République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Yahia Farès Médéa
Faculté de la technologie
Département de Génie mécanique



Mémoire de fin d'étude pour l'obtention du diplôme Master
Filière : Electromécanique
Spécialité : Mécatronique

Thème

Etude et réalisation d'un bras de robot manipulateur

Présenté par
Nadir Ouldbabaali
Mohamed Yagoubi

Proposé et dirigé par
Pr. Mouloud BOUMAHDI

Promotion Juin 2024

Remerciements

Je tiens à exprimer ma profonde gratitude à Monsieur le Professeur Mouloud Boumahdi, qui m'a fait l'honneur de diriger ce travail de recherche. Je lui suis reconnaissant pour sa confiance, ses conseils précieux, sa disponibilité constante et sa patience exemplaire tout au long de ce processus.

Je suis également reconnaissant envers Monsieur le Professeur Kezrane Mohamed qui a accepté avec bienveillance de présider notre jury de mémoire. Je lui adresse meshommages les plus respectueux pour son temps et son expertise.

Un grand merci également à Monsieur le Docteur Rebhi Redha, pour sa participation active au sein de notre jury. Vos commentaires et suggestions ont été d'une grande valeur pour l'enrichissement de ce travail.

Enfin, je tiens à remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail. Votre encouragement et votre soutien ont été une source de motivation inépuisable.

DEDICASES

Je dédie ce modeste travail :

A Celle à qui mon cœur depuis sa naissance n'a pas pu éprouver qu'amour et reconnaissance, à celle qui a donné un sens à mon existence en m'offrant une éducation digne de confiance À ma chère Mère. À mon père, pour son amour et son soutien moral depuis mon enfance. Père merci.

A tous les membres de ma famille ; ma sœur Fadila

Mon frère Ali Ma fiancé Saadi Hadjer

À la famille YAGOUBI surtout Ma grand-mère BAKHTA

À la famille GUESSAYMI surtout ma grande mère AICHA

À tous mes amis surtout les membre de groupe M2 mécatronique A mes amis proches Yousri Fouad, Longo Mohamed, Yaman eyad abu mussameh, Amine Rekia, Nadir Oueld Babaali, Sido Achir, Mami Yacine

Et à tous qui m'ont connu et aidé de près et de loin dans la réalisation de ce travail.

MOHAMED YAGOUBI

Je dédie ce modeste travail :
A Celle à qui mon cœur depuis sa naissance n'a pas pu éprouver
Qu’amour et reconnaissance, à celle qui a donné un sens à mon existence
En moffrant une éducation digne de confiance
À ma chère Mère.
À mon père, pour son amour et son soutien moral depuis mon enfance.
Père merci.
A tous les membres de ma famille ; ma sœur zohor
Mais frères Ghanou et mahdi

À la famille Ouldbabaali
À la famille cocier sourtout ma grande mère Aziza
À tous mes amis surtout les membre de groupe M2 mécatronique

A mes amis proches Mami Yacine, Yousri Fouad, Longo Mohamed, Yaman eyad abu mussameh, amine Rekia, Mohamed Yagoubi, Sido Achir,
GOLF GTD
Et à tous qui m'ont connu et aidé de près et de loin dans la réalisation de ce travail.

NADIR OUELD BABAALI

الملخص

تركز هذه الأطروحة على تصميم وتصنيع وتطوير ذراع آلية ذات ست درجات من الحرية (DOF) مخصصة للتطبيقات الصناعية، بما في ذلك التلاعب الدقيق بالأشياء الهشة. تم تصميم النظام الميكانيكي المفصلي لتوفير قدر كبير من المرونة والدقة في أداء المهام مثل اختيار العناصر ووضعها. يتم توفير التحكم في الذراع من خلال وحدة التحكم الدقيقة Arduino، المرتبطة بوحدة Bluetooth HC05 للإدارة عن بعد والتفاعل السهل. يتضمن النهج المعتمد في هذا المشروع منهجية تصميم صارمة، يتبعها إنشاء واختبار نموذج أولي وظيفي.

الكلمات المفتاحية: مناور الروبوت، التحكم، اردوينو،

Résumé

Ce mémoire se concentre sur la conception, la fabrication et le développement d'un bras robotique à six degrés de liberté (ddl) destiné à des applications industrielles, notamment la manipulation délicate d'objets fragiles. Le système mécanique articulé est conçu pour offrir une grande flexibilité et précision dans l'exécution de tâches telles que la sélection et le placement d'éléments. Le contrôle du bras est assuré par un microcontrôleur Arduino, associé à un module Bluetooth HC05 pour une gestion à distance et une interaction facilitée. L'approche adoptée dans ce projet inclut une méthodologie de conception rigoureuse, suivie par la réalisation et les tests d'un prototype fonctionnel.

Mots clés : Robot manipulateur, Contrôle, Arduino,

Abstract

This dissertation focuses on the design, manufacturing and development of a six degrees of freedom (DOF) robotic arm intended for industrial applications, including the delicate manipulation of fragile objects. The articulated mechanical system is designed to provide great flexibility and precision in performing tasks such as selecting and placing elements. Control of the arm is provided by an Arduino microcontroller, associated with an HC05 Bluetooth module for remote management and easy interaction. The approach adopted in this project includes a rigorous design methodology, followed by the creation and testing of a functional prototype.

Keywords: Robot manipulator, Control, Arduino,

Liste des tableaux

Tableau 3. 1 Caractéristiques de la carte ARDUINO 27

Liste des figures

Figure 1. 1 Unimat 1961, premier robot industriel développé aux états unis 2
Figure 1. 2 Robot cartésien 3
Figure 1. 3 Robot cylindriques 3
Figure 1. 4 Robot sphériques 4
Figure 1. 5 Robot articulés 4
Figure 1. 6 Robot SCARA 5
Figure 1. 7 Structure fonctionnelle d'un robot manipulateur 6
Figure 1. 8 Composants d'un bras manipulateur 7
Figure 1. 9 Composants d'un robot manipulateur 8
Figure 1. 10 Articulations 9

Figure 2. 1 Robot série 12
Figure 2. 2 Paramètres de Denavit-Hartenberg 13
Figure 2. 3 repérage du bras manipulateur 15
Figure 2. 4 Environnement du Solidworks 17
Figure 2. 5 La base 18
Figure 2. 6 le levier 1 19
Figure 2. 7 le levier 2 20
Figure 2. 8 le levier 3 21
Figure 2. 9 Organe terminal 22
Figure 2. 13 : Commande des moteurs du bras [1] 31

Figure 3. 1 carte Arduino 27
Figure 3. 2 : Interface IDE 28
Figure 3. 3 : Module Bluetooth HC-05 30
Figure 3. 4 : Schéma du circuit électrique [1] 32
Figure 3. 5 Bras manipulateur réalisé 33

Table des matières

الملخص v

Résumé v

Abstract vi

Liste des tableaux vii

Liste des figures viii

Table des matières ix

Introduction Générale 1

Chapitre 1 : Généralités sur les robots manipulateur 2

1.1 Introduction 2

1.2. Historique 2

1.3. Types des bras manipulateurs 3

1.3.1 Robots cartésiens 3

1.3.2 Robot cylindriques 3

1.3.3 Robot sphériques 4

1.3.4 Robot articulés 4

1.3.5 Robot SCARA 5

1.4. Structure fonctionnelle d'un robot 5

1.4.1 Système d'enduction 5

1.4.2 Structure 5

1.4.3 Système de contrôle 5

1.4.4 Interface utilisateur 6

1.4.5 Système de perception 6

1.5 Composants d'un robot manipulateur 6
1.6. Système mécanique articulé 8
1.6 Caractéristiques des Robots 9
1.6.1 Charge maximale transportable 9
1.6.2 Architecture du S.M.A. 10
1.6.3 Volume de travail 10
1.6.4 Positionnement précis 10
1.6.5 Vitesse de déplacement 10
1.6.6 Taille du robot 10
1.6.7 Coût du robot 10
1.6.8 Entretien du robot 11
1.7. Conclusion 11
Chapitre 2 : Modélisation et conception du bras manipulateur 12
2.1 Introduction 12
2.2 Description de la géométrie d'un robot série 12
2.3 Modélisation du bras manipulateur 14
2.3 Conception du bras manipulateur 17
2.8 Fabrication des pièces 23
2.8.1 Principe de l'impression 3D 23
2.8.2 Matériaux utilisés 23
2.8.3 Processus de fabrication 23
2.8.4 Avantages et limitations de l'impression 3D 24
2.9 Conclusion 24
Chapitre 3 : Système de commande du bras manipulateur 26
3.1 Introduction 26
3.2 Carte ARDUINO 26
3.2.1 Carte ARDUINO UNO 26
3.2.2 Caractéristiques 26

3.2.3 Arduino IDE 28
3.2.4 Actionneurs (Servomoteurs) 28
3.2.5 Module Bluetooth HC-05 30
3.2.6 Schéma du système électrique 31
3.5 Conclusion 33
Conclusion Générale 34
Références 35
Annexe : Programme [1] 36

Introduction Générale

Dans un monde où l'automatisation et la précision sont devenues des impératifs dans de nombreux secteurs, les robots articulés jouent un rôle crucial. Ces systèmes mécaniques complexes, propulsés par des actionneurs et dirigés par des calculateurs sophistiqués, sont conçus pour exécuter une large gamme de tâches, allant de la manipulation délicate d'objets fragiles à l'assemblage industriel de haute précision.

Ce mémoire se concentre sur l'étude d'un robot articulé spécifique, doté de six articulations rotoïdes lui conférant six degrés de liberté. Cette configuration lui permet une grande flexibilité et une précision remarquable dans l'exécution de ses fonctions. Structurée en trois chapitres, Cette étude vise à fournir une compréhension approfondie du domaine des bras manipulateurs, de leur conception à leur mise en œuvre.

Le premier chapitre offre un aperçu des concepts fondamentaux nécessaires à la compréhension des bras manipulateurs, en abordant leurs principes de fonctionnement, leurs applications et les défis associés à leur conception.

Le deuxième chapitre traite de la modélisation technique du bras manipulateur étudié ainsi que de sa conception.

Le troisième chapitre est dédié à la réalisation pratique du bras et à sa commande à l'aide de la carte Arduino, un outil de prototypage électronique largement utilisé pour sa flexibilité et sa facilité d'utilisation.

Enfin, ce mémoire se conclut par une conclusion générale des travaux réalisés. À travers cette étude, nous espérons contribuer à l'avancement des connaissances et des technologies dans ce secteur.

Chapitre 1 : Généralités sur les robots manipulateur

1.1 Introduction

Les robots manipulateurs jouent un rôle crucial dans l'automatisation industrielle, apportant précision, répétabilité et efficacité aux processus de fabrication et d'assemblage. Utilisés dans divers secteurs, tels que l'automobile, l'aéronautique, l'électronique et la santé, ces robots permettent de réaliser des tâches complexes et souvent dangereuses pour les humains, améliorant ainsi la productivité et la sécurité au travail. Ce chapitre explore les bases des robots manipulateurs, leur historique, les différents types existants, leur structure fonctionnelle et leurs composants principaux.

1.2. Historique

L'histoire des bras manipulateurs remonte au début du 20e siècle, avec les premières machines automatisées utilisées dans l'industrie. Cependant, c'est dans les années 1950 et 1960 que les véritables robots manipulateurs ont commencé à émerger, notamment avec le développement du premier robot industriel, l'Unimate, par George Devol et Joseph Engelberger. Depuis lors, la technologie des bras robotiques a connu une évolution rapide, passant de systèmes simples à des machines sophistiquées capables d'effectuer des tâches complexes dans une variété d'environnements.



