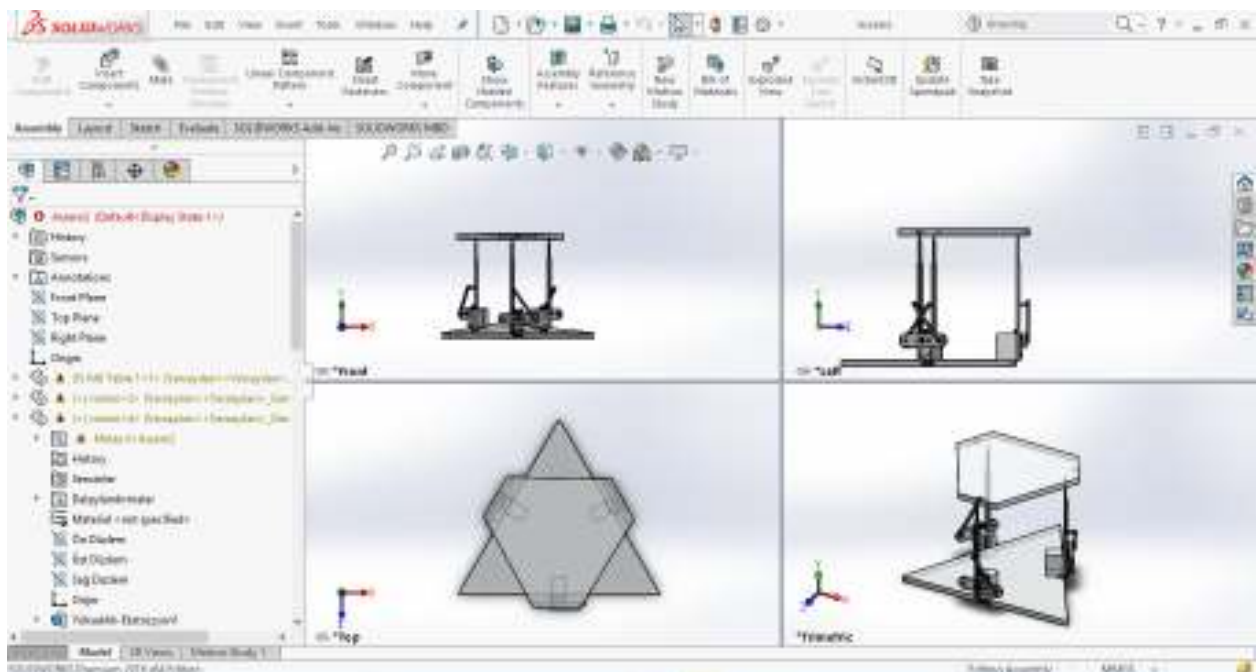


Figure III.10 Présentation de la position de la plateforme pour les quatre vues.

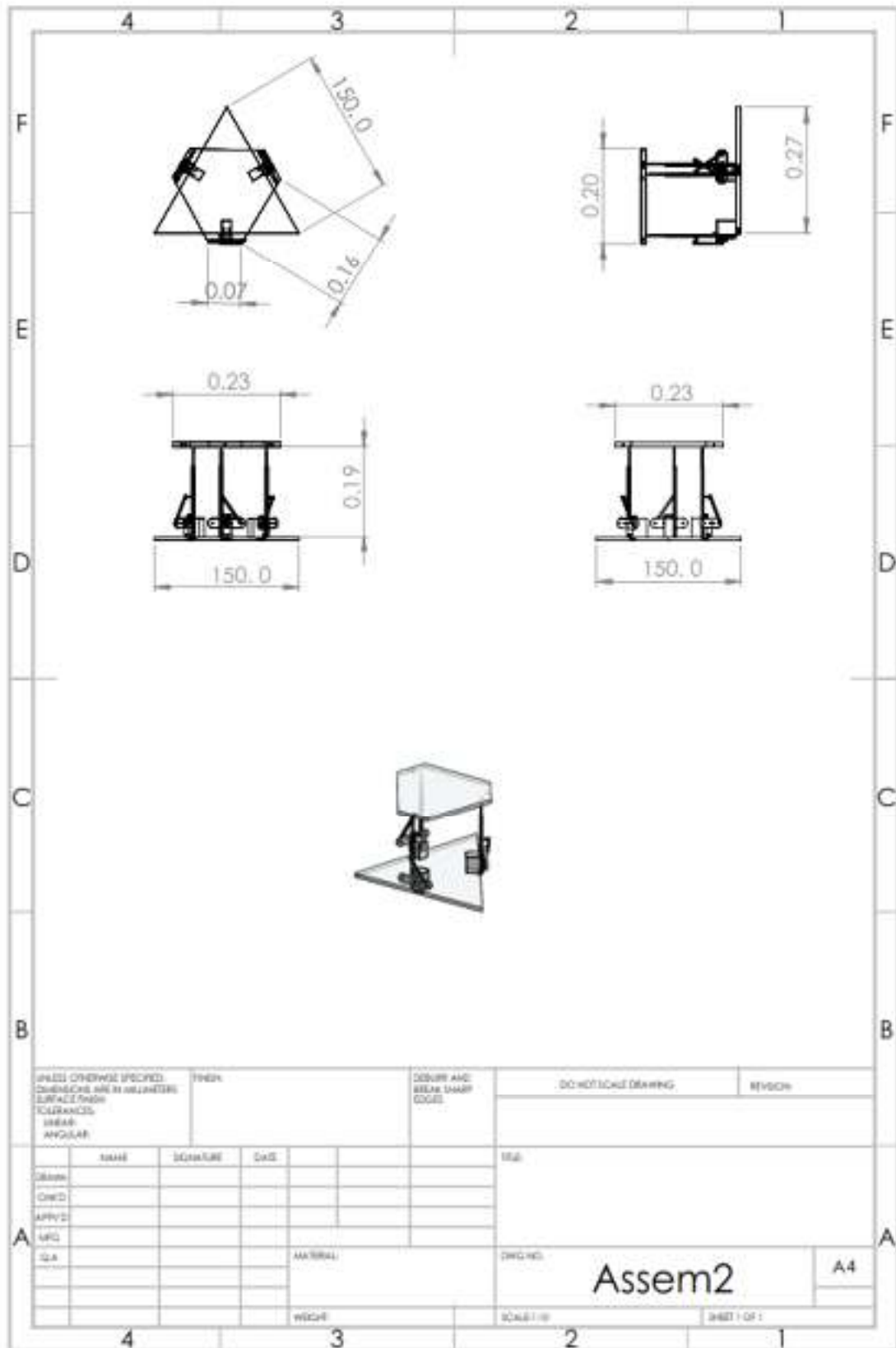


Figure III.11 Dessin d'ensemble.



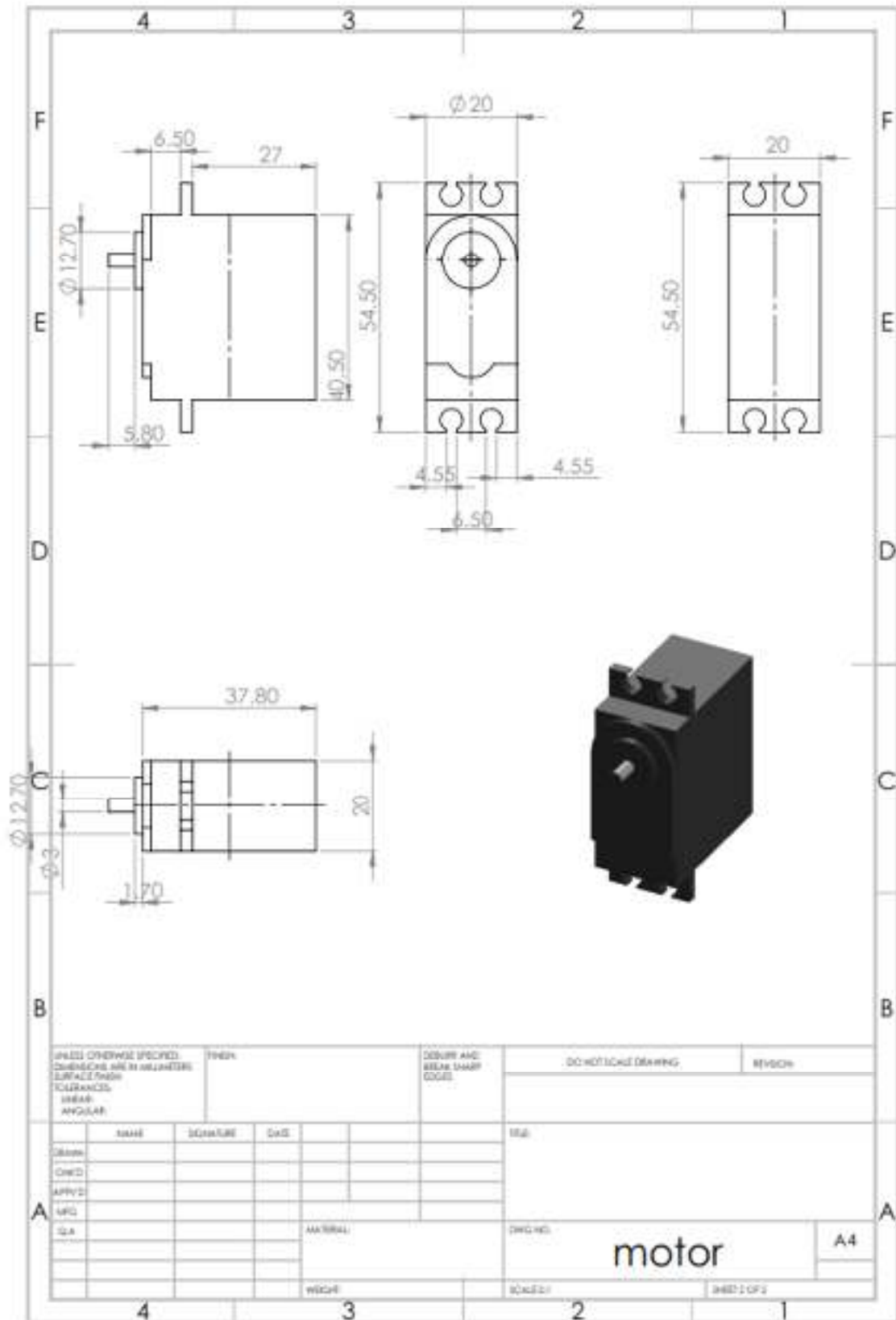


Figure III.12 le Servomoteur MG996r.

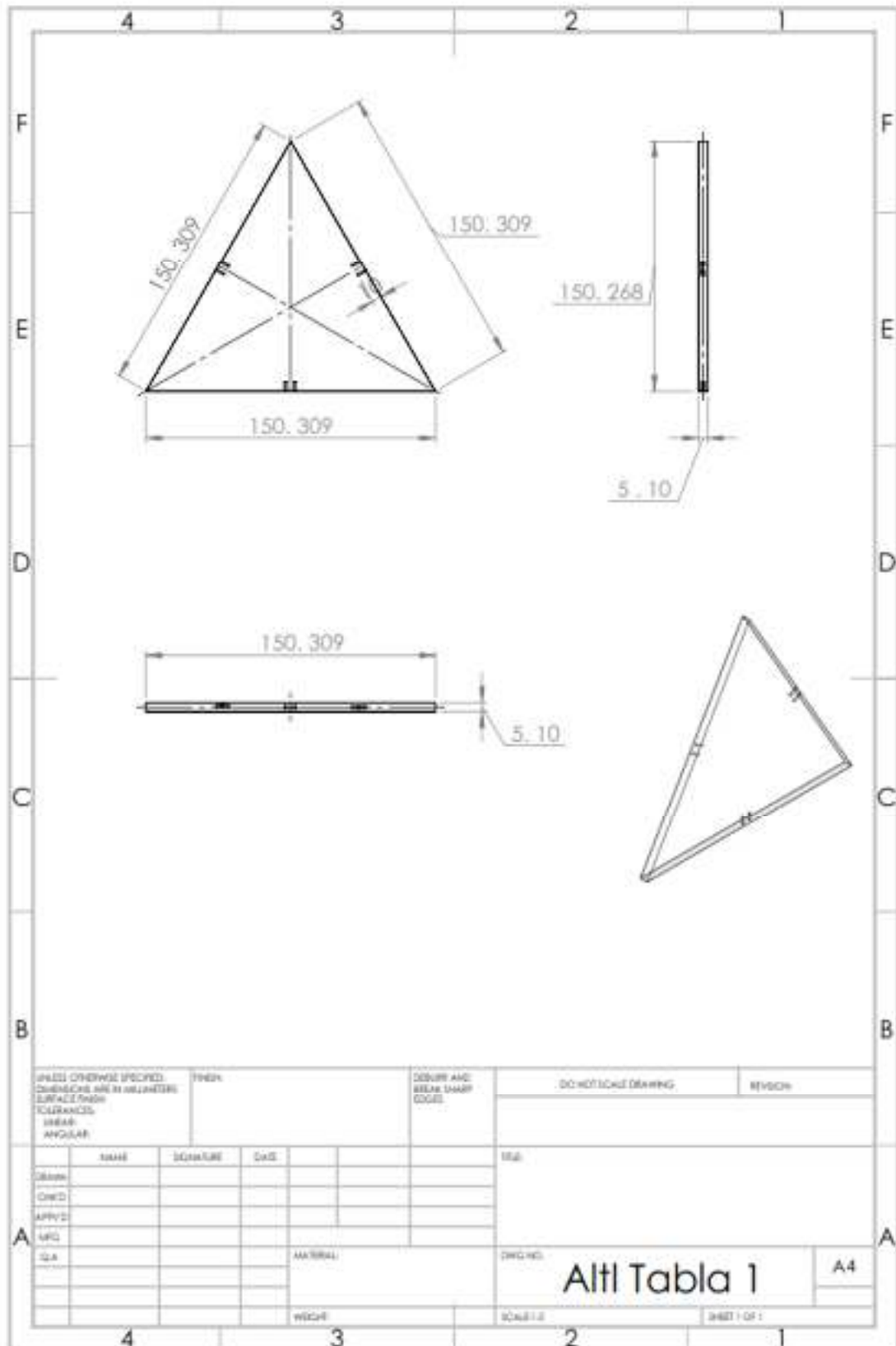


Figure III.13 La plateforme de base fixe.