Figure 1. 1 Unimat 1961, premier robot industriel développé aux états unis

1.3. Types des bras manipulateurs

Les bras manipulateurs peuvent être classés en plusieurs catégories selon leur configuration géométrique et leur mode de mouvement : cartésien (PPP), Cylindrique (RPP), Sphérique (RRP), Articulé (RRR), SCARA.

1.3.1 Robots cartésiens

Ces robots se déplacent le long de trois axes rectangulaires, x, y et z, similaire au mouvement d'une imprimante 3D. Ils sont idéaux pour des tâches nécessitant une précision de positionnement élevée dans un espace cartésien fixe.

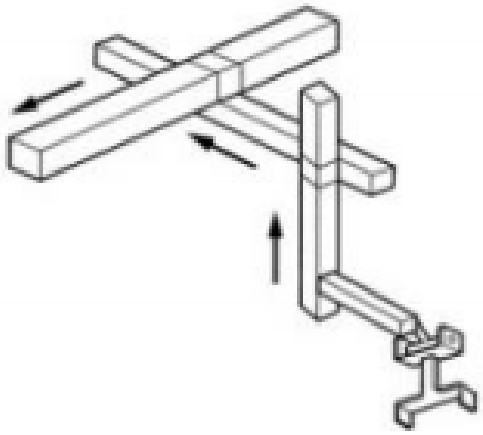


Figure 1. 2 Robot cartésien

1.3.2 Robot cylindriques

Ces robots ont une configuration qui leur permet de se mouvoir dans un espace cylindrique, avec une combinaison de mouvements linéaires et rotatifs. Ils sont souvent utilisés dans des applications nécessitant un accès à des zones circulaires ou tubulaires.

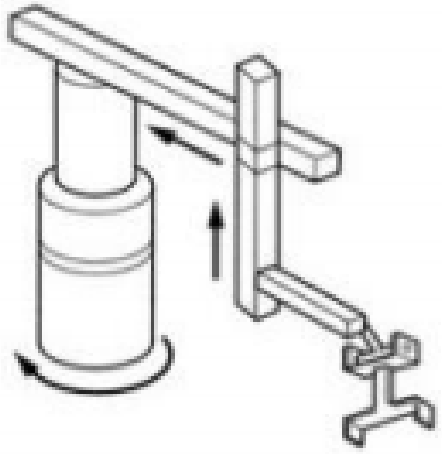


Figure 1. 3 Robot cylindriques

1.3.3 Robot sphériques

Les robots sphériques sont conçus pour se déplacer dans un espace sphérique autour d'un point fixe. Leur gamme de mouvement est définie par des angles d'élévation et d'azimut, ce qui les rend adaptés pour des tâches qui nécessitent une grande flexibilité de mouvement dans un espace limité.

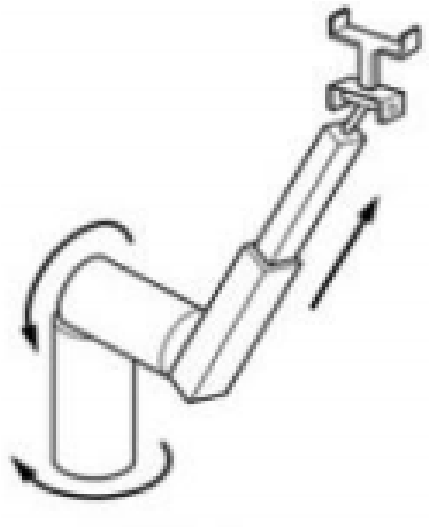


Figure 1. 4 Robot sphériques

1.3.4 Robot articulés

Ces robots sont caractérisés par une série de joints articulés qui leur permettent une grande flexibilité et une large gamme de mouvement. Ils peuvent imiter les mouvements d'un bras humain et sont donc largement utilisés dans des applications nécessitant une manipulation précise et complexe d'objets.

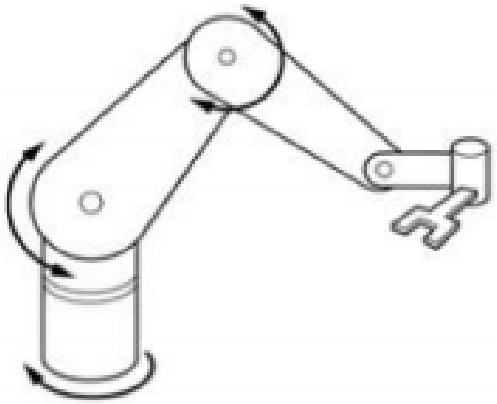


Figure 1. 5 Robot articulés

1.3.5 Robot SCARA

Un robot SCARA (Selective Compliance Assembly Robot Arm) est un type de robot manipulateur à quatre axes, conçu spécifiquement pour les applications d'assemblage et de manipulation dans les environnements industriels. Le terme "Selective Compliance" fait référence à la capacité du robot à être rigide dans les directions axiales (x et z) tout en étant flexible dans la direction radiale (y), ce qui lui permet d'accommoder les variations de position des pièces à assembler.

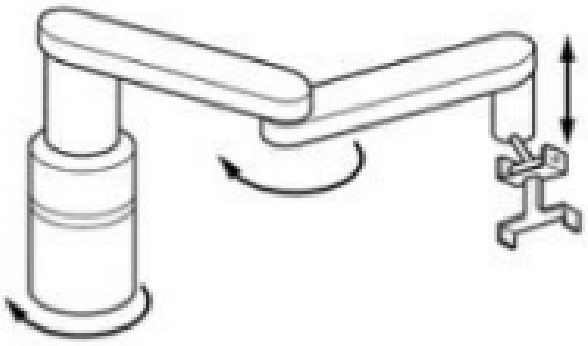


Figure 1. 6 Robot SCARA

1.4. Structure fonctionnelle d'un robot

Un robot manipulateur typique est composé de plusieurs éléments clés qui travaillent ensemble pour réaliser des tâches :

1.4.1 Système d'enduction

Comprend les moteurs et les mécanismes qui permettent au robot de se mouvoir dans les différentes dimensions de son espace de travail.

1.4.2 Structure

L'ossature du robot, qui peut être fixe ou mobile, supporte les différents composants et assure la rigidité nécessaire aux opérations.

1.4.3 Système de contrôle

Le cerveau du robot, qui gère les commandes, la logique de fonctionnement et l'interaction avec les capteurs.

1.4.4 Interface utilisateur

Permet la communication entre l'opérateur et le robot, permettant de programmer et de surveiller les opérations.

1.4.5 Système de perception

Inclut les capteurs et les caméras qui permettent au robot de percevoir son environnement et d'ajuster ses actions en conséquence.

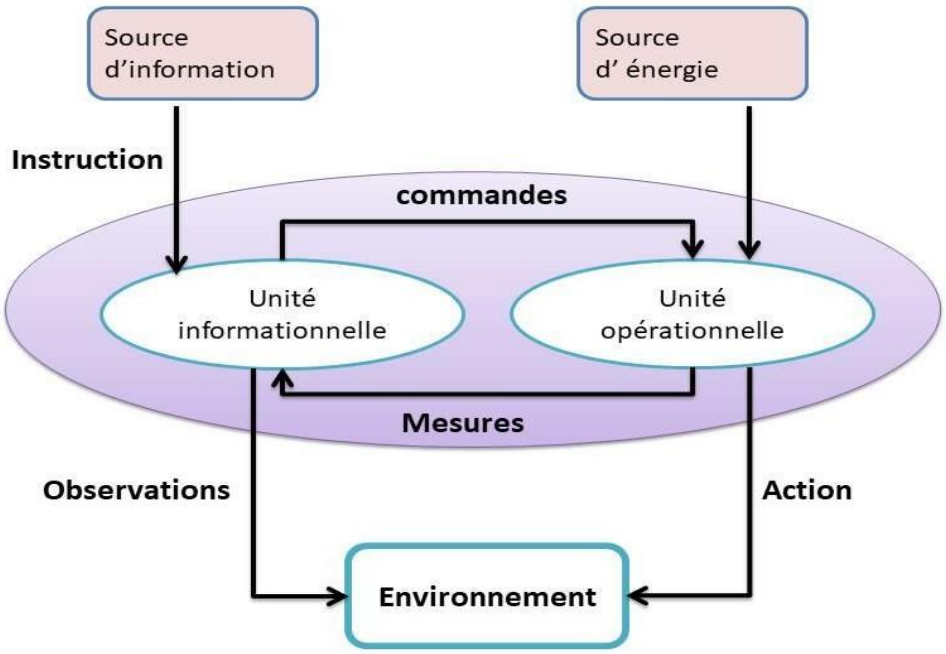


Figure 1. 7 Structure fonctionnelle d'un robot manipulateur

1.5 Composants d'un robot manipulateur

Les bras robotiques, qui tirent leur nom de leur ressemblance avec un bras humain, sont généralement montés sur une base. Ils sont constitués de plusieurs éléments clés :

Système mécanique articulé

Les bras robotiques contiennent plusieurs articulations qui agissent comme des axes permettant un certain degré de mouvement. Plus le nombre d'articulations rotatives d'un bras robotique est élevé, plus sa liberté de mouvement est grande. La plupart des bras robotiques industriels utilisent quatre à six articulations, fournissant le même nombre d'axes de rotation pour le mouvement.

Contrôleur du robot

Il s'agit du cerveau du robot, qui gère les commandes, la logique de fonctionnement et l'interaction avec les capteurs.

Outil en bout de bras

Il peut s'agir d'une pince, d'un manipulateur, d'un dispositif de soudage, etc., qui est attaché à l'extrémité du bras robotique pour effectuer des tâches spécifiques.

Actionneurs

Ce sont les moteurs et les mécanismes qui fournissent la force nécessaire au mouvement des articulations du bras robotique.

Capteurs

Ils mesurent les positions, les vitesses et les forces exercées par le bras robotique, permettant au robot de percevoir son environnement et d'ajuster ses actions en conséquence.

Systèmes de vision

Ils permettent au robot de voir et de reconnaître les objets dans son environnement, ce qui est essentiel pour des tâches telles que le positionnement précis et l'assemblage.

Systèmes d'alimentation

Ils fournissent l'énergie nécessaire au fonctionnement du robot, que ce soit par câble ou par batterie.

Composants logiciels

Ils comprennent les programmes et les algorithmes qui permettent au robot d'exécuter des tâches spécifiques et de communiquer avec d'autres systèmes.