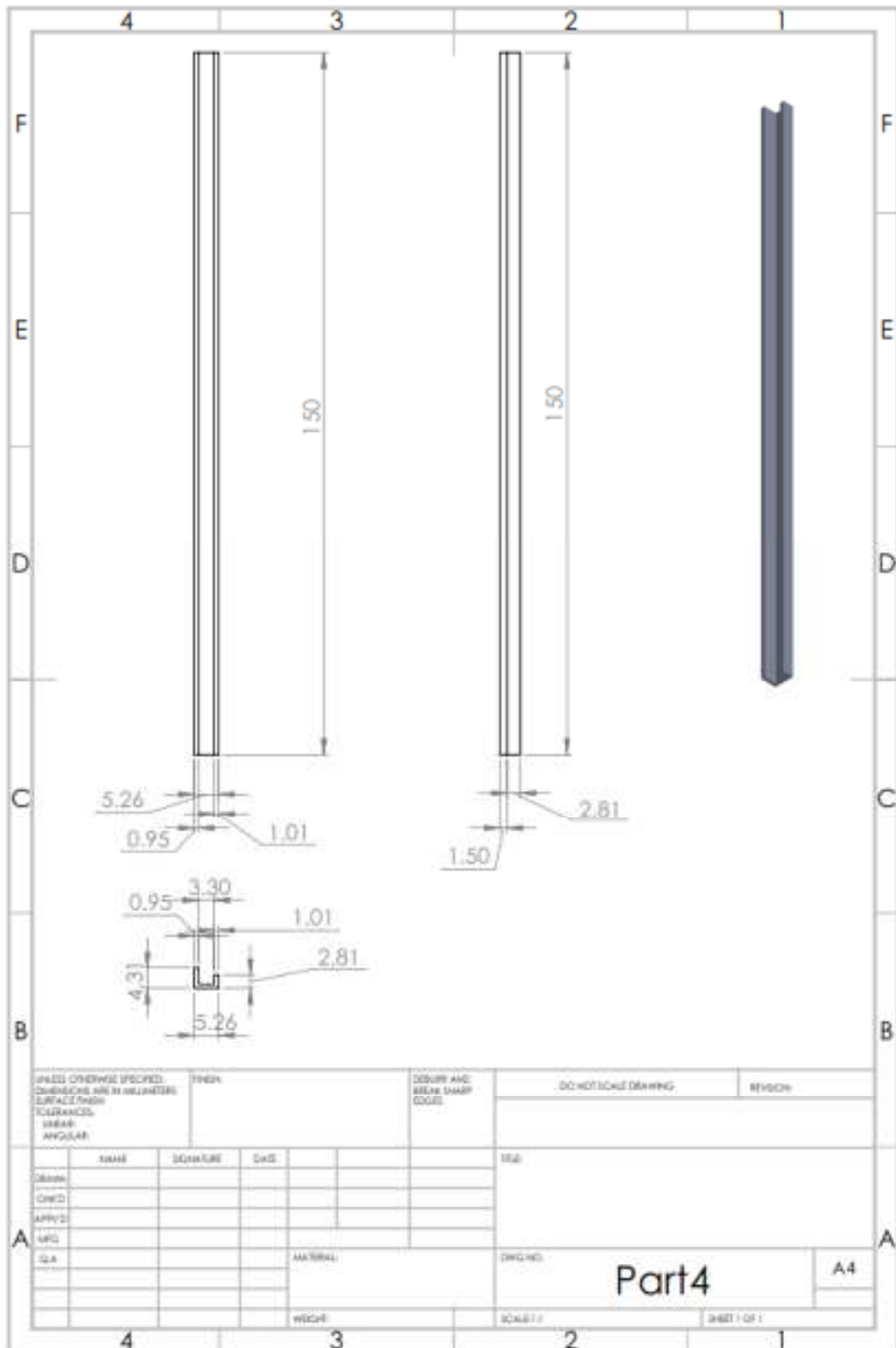


Figure III.14 Le support de la patte

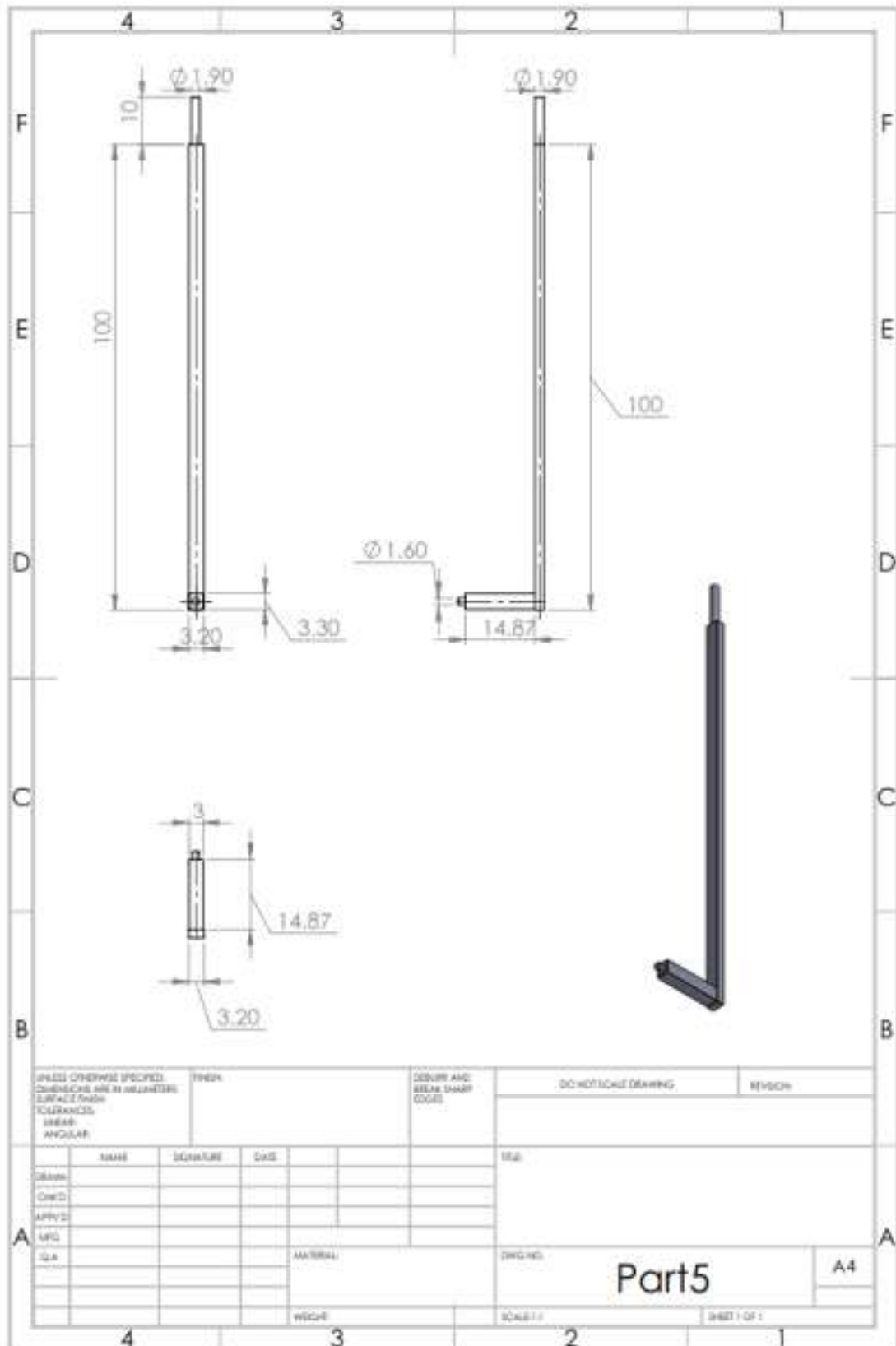


Figure III.15 La patte.

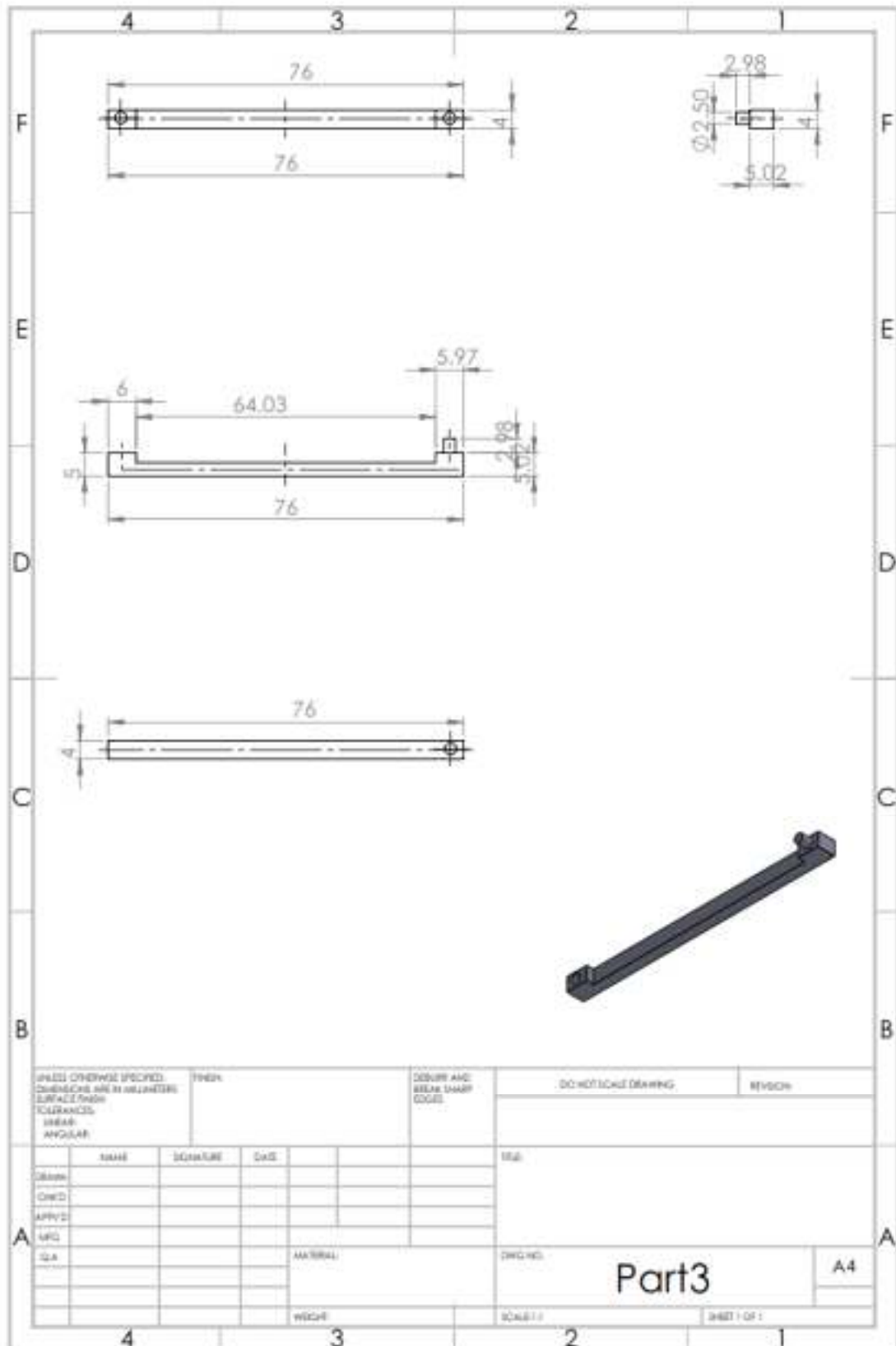


Figure III.16 Le bras.

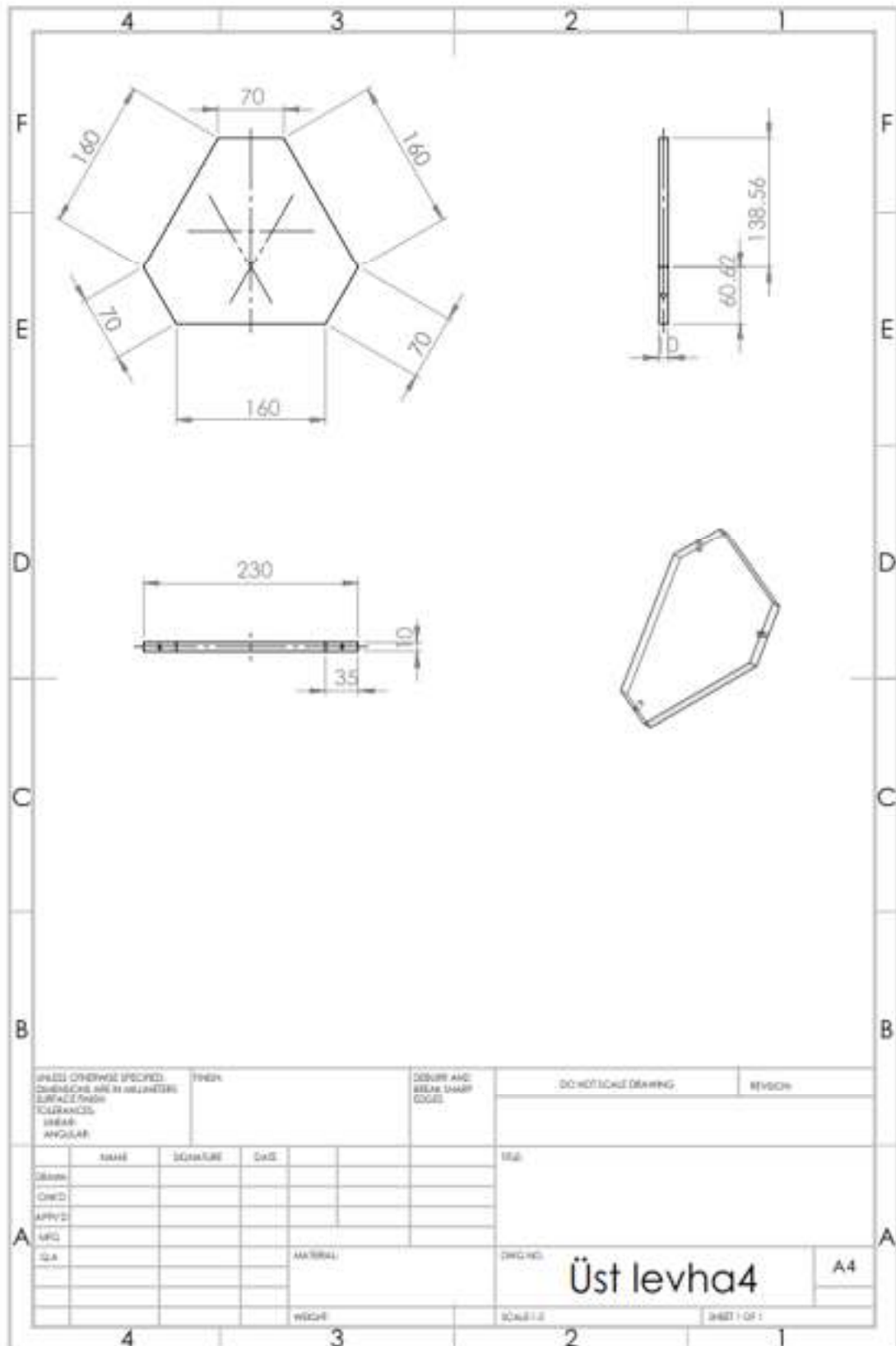


Figure III.17 La plateforme mobile.