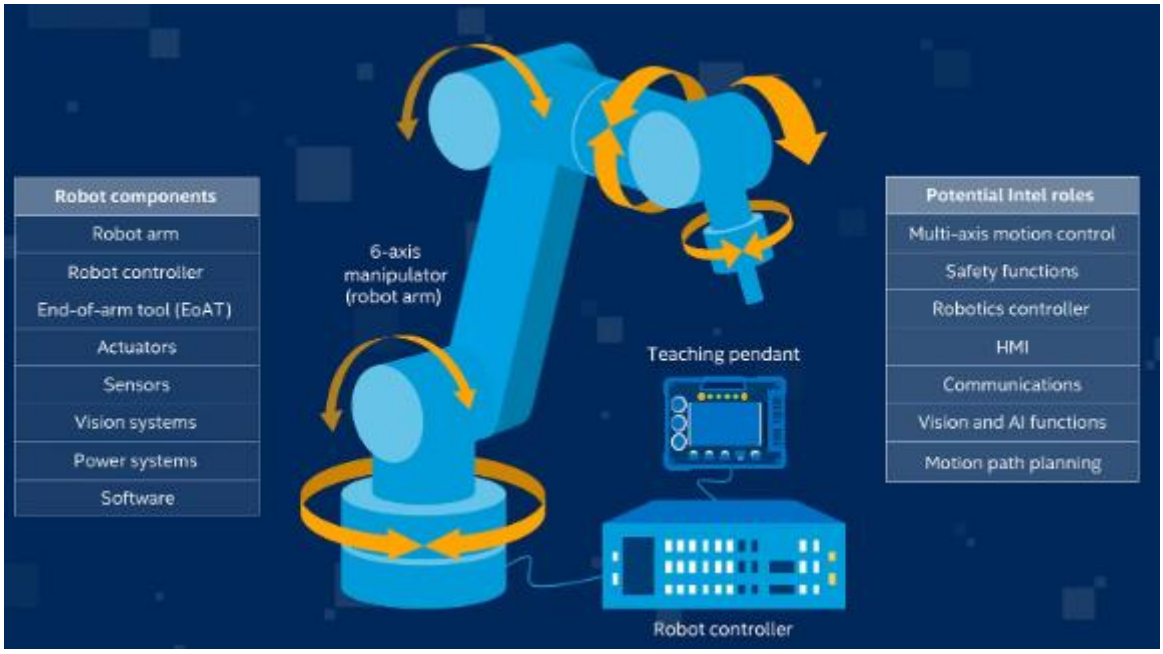


Figure 1. 8 Composants d'un bras manipulateur

1.6. Système mécanique articulé

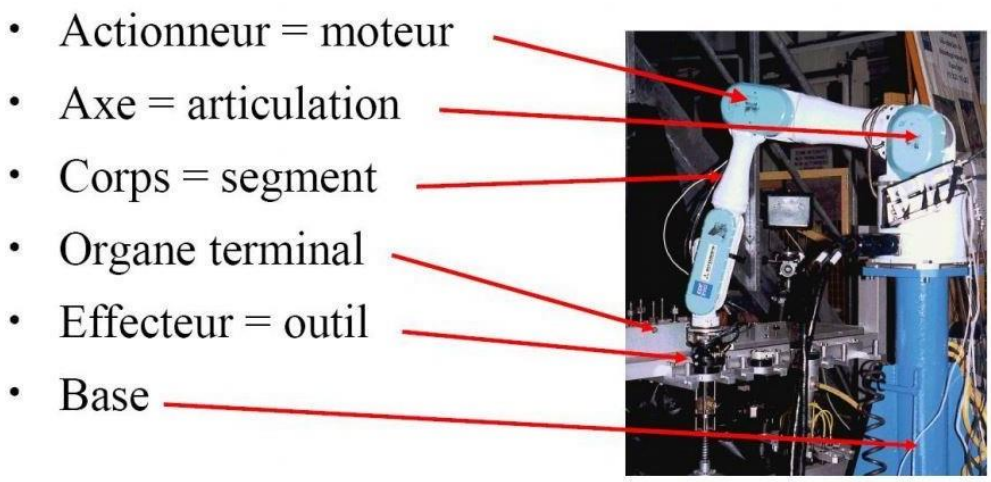


Figure 1. 9 Composants d'un robot manipulateur

La Base

Les robots industriels ont généralement une base fixée à l'emplacement du travail. Cette base fournit un point d'ancrage stable, assurant la stabilité et la précision des opérations du robot.

Segment (corps du robot)

Les segments, ou corps du robot, sont des ensembles solides et rigides qui peuvent se déplacer par rapport à la base et les uns par rapport aux autres. Ils constituent les parties principales du bras manipulateur, permettant le mouvement et l'atteinte des différentes positions dans l'espace de travail.

Organe terminal

L'organe terminal fait référence à tous les dispositifs utilisés pour manipuler des objets ou les modifier. Cela inclut les dispositifs de serrage, les dispositifs magnétiques ou à dépression. L'organe terminal permet au robot d'interagir directement avec son environnement.

Effecteur

L'effecteur est connecté à l'extrémité du bras du robot et agit comme une main. C'est la partie du robot qui entre en contact direct avec le matériau manipulé. Il existe diverses versions d'effecteurs, comme les pinces, les pompes à vide, les aimants et les torches de soudage. Certains robots peuvent changer leurs effecteurs finaux, leur permettant d'effectuer diverses tâches en fonction des besoins de l'application.

Articulation

Les articulations relient successivement deux segments, restreignant le degré de liberté de l'un par rapport à l'autre. La mobilité de l'articulation, notée mm, peut varier de 0 à 6. Lorsque m=1m=1, l'articulation est dite simple et peut être de type rotoïde ou prismatique.

-Articulation rotoïde

Une articulation rotoïde, notée R, est de type pivot. Elle limite le mouvement entre deux segments à une rotation autour d'un axe commun. La position relative entre les deux segments est déterminée par l'angle de rotation autour de cet axe.

- Articulation prismatique

Une articulation prismatique, notée P, est de type glissière. Elle limite le mouvement entre deux segments à une translation le long d'un axe commun. La position relative entre les deux segments est mesurée par la distance le long de cet axe.

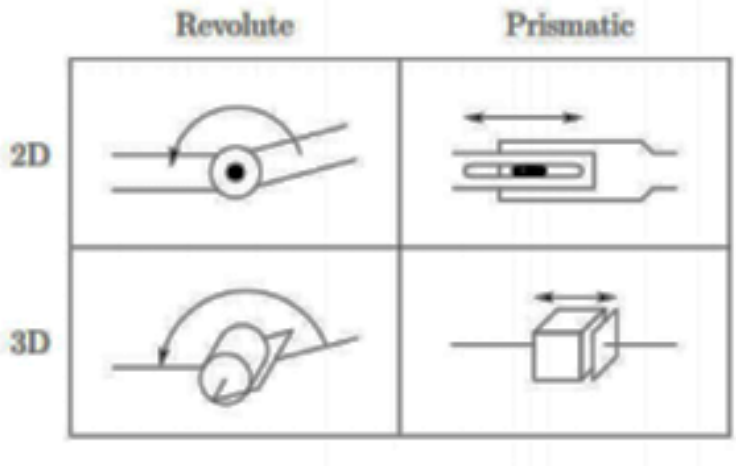


Figure 1. 10 Articulations

1.6 Caractéristiques des Robots

Lors de la sélection d'un robot manipulateur, il est crucial de prendre en compte l'application spécifique à laquelle il sera destiné. Voici quelques éléments à considérer :

1.6.1 Charge maximale transportable

Dans les conditions les plus défavorables, c'est-à-dire en élongation maximale, il est nécessaire de déterminer la charge maximale que le robot peut transporter. Cette charge peut varier de quelques kilos à plusieurs tonnes.

1.6.2 Architecture du S.M.A.

Le choix de l'architecture du Système de Manipulation Automatisée (S.M.A.) est influencé par la tâche à effectuer. Par exemple, il est important de comparer les robots à structure rigide avec ceux équipés de segments et/ou d'articulations flexibles.

1.6.3 Volume de travail

Le volume ou espace de travail (workspace) est défini comme la totalité des points accessibles par l'organe terminal du robot. Ce volume accessible est complexe à représenter pour les robots à 6 degrés de liberté (DDL) et varie en fonction de plusieurs facteurs :

- La dynamique du robot
- La taille des segments
- Les articulations avec butées

Graphiquement, on représente le volume accessible d'un point (centre de l'outil, centre du poignet) pour une orientation minimale.

1.6.4 Positionnement précis

Le positionnement précis fait référence à la capacité du robot à atteindre une position et une orientation spécifiques dans l'espace cartésien. La répétabilité, qui mesure cette capacité, est généralement d'environ 0.05 mm. Ce paramètre est essentiel pour les applications nécessitant une haute précision.

1.6.5 Vitesse de déplacement

La vitesse de déplacement, incluant la vitesse maximale lors de l'élongation maximale et la vitesse de rotation, est un facteur crucial à considérer.

1.6.6 Taille du robot

La taille du robot peut varier de quelques centaines de kilos à plusieurs tonnes. La taille doit être adaptée à l'espace disponible et à l'application prévue.

1.6.7 Coût du robot

Le coût initial d'acquisition du robot est un facteur important à considérer, en balance avec les avantages qu'il procure en termes de productivité et de précision.

1.6.8 Entretien du robot

L'entretien du robot, particulièrement pour ceux évoluant dans des environnements sensibles ou dangereux (comme les chambres froides), peut être complexe. Il est crucial de prendre en compte les besoins en maintenance pour garantir une opération fiable et continue.

1.7. Conclusion

En conclusion, les robots bras manipulateurs industriels sont des systèmes complexes et polyvalents, composés d'une variété de composants essentiels qui travaillent en harmonie pour réaliser des tâches précises et efficaces dans les environnements industriels. Les articulations rotatives offrent une grande liberté de mouvement, tandis que le contrôleur du robot gère les commandes et la logique de fonctionnement. L'outil en bout de bras, les actionneurs, les capteurs, les systèmes de vision, les systèmes d'alimentation et les composants logiciels complètent le système, permettant aux robots de s'adapter à une large gamme d'applications et de répondre aux exigences de précision et de productivité des industries modernes.

Chapitre 2 : Modélisation et conception du bras manipulateur

2.1 Introduction

La modélisation et la conception d'un bras manipulateur sont des étapes cruciales dans le développement de systèmes robotiques efficaces et précis. Un bras manipulateur à 5 degrés de liberté (DDL) offre une grande flexibilité et une capacité de mouvement sophistiquée. Ce chapitre explore la modélisation cinématique ainsi que les principes de conception de ce type de bras manipulateur.

2.2 Description de la géométrie d'un robot série

Un robot série est composé d'une séquence de n +1 chaînons et n liaisons. Les chaînons sont supposés être parfaitement rigides. Les liaisons sont soit révolutifs soit prismatiques et sont supposées parfaites (pas de jeu, pas d'élasticité). Les chaînons sont numérotés de sorte que le chaînon 0 constitue la base du robot et le chaînon n est le chaînon terminal.

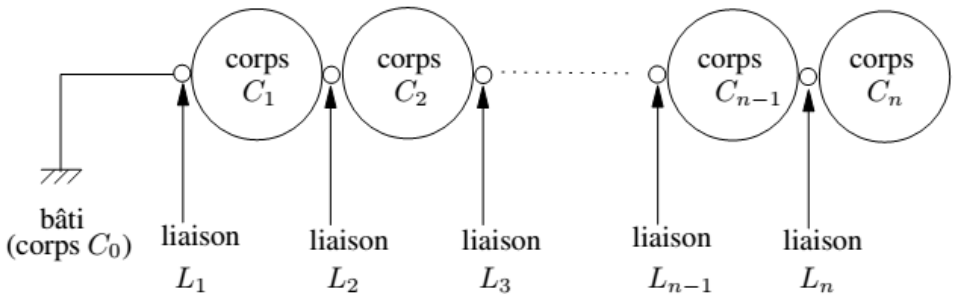


Figure 2. 1 Robot série

La liaison j relie le chaînon j au chaînon la liaison j-1 et sa variable est notée qi. Afin de définir la relation entre l'emplacement des chaînons, nous désignons un repère Rj fixé à chaque chaînon j, tel que :

l'axe zj se trouve le long de l'axe de la liaison j

l'axe xj est aligné avec la normale commune entre zj et zj +1.

L'axe yj est formé par la règle de la main droite pour compléter le système de coordonnées

(xj, yj,zj).