Dans la figure ci-dessous est indiqué comment les différentes pièces sont assemblées.



Figure III.18 La patte dans le support.

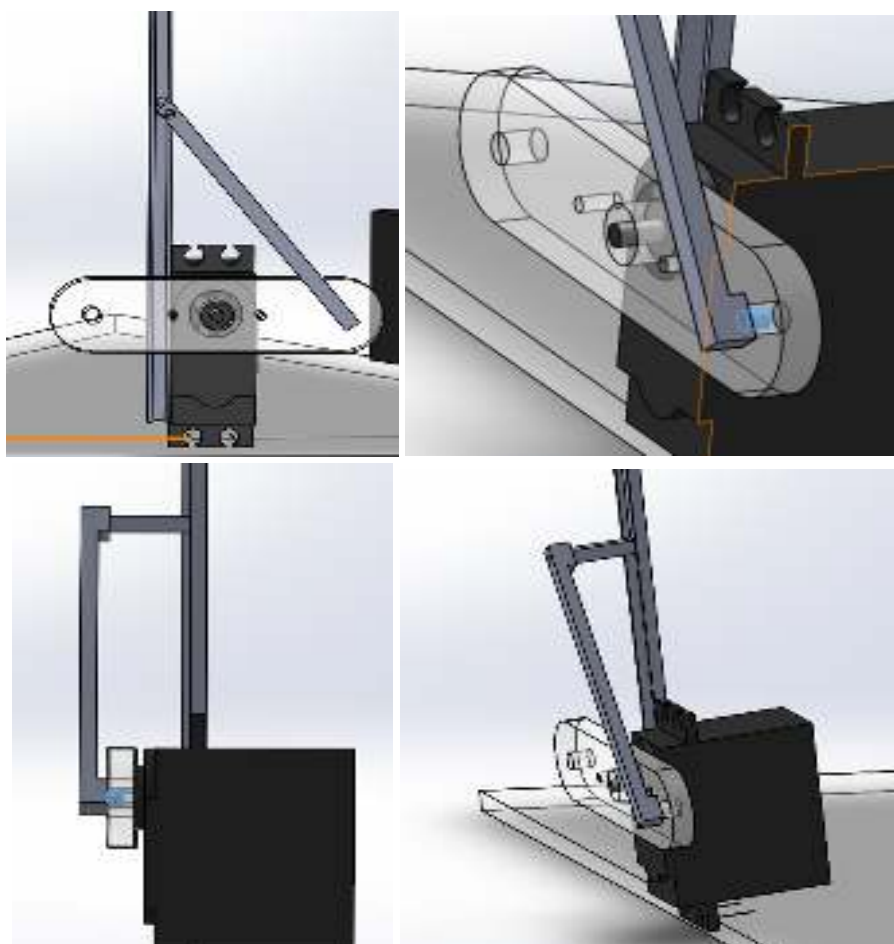


Figure III.19 les points de connexion du bras avec le servomoteur et la patte.

### III.5 Partie Logiciel (Programmation)

Cette partie sera consacré à la partie programmation de notre système. Dans ce qui suit, nous allons présenter le programme utilisé pour les différents modes de la commande de la plateforme ; les parties composantes du logiciel développé, la commande par Matlab, la commande par Smartphone, et par joystick.

#### III.5.1 Description du programme avec Matlab

MATLAB est un langage de développement informatique particulièrement dédié aux applications scientifiques. a cause de ces capacités nous l'avons utilisé pour programmer notre plateforme de Stewart a 3ddl, d'abord il y a deux modes de commande le premier est par le joystick et le deuxième par l'interface graphique, après il faut installer le support package de Arduino sur Matlab pour qu'ils puissent communiquer entre eux, ensuite créer une interface graphique de pilotage en boucle ouvert de la plateforme .

On remarque que le programme est subdivisé en 4 blocs :

- Bloc 1 : configuration des ports (input, output, le COM d'Arduino)
- Bloc 2 : Programme de chaque type de commande (joystick ou avec l'interface graphique)
- Bloc 3 : initialisation de position des servomoteurs.
- Bloc 4 : appliquer la commande sur la sortie.

##### III.5.1.1 Présentation de l'interface graphique

L'interface graphique, appelée communément la GUI (Graphical User Interface), est le lien entre l'utilisateur et la machine.

De nos jours, les outils de réalisation d'interface graphiques ne manquent pas, comme le DELPHI, MATLAB, LABVIEW, VISUEL BASIC, mais il faut bien savoir choisir le logiciel avec quoi on va travailler. En ce qui nous concerne, nous avons choisi d'utiliser « Matlab » vue notre familiarité avec et l'existence d'une bibliothèque d'outils d'interfaçage avec Arduino.



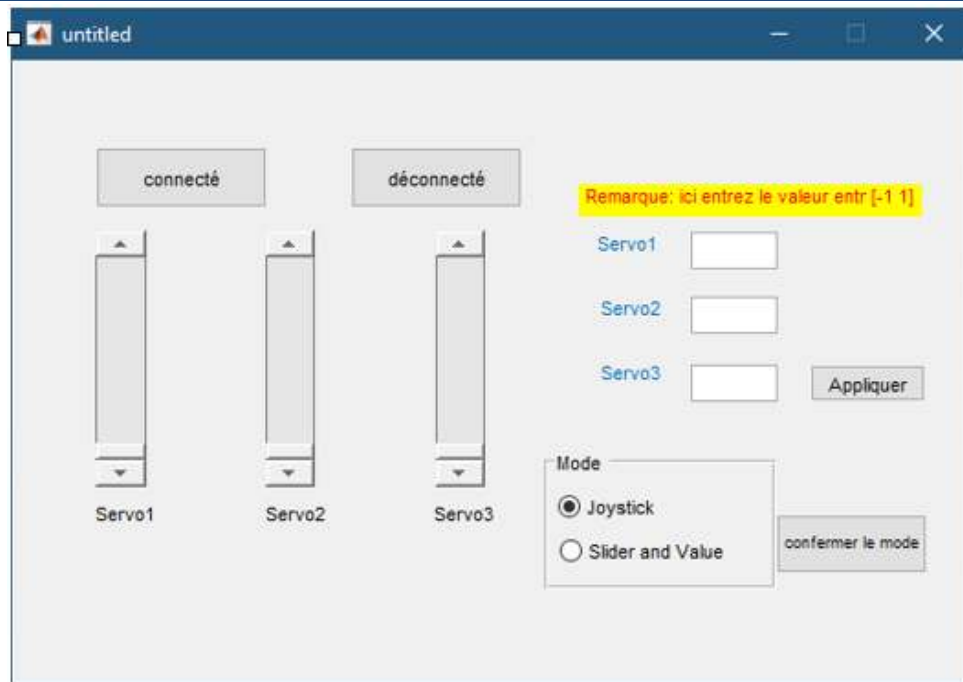


Figure III.20 L'interface graphique utiliser dans Matlab.

Cette interface a pour but de varier les positions réelles des actionneurs, il y a 2 boutons pour connecter et déconnecter la carte et une fenêtre cadre pour le choix du mode du pilotage (Joystick ou Slider et la valeur affectée), pour varier la position du servo il suffit juste de faire varier le Slider de chaque servomoteur en entrant une valeur entre [-1 1] puis cliquer sur le bouton 'appliquer'.

### III.5.1.2 Instructions interface Matlab-Arduino

Instruction	Fonctionnement
<code>global a;</code>	Déclaration de variable globale.
<code>a=arduino('COM7');</code>	Connecté la carte Arduino.
<code>a.pinMode(9, 'output');</code>	Configuré le port comme output.
<code>servoAttach(a, 9);</code>	Attaché le servomoteur aux pin 10.
<code>servoWrite(a, 9, Y1)</code>	Ecrire dans le servomoteur.
<code>pause(0.02);</code>	delay de 20ms .
<code>slider=get(hObject, 'Value')</code>	Lire la valeur obtenue de Slider.
<code>set(handles.text14, 'String', num2str(Y));</code>	Transformé la valeur on String.
<code>Y1=round(Y)</code>	Rondir le nombre Y un nombre naturel.
<code>edt1=str2double(get(handles.edit1, 'String'))</code>	Transformé la valeur obtenue de edit1 on type double.
<code>blah=get(handles.mode, 'SelectedObject')</code>	Obtenue le mode selection.
<code>Selection=get(blah, 'String')</code>	Transformé la variable en String.
<code>selection2=strcmp(Selection, 'Joystick')</code>	Compare le String avec Joystique.
<code>if else end</code>	La fonction if else.
<code>While(1) end</code>	Function while.
<code>x_pos = analogRead(a, 1);</code>	Lire la valeur analogique de pin A0
<code>Xr1=mapfun(x_pos, 0, 1023, 0, 180);</code>	Remappé un nombre d'une plage à une autre.
<code>delete(instrfind({'COM7'}, {'DEMO'}));</code>	Déconnecté l'Arduino.

Tableaux III.2 Code Matlab.

### III.5.2 Description de commande avec Smartphone

Pour connecter le Smartphone avec l'Arduino il y a plusieurs méthodes comme par exemple le WIFI, le Bluetooth, Ethernet Shield, IR...etc. Dans ce projet on a utilisé le module Bluetooth HC-05 qui va nous permettre de commander à distance la plateforme sous une application Android.

#### III.5.2.1 L'Application remoteXY

Remotexy est une application Android qui permet de créer et d'utiliser une interface utilisateur graphique mobile pour les cartes Arduino à contrôler via un smartphone ou une tablette. Pour configurer une application il faut aller au site web remotexy.com choisir l'affichage et la décoration selon leur combinaison en plaçant les éléments à l'écran à l'aide de l'éditeur en ligne. Après le développement de l'interface graphique, le code source du microcontrôleur qui implémente votre interface. Le code source fournit une structure pour l'interaction entre votre programme, les contrôles et l'affichage. Ainsi, on peut facilement implanter le source code sur la carte Arduino [13].



Figure III.21 l'application Remotexy dans le Smartphone.

#### III.5.2.2 Le programme avec Arduino IDE

L'environnement de programmation Arduino est en fait un système de développement intégré (EDI) dédié au langage Arduino. Dans notre cas le sketch obtenu à travers le site web de l'application remotexy est utilisé pour recevoir les données de consigne d'entrée - sortie (valeurs numériques) via le module Bluetooth. Les valeurs de consigne sont appliquées aux équations pour déterminer par programmation les angles de commande des Servomoteurs en sortie.

Instruction	Fonctionnement
<code>#include &lt;RemoteXY.h&gt;</code>	Bibliothèque de l'application remotexy.
<code>uint8_t RemoteXY_CONF[]</code>	Configuration de toutes les variables d'interface.
<code>int8_t joystick_1_x;</code>	Coordonné x de la position de joystick.
<code>#include &lt;Servo.h&gt;</code>	Bibliothèque de servo moteur.
<code>int SERVO_X_Center = 1500;</code>	Initialisation de position de servo moteur.
<code>Servo SERVO_X;</code>	Le nome de servo moteur.
<code>SERVO_Z.attach(9);</code>	Attaché le servo au pin 9.
<code>Serial.begin(9600);</code>	Définit le débit de données en bits par seconde (bauds) pour la transmission de données en série.
<code>SERVO_X.writeMicroseconds();</code>	Ecrire dans le servo moteur en microseconde.
<code>Serial.print("X=");</code>	Instruction d'affichage.

Tableaux III.3 Instructions Arduino utilisées.

### III.6 Conclusion

Une description détaillée de notre projet est présentée dans ce chapitre. Un très grand effort a été nécessaire pour arriver à concrétiser ce projet. Il fallait tenir en considération et résoudre en même temps les difficultés mécaniques de la plateforme, la conception et la programmation des différentes parties électroniques sans oublier l'interfaçage entre elles, et ceci dans une période assez courte dédiée à notre projet.

# **Chapitre IV**

## **Expérimentation et Résultats**

### IV.1 Introduction

L'utilité des tests c'est de connaître les capacités techniques de la plateforme de Stewart à 3 ddl à suivre les mouvements sollicités par l'utilisateur soit par Matlab via une interface graphique, soit par un joystick ou soit par un smartphone. Ce projet est un projet multidisciplinaire : il contient de la mécanique, de l'électronique, de la communication sans fil et de commande des systèmes dynamiques. Après un très grand effort de mise au point et d'ajustement des différents modules, on a abouti à faire fonctionner parfaitement la plateforme dans tous les cas de figures et scénario de modes de commande.

### IV.2 Les différentes modes de commande

#### IV.2.1 Pilotage de la plateforme par interface graphique

Les deux photos ci-dessous montrent l'utilisation de l'interface graphique pour commander la plateforme selon les deux modes :

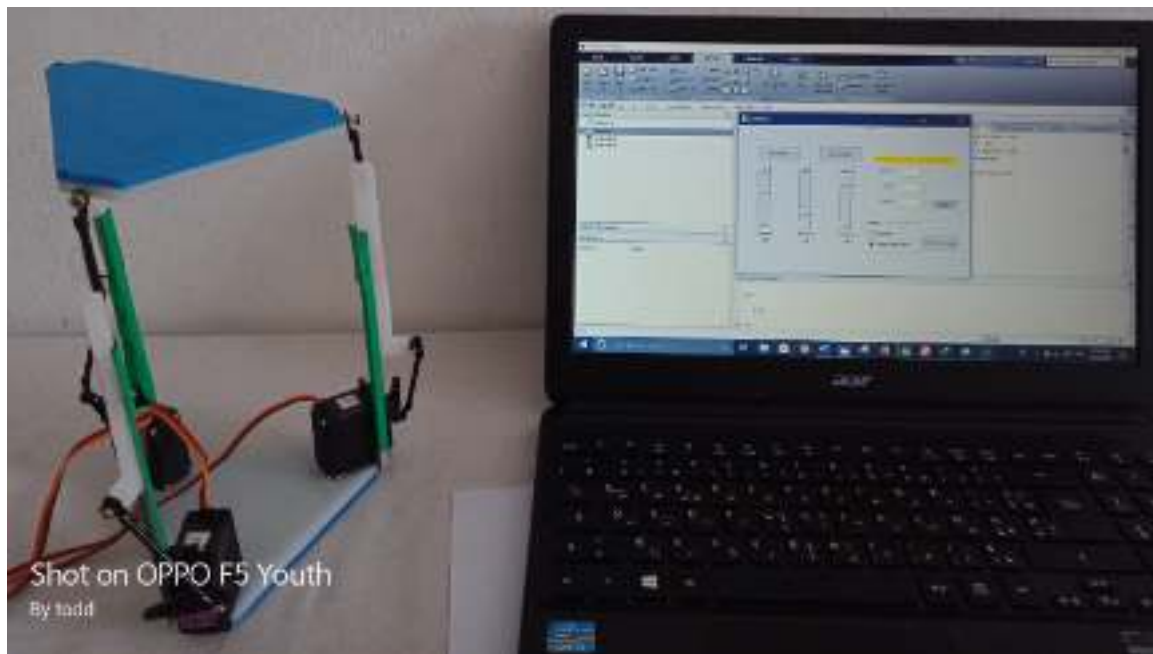


Figure IV.1 Contrôle de la plateforme sous Interface Matlab (GUI avec Slider).

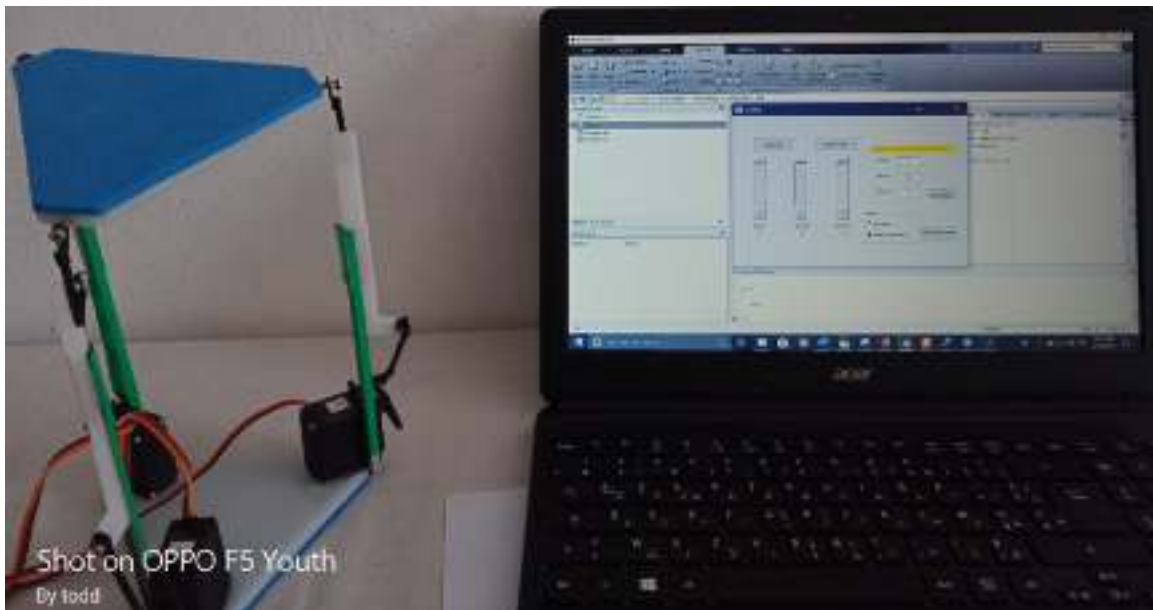


Figure IV.2 Réponse de la plateforme à des coordonnées consignées sous Matlab.

### IV.2.2 Contrôle par boîtier de commande sous Matlab

Un boîtier de commande composé d'un joystick et deux boutons poussoirs sont utilisés pour commander la plateforme comme il est montré dans la photo ci-dessous :



Figure IV.3 Le boîtier de commande de la plateforme sous Matlab.