Repérage des organes mobiles du bras selon Denavit-Hartenberg

On associe au i-ème corps le repère $R_i = (O_i, x_i, y_i, z_i)$, avec $i = 0, 1, ..., n$. Pour placer les repères R_1 à R_n selon la règle des paramètres de Denavit-Hartenberg, il convient de suivre les règles suivantes, pour $i = 2, 3, ..., n$:

- z_{i-1} est l'axe de la liaison L_i, librement orienté.
- O_{i-1} est le pied de la perpendiculaire commune aux axes des liaisons L_i et L_{i+1}, situé sur L_{i-1}.
- x_{i-1} est le vecteur unitaire de cette perpendiculaire commune, orienté de L_i vers L_{i+1}.

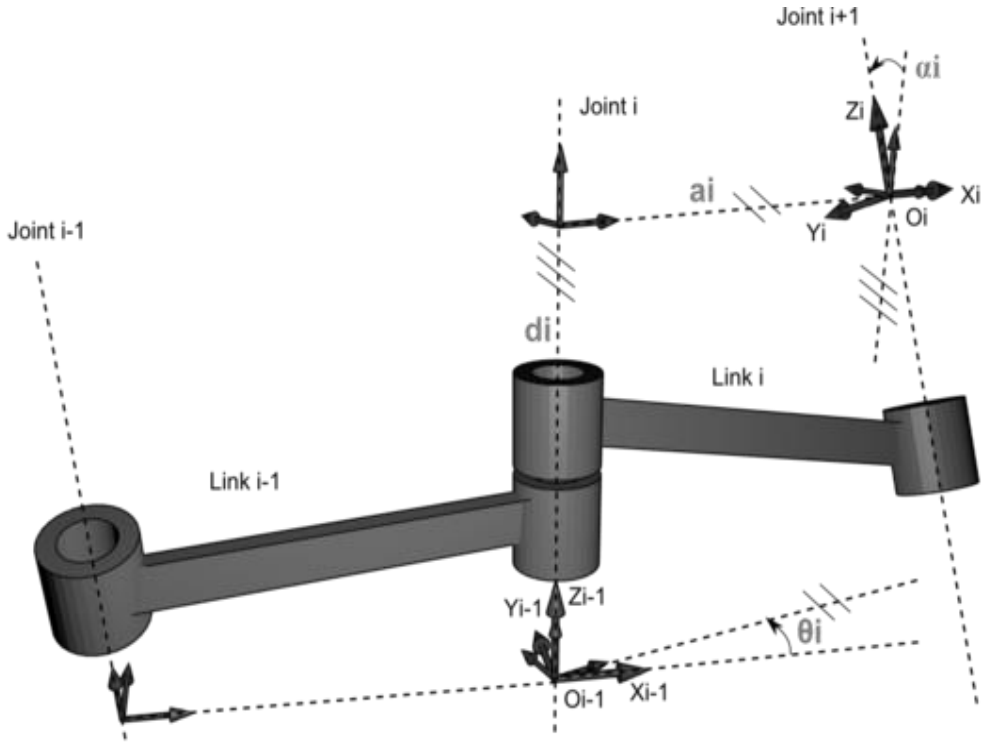


Figure 2. 2 Paramètres de Denavit-Hartenberg

Paramétrage selon la méthode des paramètres de Denavit-Hartenberg

Les paramètres α_i, a_i, θ_i, et d_i, avec $i = 1, 2,..., n$, placés sur cette figure sont appelés paramètres de Denavit-Hartenberg modifiés. On peut les définir comme suit :

z_i est l'axe de la liaison i.

α_i (twist angle) est l'angle algébrique entre z_i et z_{i+1}, mesuré autour de x_i ;

a_i (link length) est la distance arithmétique de la perpendiculaire commune aux axes des liaisons L_i et L_{i+1} mesurée le long de x_i ;

θ_i (joint angle) est l'angle algébrique entre x_{i-1} et x_i, mesuré autour de z_i ;

r_i (offset length) est la distance algébrique du point O_i à la perpendiculaire, mesurée le long de z_i. c'est la distance de x_{i-1} à x_i, , mesurée le long de z_i.

Relations géométriques

Conformément au paramétrage de Denavit Hartenberg, la matrice de transformation entre les corps C_{i-1} et C_i est donc :

La matrice de transformation du repère Rj-1 au repère trame Rj est exprimée en fonction des quatre paramètres géométriques suivants

$$ {}^{i-1}T_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

2.3 Modélisation du bras manipulateur

Le bras robotique étudié possède cinq degrés de liberté (DOF) ainsi qu'un mouvement de pince (5+1). Il est comparable au bras humain en termes de nombre d'articulations. Ces articulations permettent la rotation de l'épaule, le mouvement avant-arrière de l'épaule, le mouvement du coude, le mouvement vertical du poignet, la rotation du poignet, et le mouvement de la pince.

Le bras manipulateur possède cinq articulations rotatives et une pince mobile. L'articulation 1 représente l'épaule avec un axe de mouvement z1. Cette articulation permet une rotation angulaire θ1 autour de l'axe z1 dans le plan x1y1. L'articulation 2, identifiée comme le bras supérieur, a un axe perpendiculaire à celui de l'articulation 1 et permet une rotation angulaire θ2 autour de l'axe z2 dans le plan x2y2. Les axes z3 de l'articulation 3 (avant-bras) et z4 de l'articulation 4 (poignet) sont parallèles à l'axe z2 de l'articulation 2, offrant des rotations angulaires θ3 et θ4 respectivement dans les plans x3y3 et x4y4. L'articulation 5 est identifiée comme la rotation de la pince, avec un axe z5 perpendiculaire à l'axe z4z4, permettant des mouvements angulaires θ5 dans le plan x5y5 . Une vue graphique de toutes les articulations est présentée dans la Figure 2.3.

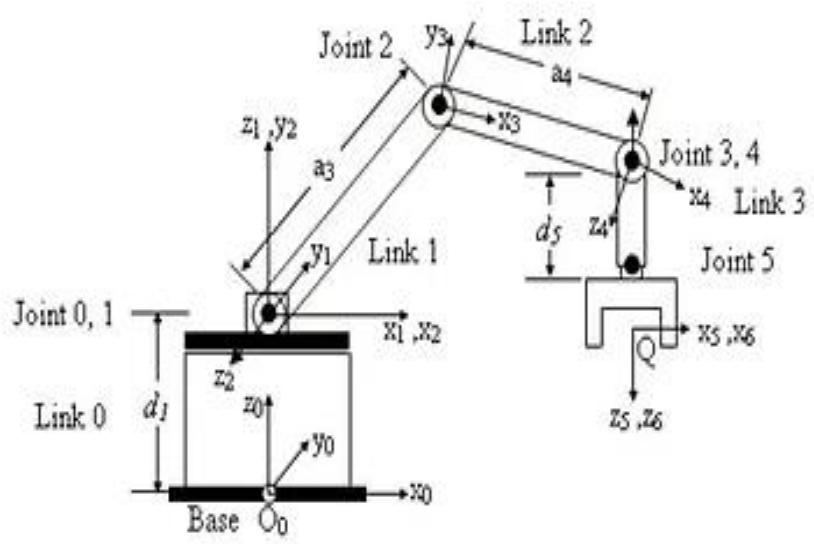


Figure 2. 3 repérage du bras manipulateur

Paramètres de DENAVIT-HERTENBERG

i	α_i	a_i	d_i	θ_i
1	0	0	d1	$\theta 1$
2	90°	0	0	$\theta 2$
3	0	a3	0	$\theta 3$
4	0	a4	0	$\theta 4\text{-}90°$
5	-90°	0	d5	$\theta 5$
6	0	0	0	$\theta 6$

Relations géométriques

$$T_{0-1}=\begin{bmatrix}\cos(\theta_1) & -\sin(\theta_1) & 0 & 0\\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0\\ 0 & 0 & 1 & d_1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$T_{1-2}=\begin{bmatrix}\cos(\theta_2) & 0 & \sin(\theta_2) & 0\\ \sin(\theta_2) & 0 & -\cos(\theta_2) & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$T_{2-3}=\begin{bmatrix}\cos(\theta_3) & -\sin(\theta_3) & 0 & a_3\cos(\theta_3)\\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3\sin(\theta_3)\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$T_{3-4} = \begin{bmatrix} \sin(\theta_4) & \cos(\theta_4) & 0 & a_4\cos(\theta_4) \\ -\cos(\theta_4) & \sin(\theta_4) & 0 & -a_4\sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{4-5} = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{5-6} = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T0_6= T0_1*T1_2*T2_3*T3_4*T4_5*T5_6

A partir de la matrice de transformation homogène T0_6, on en déduit le modèle géométrique.

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

*dx= d5*sin(θ1 + θ2) + a4*cos(θ1 + θ2)*sin(θ3 + θ4) + a3*cos(θ1 + θ2)*cos(θ3)*

*dy= a4*sin(θ1 + θ2)*sin(θ3 + θ4) - d5*cos(θ1 + θ2) + a3*sin(θ1 + θ2)*cos(θ3)*

*dz=d1 - a4*cos(θ3 + θ4) + a3*sin(θ3)*

nx=cos(θ6)(cos(θ1 + θ2)*cos(θ3 + θ4)*sin(θ5) + cos(θ1 + θ2)*sin(θ3 + θ4)*cos(θ5)) - sin(θ1 + θ2)*sin(θ6)*

ny=cos(θ6)(cos(θ3 + θ4)*sin(θ1 + θ2)*sin(θ5) + sin(θ1 + θ2)*sin(θ3 + θ4)*cos(θ5)) + cos(θ1 + θ2)*sin(θ6)*

*nz=-cos(θ3 + θ4 + θ5)*cos(θ6)*

ox=- sin(θ6)(cos(θ1 + θ2)*cos(θ3 + θ4)*sin(θ5) + cos(θ1 + θ2)*sin(θ3 + θ4)*cos(θ5)) - sin(θ1 + θ2)*cos(θ6)*

*oy=cos(θ1 + θ2)*cos(θ6) - sin(θ6)*(cos(θ3 + θ4)*sin(θ1 + θ2)*sin(θ5) + sin(θ1 + θ2)*sin(θ3*

*+ θ4)*cos(θ5))*

*oz=cos(θ3 + θ4 + θ5)*sin(θ6)*

*ax=cos(θ3 + θ4 + θ5)*cos(θ1 + θ2)*

*ay=cos(θ3 + θ4 + θ5)*sin(θ1 + θ2)*

az=sin(θ3 + θ4 + θ5)

2.3 Conception du bras manipulateur

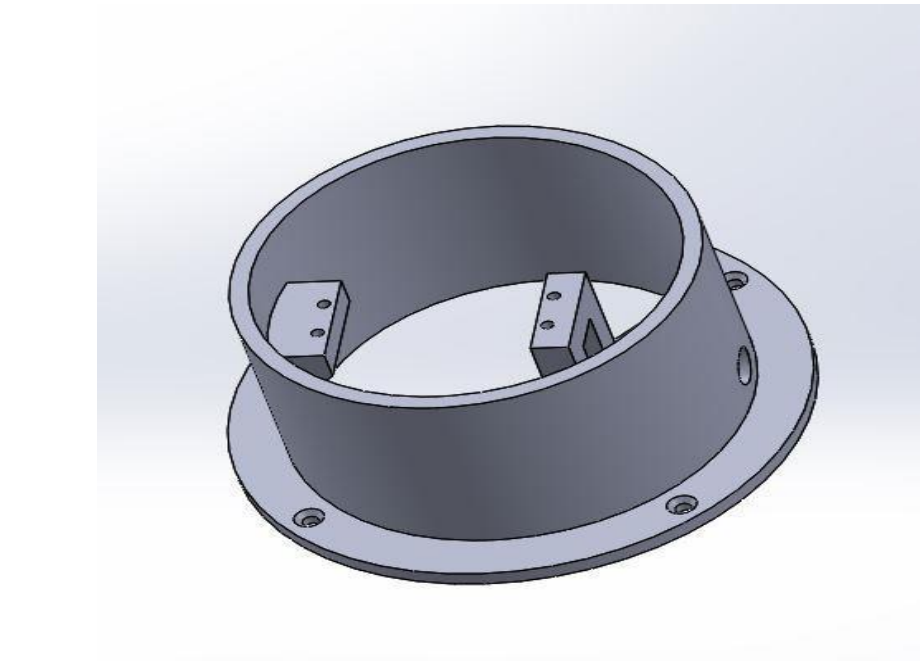
La conception du bras manipulateur a été réalisée à l'aide du logiciel SolidWorks, développé par Dassault Systèmes (DSS). SolidWorks, un outil de conception assistée par ordinateur (CAO), permet de créer des modèles solides en 2D et 3D de manière intuitive et sans complexité.



Figure 2. 4 Environnement du Solidworks

Eléments mécaniques du bras [3]

La base



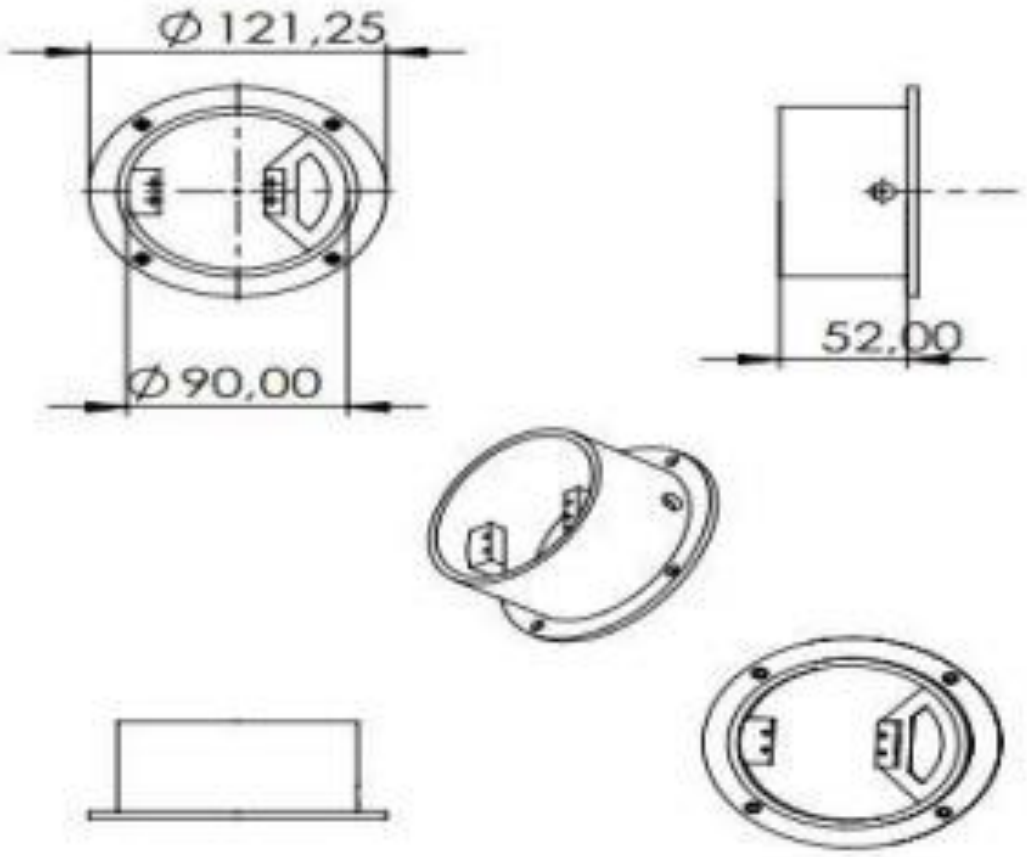


Figure 2. 5 La base

Levier 1

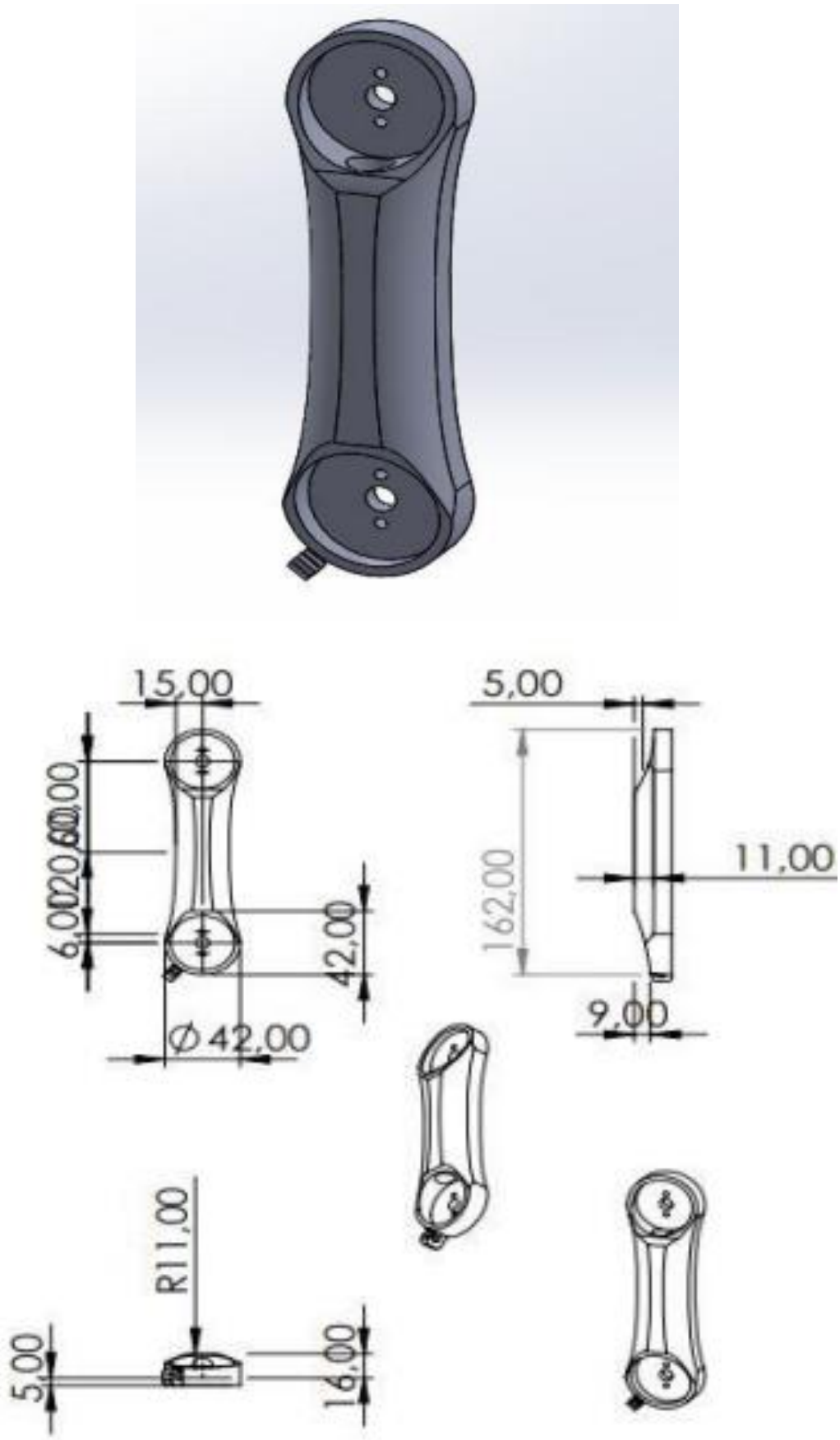


Figure 2. 6 le levier 1

Le levier 2

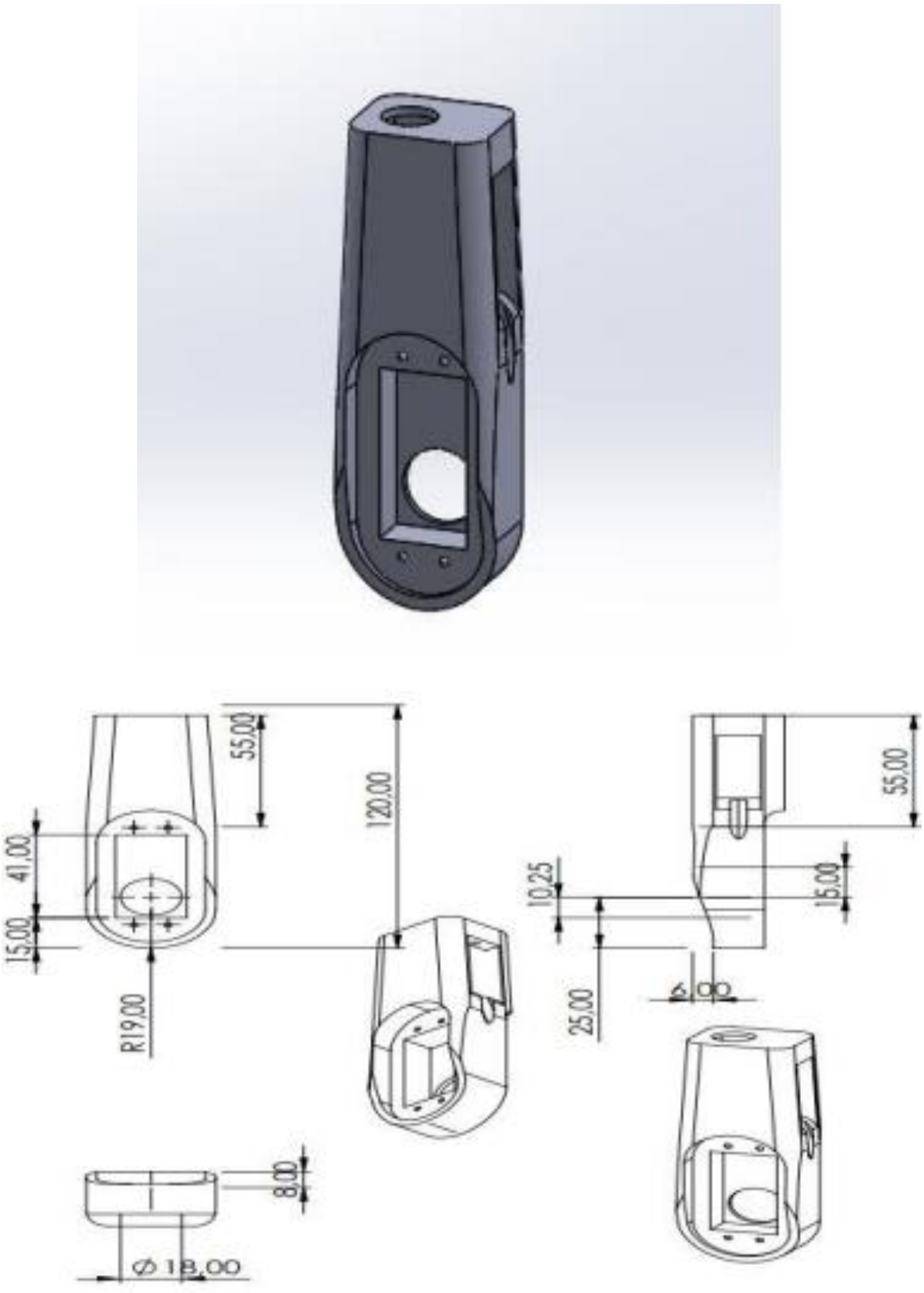
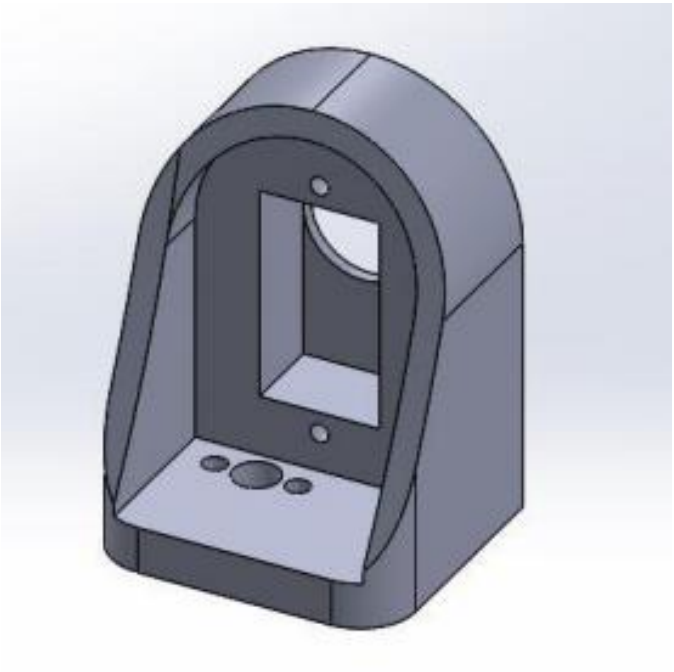