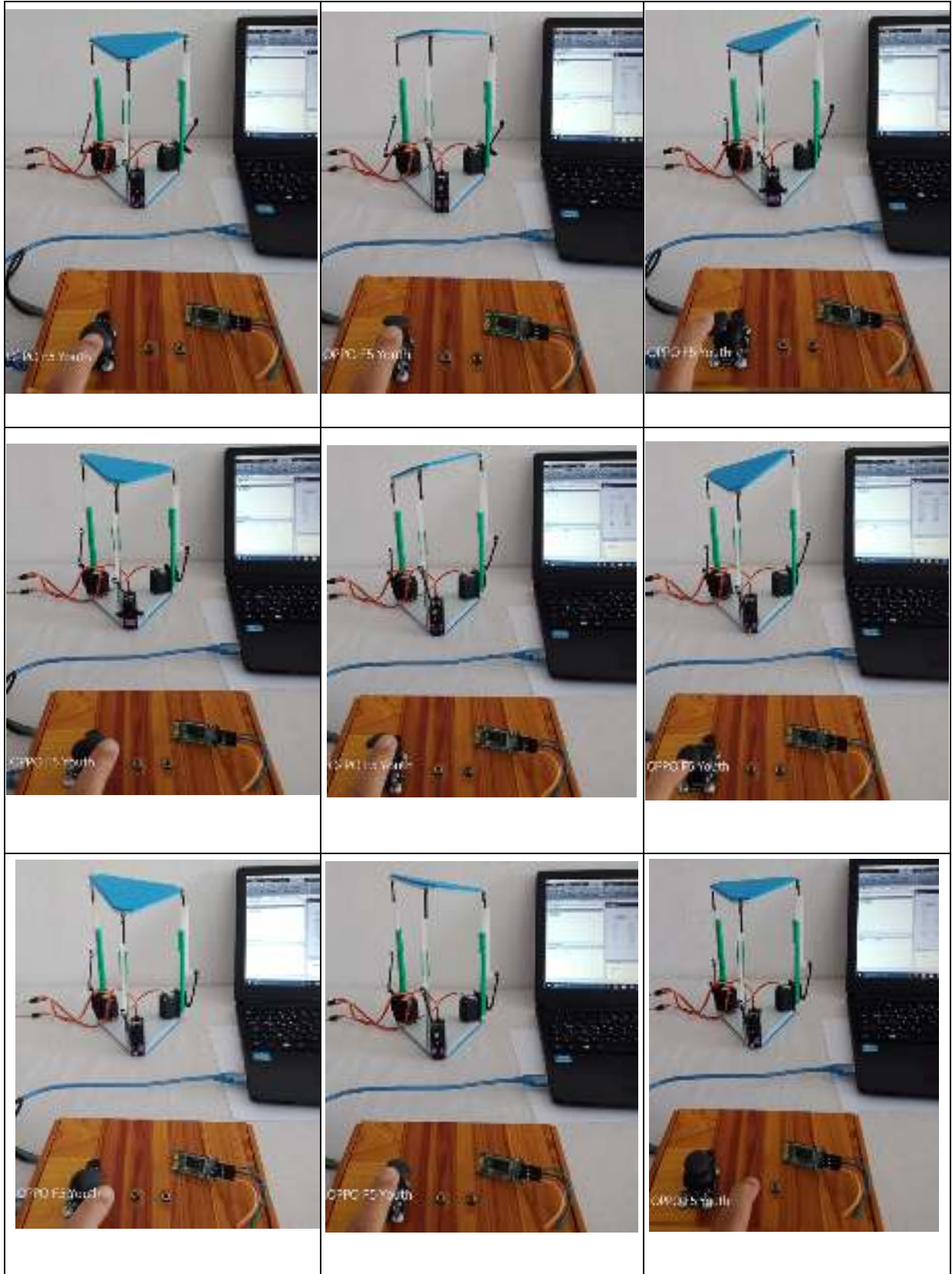
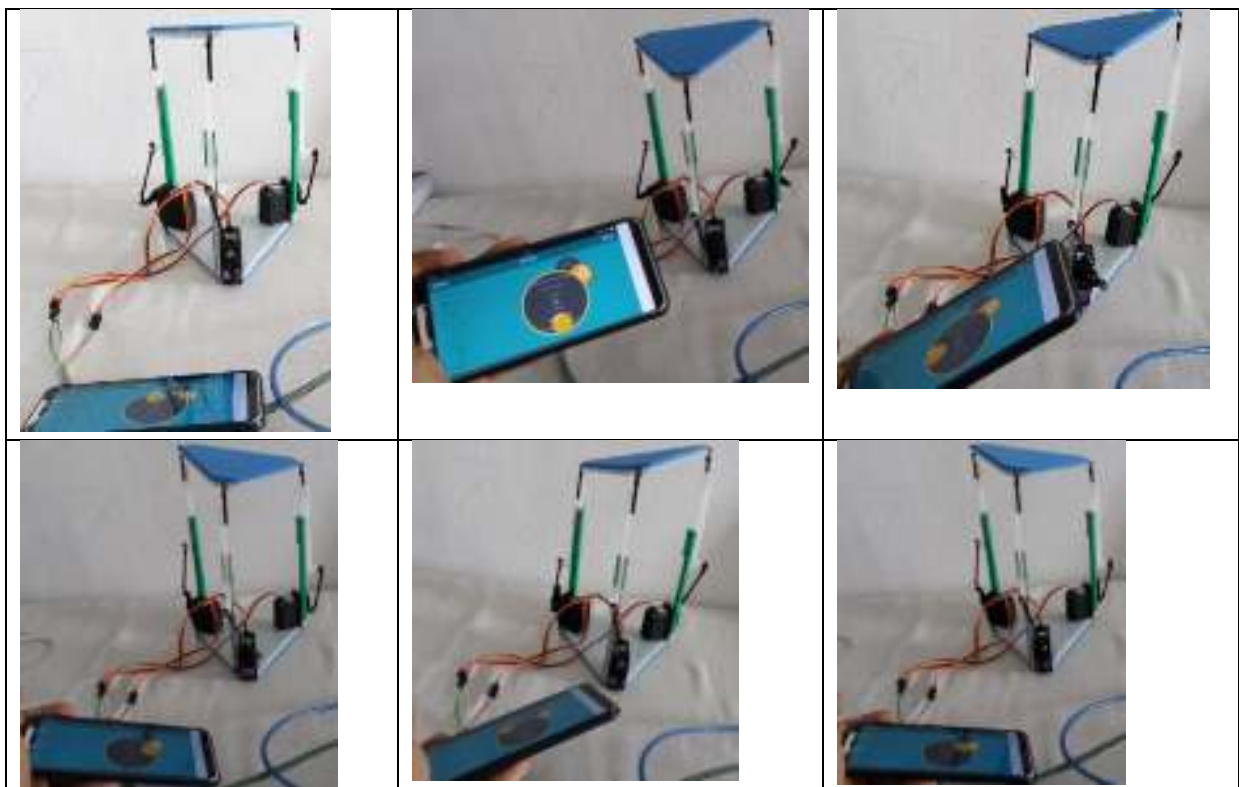




Tableau IV.1 Commande de la plateforme sous Matlab en utilisant le joystick.

### IV.2.3 Commande par Smartphone

Les images ci-dessous montrent les différentes configurations de la plateforme mobile commandée par Smartphone :





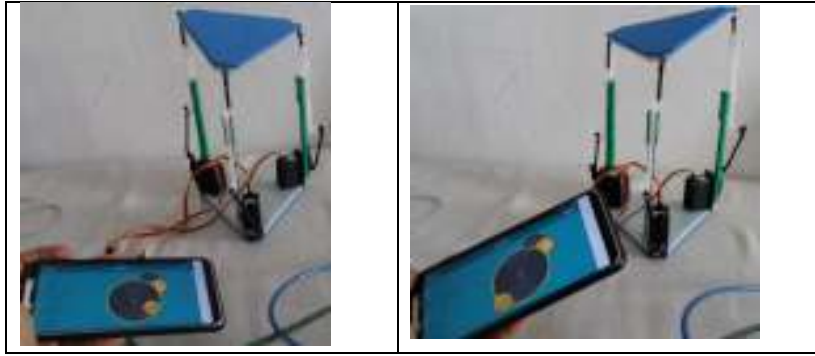


Tableau IV.2 Différentes orientations de la plateforme.

### IV.3 conclusion

Après maintes tentatives et des recherches intenses concernant tous les aspects de ce projet, notamment , l'aspect du choix de matériel , la programmation et la conception mécanique on est arrivé à réaliser cette plateforme selon le cahier des charges envisagé. Les tests de fonctionnement et de commande de la plateforme sont très concluants comme il est indiqué par les photos ci-dessus.

# **Chapitre IV**

## **Expérimentation et Résultats**

### V.1 Conclusion

Ce projet nous a permis de découvrir et de consolider nos connaissances en électronique, en programmation microcontrôleur et en mécanique. Pour cela, nous avons conçu et réalisé un prototype d'une plateforme Stewart à 3 degré de liberté commandée via une interface sous Matlab, par joystick ou par un Smartphone via le Bluetooth. De gros efforts nous ont été nécessaires pour avoir pu finalement réaliser ce projet qui fonctionne parfaitement. On a rencontré plusieurs difficultés lors de ce projet concernant la réalisation de cette plateforme Stewart à 3ddl. Parmi les problèmes qu'on a affronté on peut citer :

- Le problème de la construction mécanique de la plateforme pour choisir la configuration la plus adéquate.
- Le problème de programmation de l'algorithme de commande et d'interface avec la plateforme ( système à plusieurs facettes).
- Le problème de manque de matériel sur le marché.
- Problème de mise au point du module bluetooth HC-0 5(déconnection intempestives) au départ.

Dans ce projet, nous avons donc pu atteindre plusieurs de nos objectifs, comme :

- L'étude et la compréhension du système (conception et schéma électronique)
- La programmation de Arduino sous Matlab.
- La création d'une interface graphique (GUI) de commande de la plateforme sous Matlab.
- Le pilotage des servomoteurs par Bluetooth d'un smartphone.
- La commande d'un système non linéaire en boucle ouverte.
- La conception mécanique de toute la plateforme.
- L'utilisation de SolidWorks pour la conception des pièces.

### V.2 Perspectives

Le succès de ce projet nous a incité à proposer d'autres objectifs comme perspectives pour un travail futur à citer par exemple :

- L'implémentation d'un algorithme de commande en boucle fermée.
- L'augmentation de degré de liberté de 3 vers 6.
- Développer des applications basées sur l'utilisation de cette plateforme (jeux, robotique, stabilisation d'objets sur des véhicules mobiles...).
- Utilisation de vérins à la place de servo moteur pour la manipulation d'objets massifs.