Figure 2. 7 le levier 2

Le levier 2



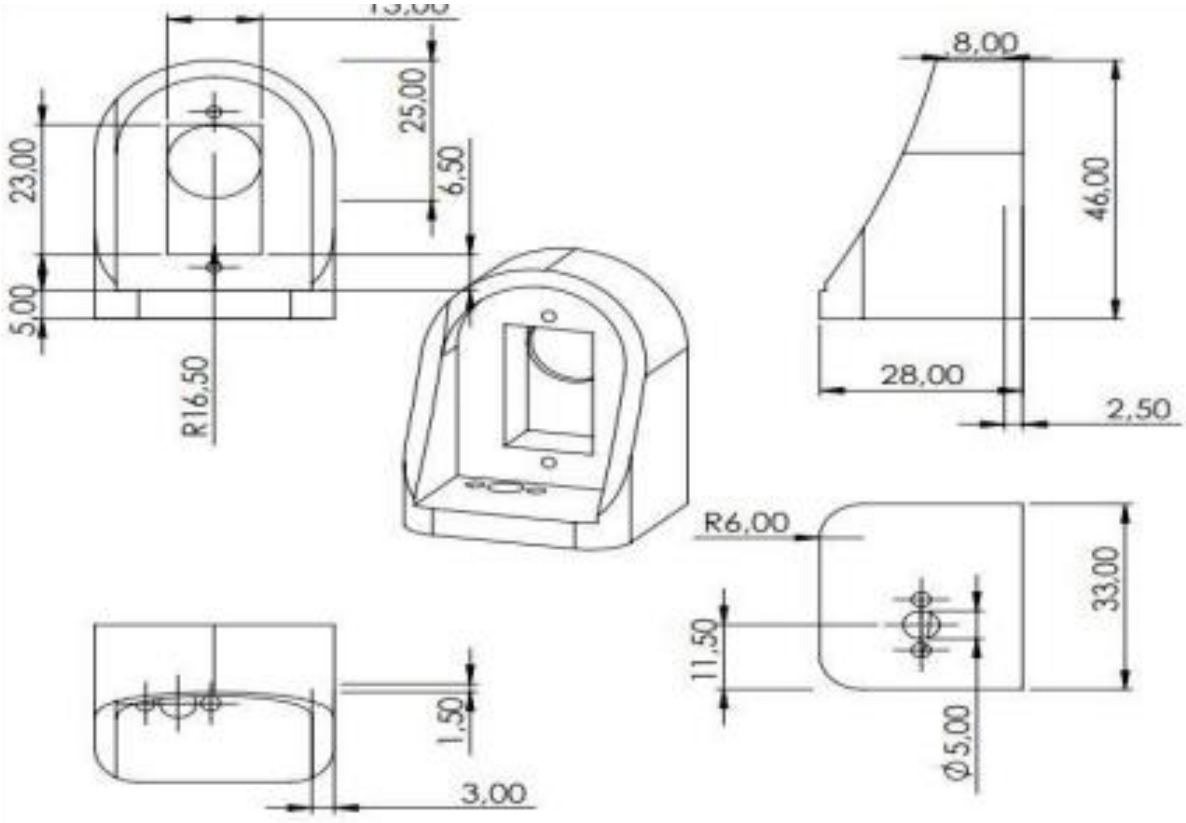


Figure 2. 8 le levier 3

L’organe terminale (pince)

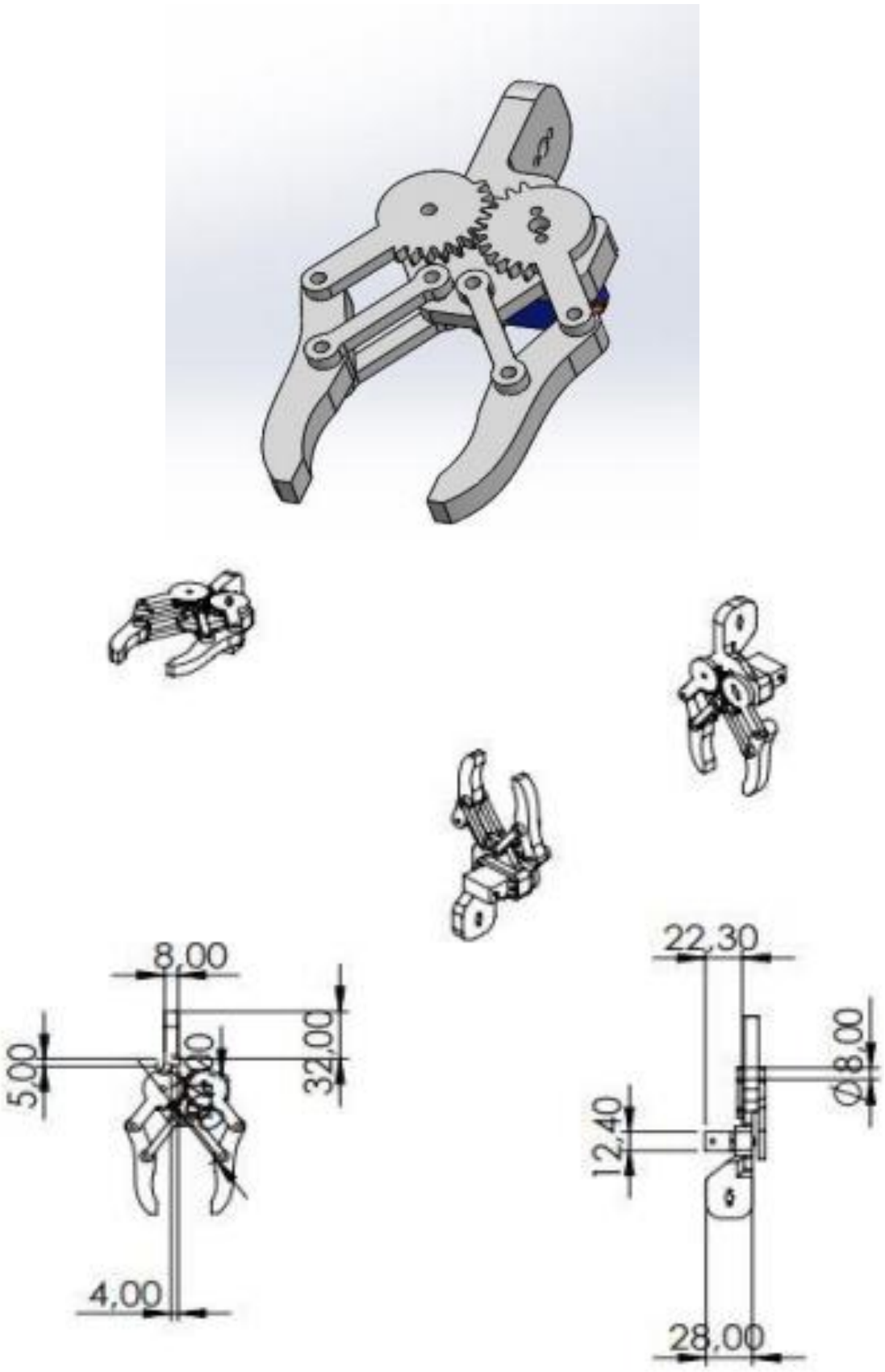


Figure 2. 9 Organe terminal

2.8 Fabrication des pièces

L'impression 3D, également connue sous le nom de fabrication additive, est une technologie révolutionnaire qui permet de créer des objets tridimensionnels en ajoutant de la matière couche par couche. Cette méthode est particulièrement avantageuse pour la fabrication des pièces plastiques des robots, offrant flexibilité, précision et rapidité. Dans ce chapitre, nous aborderons le processus de fabrication des pièces plastiques du robot en utilisant une imprimante 3D, les matériaux utilisés, et les étapes de préparation et d'impression.

2.8.1 Principe de l'impression 3D

L'impression 3D repose sur le principe de l'ajout de matière, où un objet est construit en superposant des couches successives de matériau. Ce processus commence par la création d'un modèle numérique 3D à l'aide de logiciels de conception assistée par ordinateur (CAO) tels que SolidWorks ou Autodesk Fusion 360. Le modèle est ensuite exporté au format STL, qui est compatible avec les logiciels d'impression 3D.

2.8.2 Matériaux utilisés

Pour la fabrication des pièces plastiques du robot, plusieurs types de matériaux peuvent être utilisés. Les plus courants sont :

PLA (Acide Polylactique) : Un plastique biodégradable fabriqué à partir de ressources renouvelables comme l'amidon de maïs. Il est facile à imprimer et offre une bonne résistance mécanique.

ABS (Acrylonitrile Butadiène Styrène) : Un plastique robuste et durable, idéal pour les pièces nécessitant une grande résistance à l'impact.

PETG (Polyéthylène Téréphtalate Glycol) : Un matériau offrant un bon compromis entre PLA et ABS, avec une bonne résistance mécanique et chimique.

2.8.3 Processus de fabrication

2.8.3.1 Préparation du modèle 3D

La première étape consiste à concevoir les pièces du robot en utilisant un logiciel de CAO. Une fois la conception terminée, le modèle 3D est exporté au format STL. Ce fichier est ensuite importé dans un logiciel de tranchage (slicer) comme Cura ou PrusaSlicer, qui convertit le modèle en couches successives et génère le code G-code nécessaire à l'imprimante 3D.

2.8.3.2 Configuration de l'imprimante 3D

Avant de commencer l'impression, il est important de configurer correctement l'imprimante 3D. Cela inclut le nivellement du plateau d'impression, le chargement du filament, et le réglage des paramètres d'impression tels que la température de la buse et du plateau, la vitesse d'impression, et l'épaisseur des couches.

2.8.3.3 Impression des pièces

Une fois l'imprimante configurée et le fichier G-code chargé, l'impression peut commencer. L'imprimante dépose le filament fondu couche par couche pour construire la pièce. Ce processus peut varier en durée en fonction de la complexité et de la taille de la pièce. Pendant l'impression, il est crucial de surveiller le processus pour éviter les erreurs et assurer une qualité optimale.

2.8.3.4 Post-traitement

Après l'impression, les pièces peuvent nécessiter un post-traitement. Cela peut inclure le retrait des supports, le ponçage, le collage de différentes pièces, ou l'application de finitions comme la peinture ou le vernissage pour améliorer l'apparence et la durabilité des pièces.

2.8.4 Avantages et limitations de l'impression 3D

Avantages

- Personnalisation : Permet de fabriquer des pièces sur mesure adaptées aux besoins spécifiques du projet.
- Rapidité : Réduit le temps de fabrication par rapport aux méthodes traditionnelles.
- Coût : Diminue les coûts de production pour les petites séries ou les prototypes.

Limitations

- Taille des pièces : Limitée par les dimensions de l'imprimante 3D.
- Résistance mécanique : Les pièces imprimées peuvent ne pas être aussi robustes que celles fabriquées par des méthodes traditionnelles.
- Qualité de surface : Peut nécessiter des post-traitements pour obtenir une finition lisse.

2.9 Conclusion

Dans ce chapitre, nous avons exploré la modélisation du robot en détaillant les modèles géométriques, offrant ainsi une compréhension approfondie des principes fondamentaux de la modélisation robotique pour des applications avancées. De plus, nous avons exposé les pièces conçues avec le logiciel SolidWorks. Enfin, nous avons décrit le

processus de fabrication des pièces par enlèvement de matière, soulignant l'importance de ce procédé dans la réalisation des composants du robot.

Chapitre 3 : Système de commande du bras manipulateur

3.1 Introduction

Le présent projet porte sur l'étude et le développement d'un bras manipulateur commandé par une carte Arduino Uno et un module Bluetooth HC-05. Dans ce chapitre, nous nous concentrerons sur les composants et équipements électroniques nécessaires à la mise en œuvre et à la conception de notre projet. Nous discuterons également de la sélection de ces composants en fonction de leurs caractéristiques spécifiques.

3.2 Carte ARDUINO

La carte ARDUINO sert de cerveau à notre robot en fauteuil roulant. Il s'agit d'une carte microcontrôleur open source. Elle est largement utilisée dans les milieux éducatifs et dans les projets professionnels en raison de sa nature conviviale et de son vaste soutien communautaire.

Les modules originaux pour chaque version d'Arduino sont fournis par la société italienne Smart Projects. Certaines cartes de marque Arduino sont fabriquées par la société américaine SparkFun Electronics. Il existe à ce jour des versions de cartes de type Arduino.

Pour notre projet, nous avons choisi la carte Arduino Uno.

3.2.1 Carte ARDUINO UNO

L'Arduino Uno sert de cerveau à notre robot en fauteuil roulant. Il s'agit d'une carte microcontrôleur open source basée sur le microcontrôleur Microchip ATmega328P. L'Arduino Uno est largement utilisé dans les contextes éducatifs et les projets professionnels en raison de sa nature conviviale et de son vaste soutien communautaire.

3.2.2 Caractéristiques

La carte Arduino Uno est basée sur un microcontrôleur ATmega328 et est équipée des fonctionnalités suivantes.

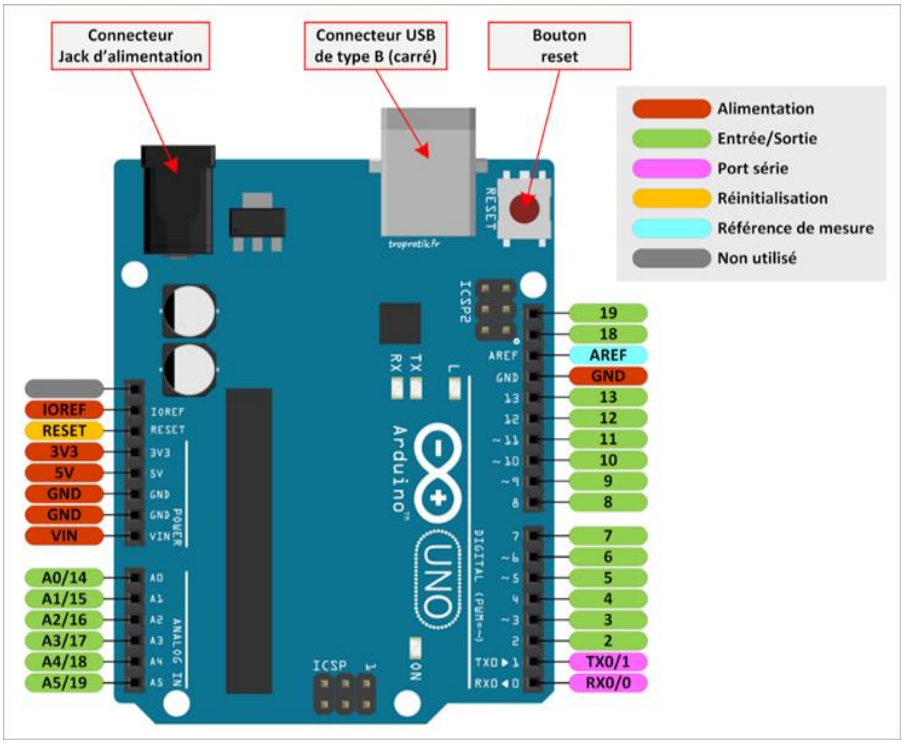


Figure 3. 1 carte Arduino

Tableau 3. 1 Caractéristiques de la carte ARDUINO

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5v)	40 mA (ATTENTION : 200mA cumule pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie3.	50 mA
Intensité maxi disponible pour la sortie 5v	Fonction de l'alimentation utilisée - 500mA max si port USB utilise seul
Mémoire programme Flash	3v 32 KB (ATmega328) dont 0.5 KB sont utilisés par le boot loader
Mémoire SRAM (mémoire volatile)	2 KB(ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

3.2.3 Arduino IDE

Arduino est un environnement de développement intégré (IDE) permettant de créer des programmes pour les cartes Arduino. Cet IDE, fonctionnant sur votre système, permet de développer des "croquis" (programmes en langage Arduino) pour diverses cartes Arduino. Le langage de programmation utilisé par Arduino, appelé Processing, est très simple et ressemble beaucoup au langage C.

Après avoir écrit le croquis dans l'IDE Arduino, il est nécessaire de le télécharger sur la carte Arduino afin de l'exécuter. Les programmes Arduino s'exécutent de manière séquentielle, c'est-à-dire que les instructions sont exécutées l'une après l'autre.

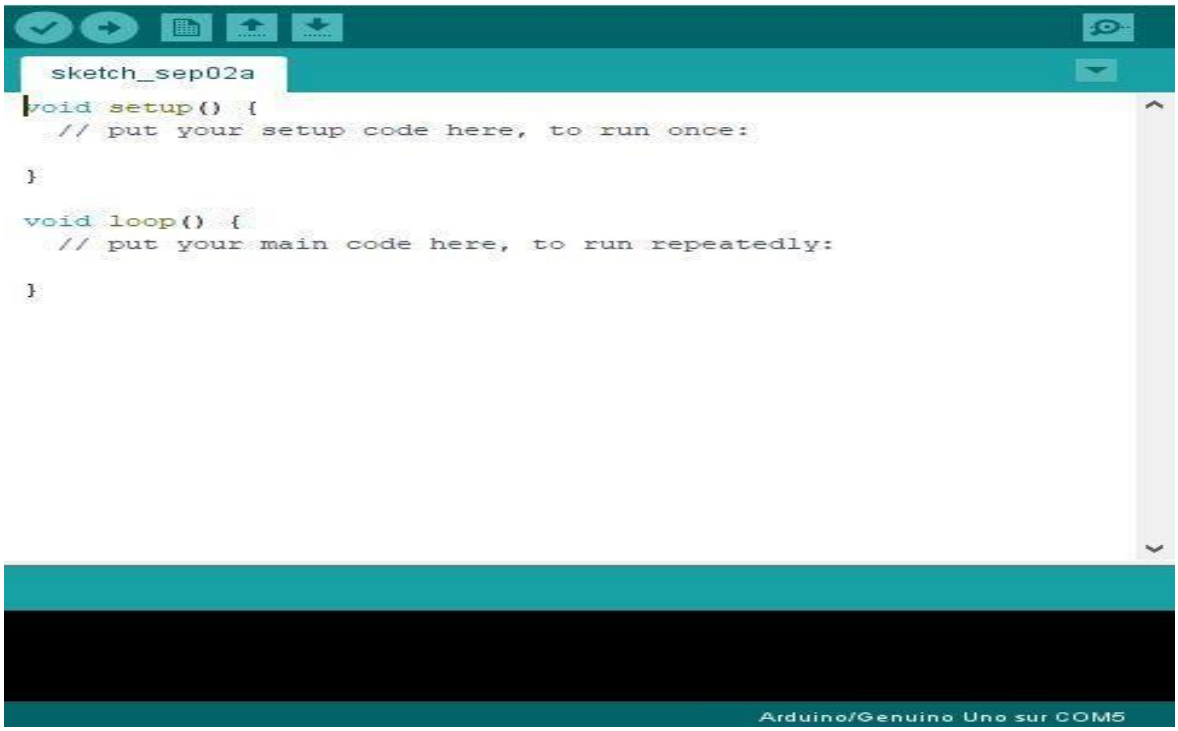


Figure 3. 2 : Interface IDE

3.2.4 Actionneurs (Servomoteurs)

Un servomoteur est un moteur à courant continu équipé d'un axe de rotation, d'un capteur de position angulaire et d'une carte électronique. Il est utilisé pour réguler la position de l'axe et contrôler le moteur. Les servomoteurs se distinguent par leur précision et leur puissance accrues. Ils fonctionnent en utilisant un système de boucle fermée, ce qui leur permet de maintenir une position prédéfinie selon les consignes données.

Avantages des servomoteurs

L'un des principaux atouts des servomoteurs réside dans leur capacité à tourner à grande vitesse tout en maintenant un couple nominal constant. Ils peuvent également

supporter jusqu'à près de deux fois leur couple nominal pendant une courte durée. Cependant, il est crucial de gérer l'échauffement qui en résulte en dissipant la chaleur. Les codeurs intégrés aux servomoteurs garantissent leur précision et leur fiabilité.

Inconvénients des servomoteurs

Les servomoteurs peuvent être coûteux et nécessitent des codeurs sophistiqués. Une configuration incorrecte du codeur peut entraîner des besoins de maintenance supplémentaires. Pour utiliser des vitesses de rotation faibles ou augmenter le couple disponible, il est possible d'ajouter un réducteur de vitesse aux servomoteurs. Cependant, cela ajoute une dimension de complexité au système, augmentant ainsi le coût.

Dans ce projet, les servomoteurs utilisés sont :

MG996R Servomoteur

SG90 Micro Servomoteur



MG996R Servomoteur



SG90 Micro Servomoteur

Paramètres techniques des servomoteurs

Références / Spécifications	SG90	MG996R
Poids	9g	55 g
Couple d'arrêt	1.8 kgf ·cm	9.4 kgf ·cm (4.8 V), 11 kgf ·cm (6 V)
Tension de fonctionnement	4.8 V (~5V)	4.8 V à 7.2 V
Vitesse	0.1 s/60°	0.17 s/60o (4.8 V) ,0.14 s/60o (6V).
Rotation angle	180°	180°

3.2.5 Module Bluetooth HC-05

Le module Bluetooth HC-05 permet d'ajouter une connectivité Bluetooth à vos projets à microcontrôleur via une communication série TTL. Ce module est livré avec un cordon de raccordement à 6 broches.