# Bibliographie

- [1] Stewart, D., 1965. "A platform with six degree of freedom". Proc. of the IMechE, Vol. 180, 371-85. Leung, E. F. (2012). Online Learning. Singapore: Pearson.
- [2] J.-P. Merlet, *Parallel Robots*. Springer, 2000.
- [3] R.Clavel, "Conception d'un robot parallèle rapide à 4 degrés de liberté", thèse de Doctorat No 925, EPFL, 1991.
- [4] Y. X. Su and B. Y. Duan, "The application of the Stewart platform in large spherical radio telescopes," *Journal of Robotic Systems*, vol. 17, no. 7, pp. 375–383, 2000.
- [5] M.Girone, G. Burdea, M. Bouzit, V. Popescu, and J. Deutsch, "A Stewart platform-based system for ankle telerehabilitation," *Autonomous Robots*, vol. 10, pp. 203–212, 2001.
- [6] Valente, T. V., Lebrón, R. M., Perondi, E. A., 2015. "Dynamic model of an electrohydraulic Stewart platform manipulator in joint space". *23rd ABCM International Congress of Mechanical Engineering*.
- [7] Sylvain Guegan, Wisama Khalil, Damien Chablat, Philippe Wenger. Modélisation Dynamique d'un Robot Parallèle à 3-DDL : l'Orthoglide. CIFA, Jul 2002, Nantes, France. pp.1-6. hal-00162630
- [8] AMOURI AMMAR « modélisation dynamique d'un robot parallèle forme de plusieurs modules empilés » mémoire de MAGISTER, Université Larbi Ben M'hidi Oum-ElBouaghi, Faculté Des Sciences et Sciences Appliquées, 2011.
- [9] J.P. Merlet. *Les Robots Parallèles*. Hermes Science Publications, 1997. Deuxième Edition.
- [10] D.L. Pieper. "The Kinematics of Manipulators under Computer Control," Ph.D.Dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, 1968.
- [11] Ampelmann. Web page, <http://www.ampelmann.nl/>. 06/17/2019
- [12]<https://github.com/djnugent/StewartATV/blob/master/Maths%20of%20the%20Stewart%20Platform%20v5.pdf> , 27/05/2019
- [13] <http://remotexy.com/en/help/> 3/06/2019
- [14] <https://www.ukonline.be/cours/embeddedsystems/programming/chapitre1-1>  
<http://www.planetarduino.org/?cat=2494>  
<https://duino4projects.com/arduino-controlled-rotary-stewart-platform/>  
<https://grabcad.com/library/simulator-stewart-platform-1>  
[https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2017/ps158\\_aw698\\_eb645/ps158\\_aw698\\_eb645/index.html](https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2017/ps158_aw698_eb645/ps158_aw698_eb645/index.html)  
<https://atmelcorporation.wordpress.com/2014/06/12/driving-a-rotary-stewart-platform%E2%80%A8/>

# Annexe

## A.1 Les Etapes de fabrication de la plateforme

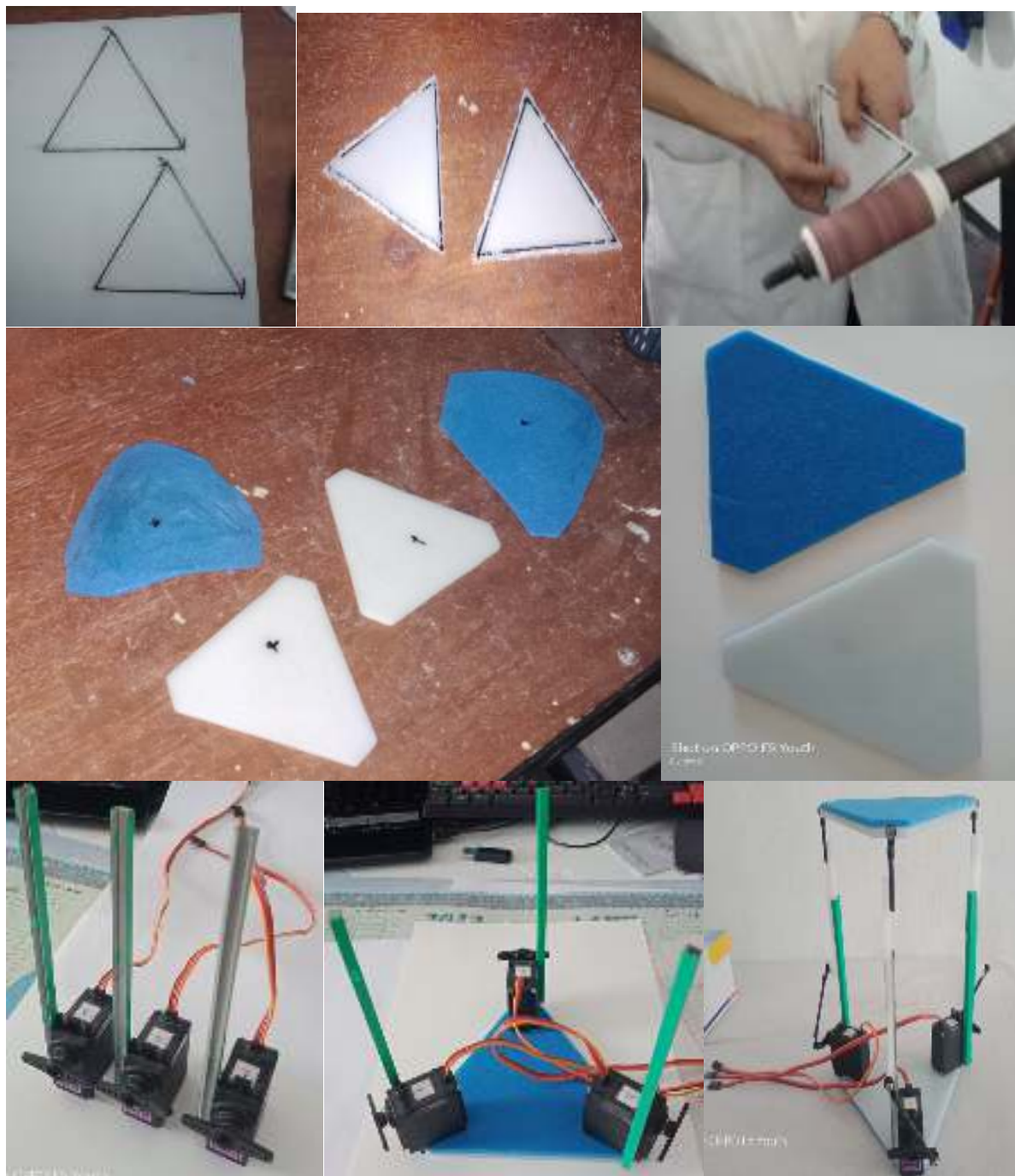


Figure A.1 les étapes de construction de Projet

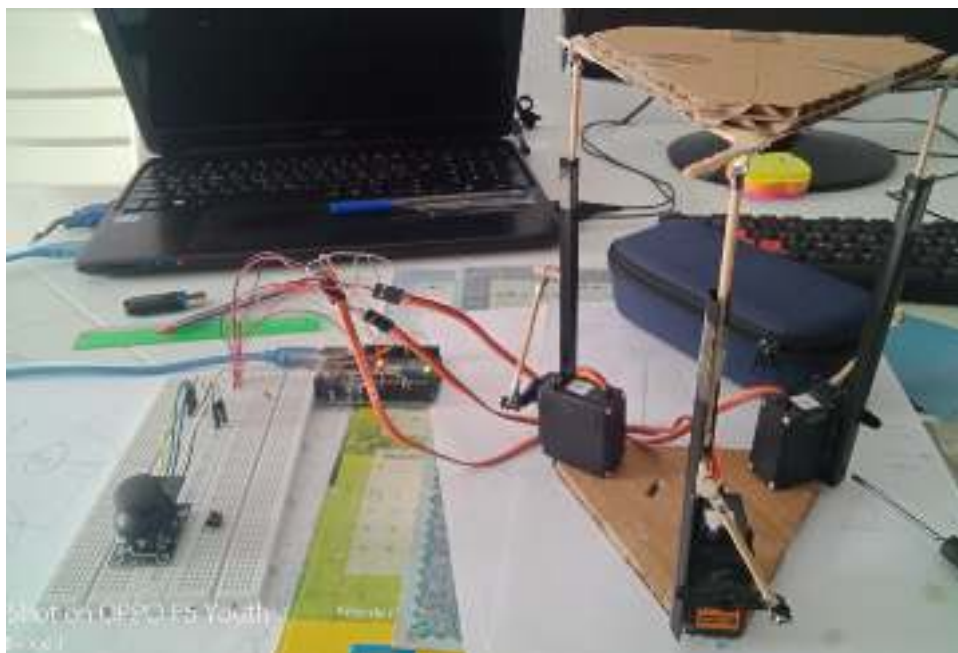


Figure A.2 Prototype numéro 1

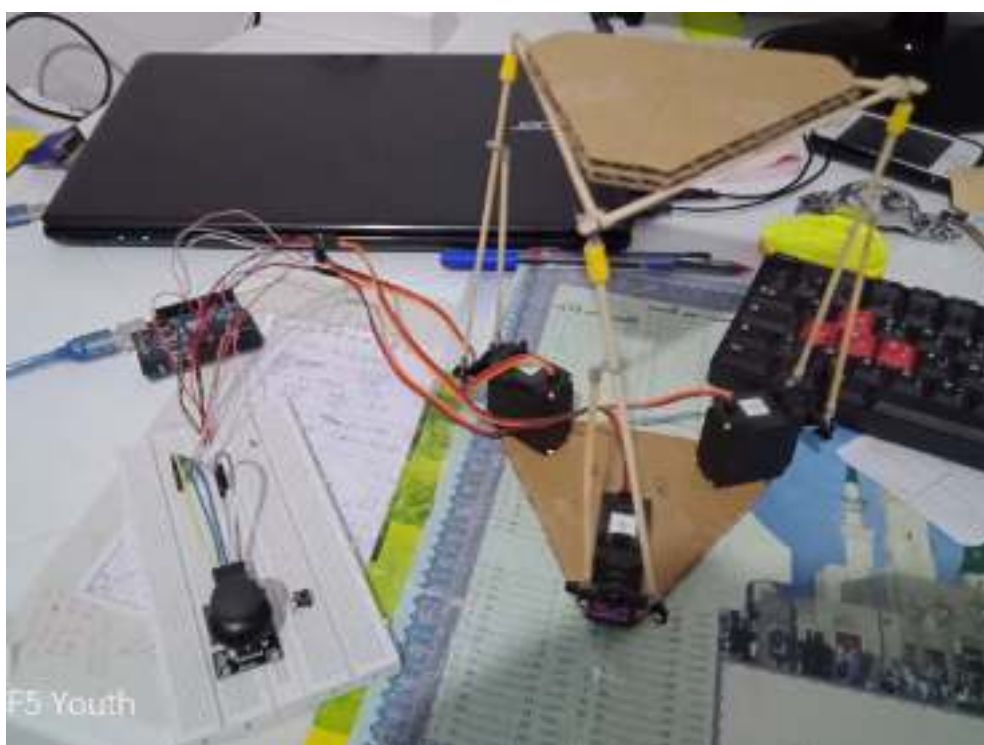


Figure A.3 Prototype numéro 2



## A.2 Caractéristiques de Arduino Uno

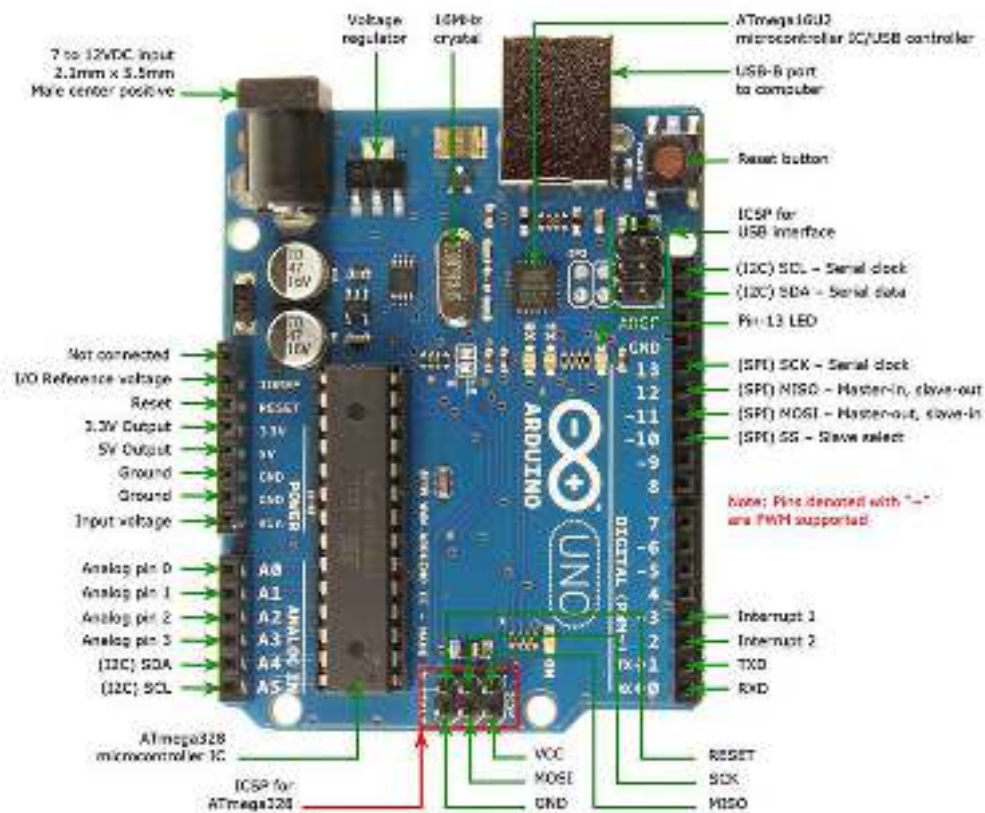


Figure A.4 Synoptique d'une carte Arduino Uno

Details	Specifications
Microcontroller	ATmega328
Operating voltage	5V
Input voltage—recommended	7-12V
Input voltage—limits	6-20V
Digital I/O pins	14 (of which six provide PWM output)
Analogue Input pins	6
DC current per I/O pin	40 mA
DC current for 3.3V pin	50 mA
Flash memory	32 kB (ATmega328), of which 0.5 kB is used by bootloader
SRAM	2 kB (ATmega328)
EEPROM	1 kB (ATmega328)
Clock speed	16 MHz

Figure A.5 Caractéristiques d'une carte Arduino Uno

### A.3 Caractéristiques de module Bluetooth HC-05

Type de module de communication	Bluetooth
Tension d'alimentation	5V DC
Taux de transfert	2Mbps
Puissance de sortie de l'émetteur, dBm	6dBm
Montage	SMD
Caractéristiques des modules de communication	antenna
Dimensions	27x13x2mm
Température de fonctionnement	-20...55°C
Bluetooth class	2
Band	2,4GHz
Type de modulation	GFSK

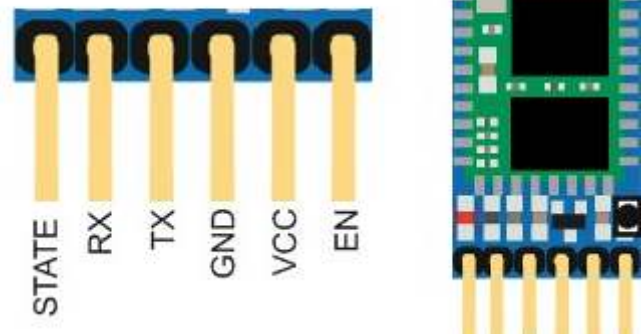


Figure A.6 Broche de Module Bluetooth HC-05

### A.4 Caractéristiques de Servomoteur MG996r

#### Caractéristiques techniques:

- Dimensions: 40 x 20 x 36,5 mm
- Type standard "TowerPro MG995"
- Poids: environ 60g
- Vitesse : 0.16 sec/60° sous 4.8V - 0.13 sec/60° sous 6.0V
- Couple : 11Kg/cm sous 4.8V - 13Kg/cm sous 6.0V
- Tension: 4.8V - 6V
- Prise servomoteur standard Orange Rouge Marron
- Débattement angulaire : 180°
- Consommation : 120 mA sans charge  
1 450 mA pour 11Kg/cm environ
- Temps de réponse de largeur d'impulsion : 5 microsecondes ou moins
- Erreur de déviation angulaire : retour à 0 degrés, 45 ° autour de chaque 3 ° ou moins d'erreur.

Orange => Signal PWM TTL standard de modélisme, de période 20ms avec un temps haut compris entre 0,7 et 2,3ms environ

Rouge => Alimentation positive entre 4,8V et 6V

Marron => Masse de l'alimentation et du signal

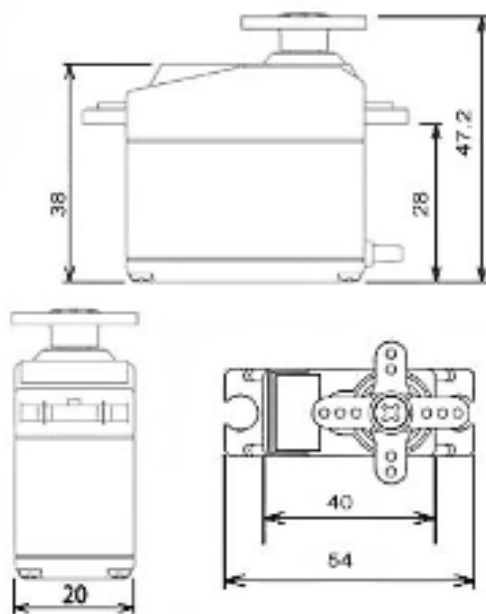


Figure A.7 Démonstrations de servomoteur MG996r

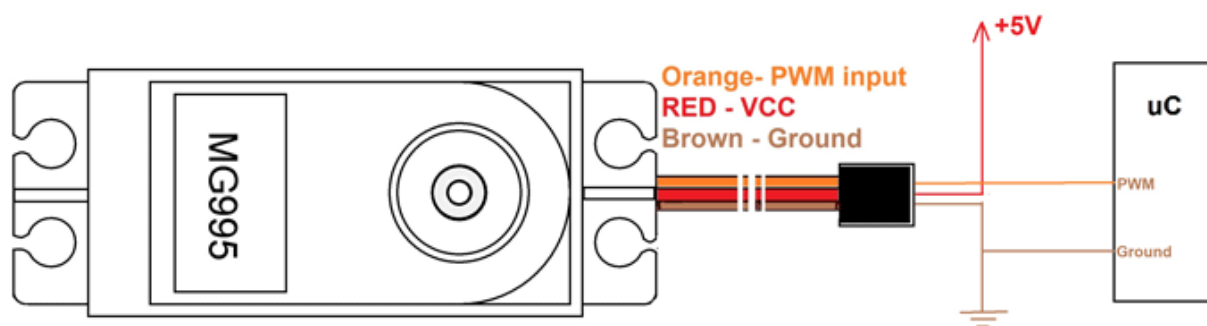


Figure A.8 Brochage de servomoteur MG996r