Figure 3. 3 : Module Bluetooth HC-05

Caractéristiques techniques :

- **Alimentation :** 3,6 à 6 Vcc (régulateur 3 Vcc intégré)
- **Version Bluetooth :** 2.0
- **Portée :** jusqu'à 10 mètres
- **Liaison série :** de 4800 à 1382400 bauds
- **Antenne :** intégrée, 2,4 GHz
- **Sorties :** Key, Vcc, Gnd, TXD, RXD, State
- **Configuration :** via commandes AT
- **Longueur du cordon :** environ 20 cm
- **Dimensions :** 35 x 16 x 12 mm

Ce module est idéal pour intégrer une communication sans fil Bluetooth dans vos projets, offrant une portée efficace et une configuration simple via des commandes AT.

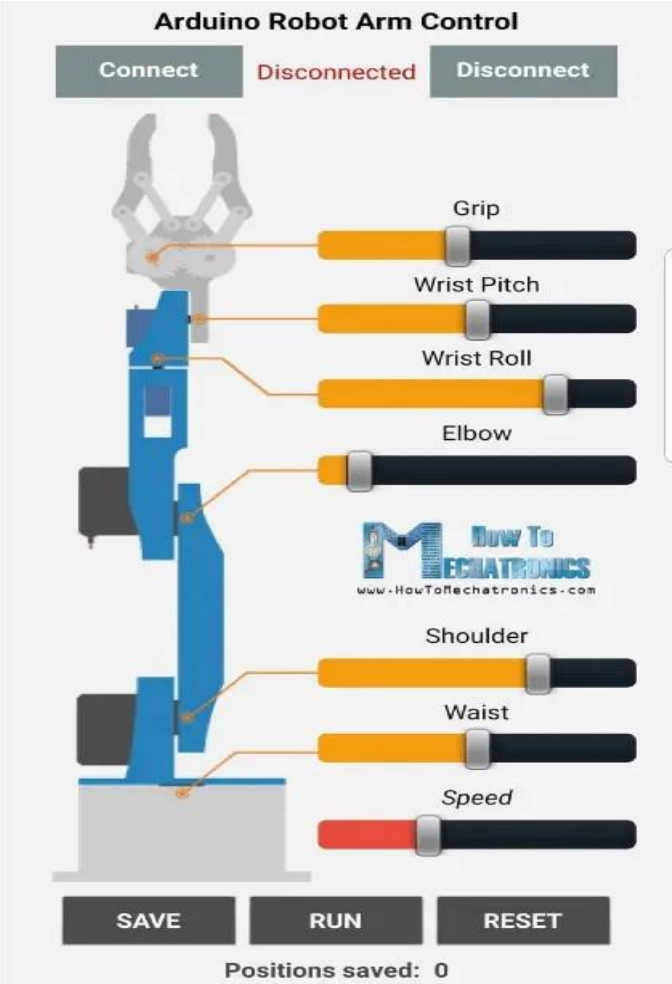


Figure 2. 10 : Commande des moteurs du bras [1]

3.2.6 Schéma du système électrique

Le schéma de circuit de ce projet est en réalité assez simple. Nous avons besoin d'une carte Arduino UNO et d'un module Bluetooth HC-05 pour communiquer avec le smartphone. Les broches de commande des six servomoteurs sont connectées aux six broches numériques de la carte Arduino. (voir la figure 2.14)

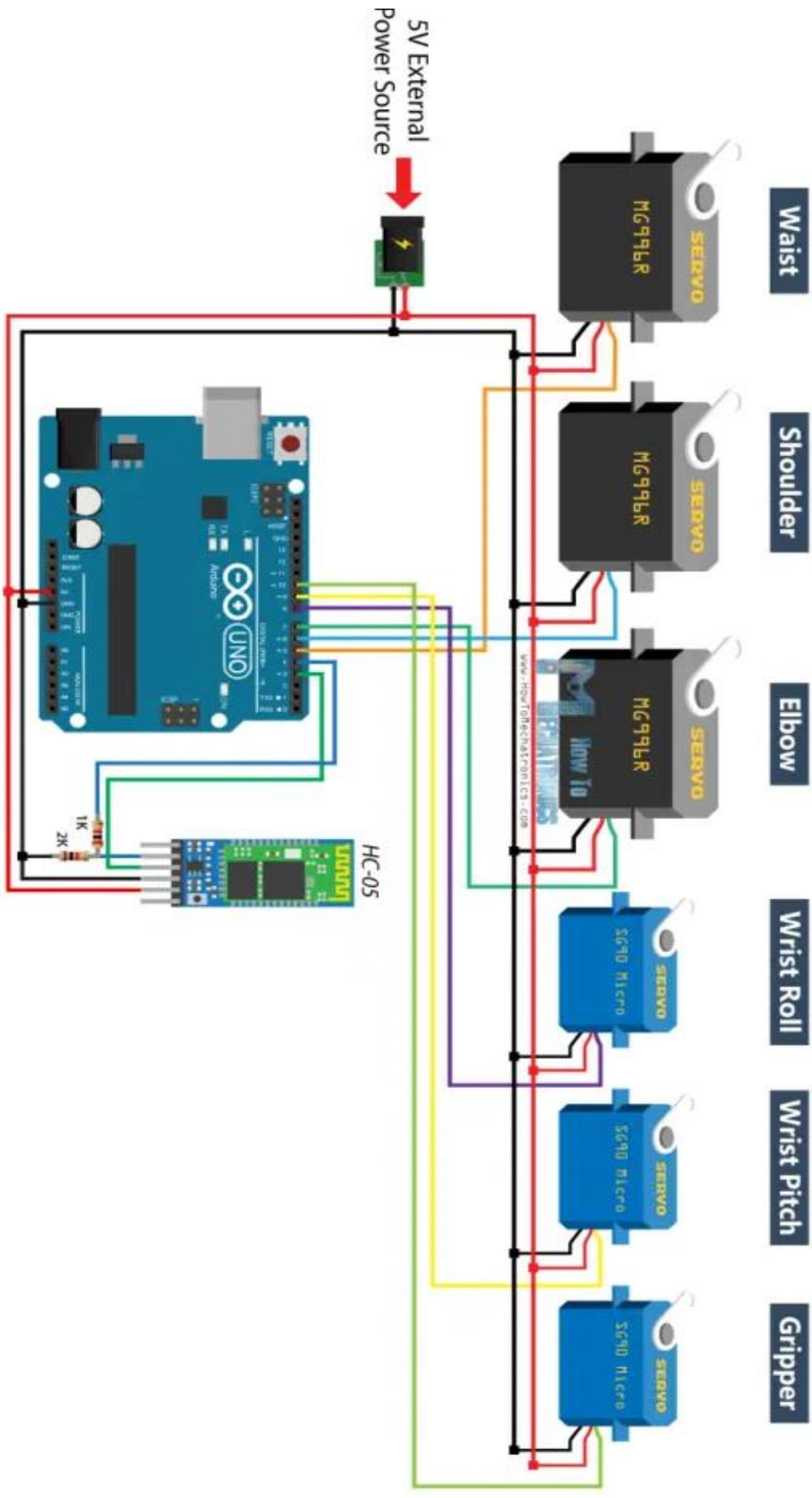


Figure 3. 4 : Schéma du circuit électrique [1]



Figure 3. 5 Bras manipulateur réalisé

3.5 Conclusion

Dans ce chapitre, nous avons exploré les différents aspects matériels et logiciels essentiels à la mise en œuvre de notre projet. Nous avons détaillé les composants individuels de notre bras manipulateur et décrit comment chaque articulation est contrôlée. L'utilisation de la carte Arduino Uno permet de gérer efficacement les articulations du bras à l'aide de Module Bluetooth, démontrant ainsi la simplicité et la flexibilité de notre approche.

.

Conclusion Générale

Dans un contexte où l'automatisation et la précision sont devenues essentielles dans divers secteurs industriels, les robots articulés occupent une place cruciale. Ces systèmes mécaniques sophistiqués, animés par des actionneurs et pilotés par des calculateurs avancés, sont conçus pour accomplir une vaste gamme de tâches, depuis la manipulation minutieuse d'objets fragiles jusqu'à l'assemblage de haute précision en milieu industriel.

Ce mémoire s'est focalisé sur l'étude approfondie d'un robot articulé spécifique doté de six articulations rotoïdes, offrant ainsi six degrés de liberté. Cette configuration confère une grande flexibilité et une précision remarquable lors de l'exécution de ses fonctions. Structuré en trois chapitres, ce travail vise à fournir une compréhension exhaustive du domaine des bras manipulateurs, depuis leur conception jusqu'à leur mise en œuvre pratique.

Nous avons exploré la modélisation des robots en abordant les aspects géométriques, cinématiques et dynamiques. Cette approche intégrée fournit une base solide pour la compréhension des principes fondamentaux de la robotique, essentielle pour des applications et des analyses avancées.

De plus, nous avons examiné en détail les composants matériels et logiciels nécessaires à la réalisation de notre projet. Chaque articulation du bras manipulateur a été soigneusement décrite, tout en mettant en évidence le contrôle précis réalisé à l'aide d'une carte Arduino Uno et d'un module Bluetooth. Cette combinaison technologique illustre la simplicité et la flexibilité de notre approche, ouvrant ainsi la voie à des innovations futures dans le domaine de la robotique.

Références

[1] https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/

[2] https://smartsurgicalrobot.wixsite.com/robo/about-robotic-arm

[3] Djebarni, D. Hamdi, A(2021). *Etude et conception d'un bras de robot* (Master dissertation, Faculté des Sciences et Technologies).

[4] Philippe Coiffet, "La robotique: principes et applications", Edition Hermès, 1986.

[5] Etienne Dombre et Wissama Khalil, "Modélisation et commande des robots", Edition Hermès Paris, 1988.

Annexe : Programme [1]

```
#include <SoftwareSerial.h>
#include <Servo.h>

Servo servo01;
Servo servo02;
Servo servo03;
Servo servo04;
Servo servo05;
Servo servo06;

SoftwareSerial Bluetooth(3, 4);

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos;
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos;
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50], servo06SP[50];
int speedDelay = 20;
int index = 0;
String dataIn = "";

void setup() {
  servo01.attach(5);
  servo02.attach(6);
  servo03.attach(7);
  servo04.attach(8);
  servo05.attach(9);
  servo06.attach(10);
  Bluetooth.begin(38400);
  Bluetooth.setTimeout(1);
  delay(20);
  servo1PPos = 90;
  servo01.write(servo1PPos);
  servo2PPos = 150;
  servo02.write(servo2PPos);
  servo3PPos = 35;
  servo03.write(servo3PPos);
  servo4PPos = 140;
  servo04.write(servo4PPos);
  servo5PPos = 85;
  servo05.write(servo5PPos);
  servo6PPos = 80;
  servo06.write(servo6PPos);
}

void loop() {

  if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString();


    if (dataIn.startsWith("s1")) {
```

```
  String dataInS = dataIn.substring(2, dataIn.length());

  if (servo1PPos > servo1Pos) {
    for ( int j = servo1PPos; j >= servo1Pos; j--) {
      servo01.write(j);
      delay(20);
    }
  }

  if (servo1PPos < servo1Pos) {
    for ( int j = servo1PPos; j <= servo1Pos; j++) {
      servo01.write(j);
      delay(20);
    }
  }
  servo1PPos = servo1Pos;
}


if (dataIn.startsWith("s2")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo2Pos = dataInS.toInt();

  if (servo2PPos > servo2Pos) {
    for ( int j = servo2PPos; j >= servo2Pos; j--) {
      servo02.write(j);
      delay(50);
    }
  }
  if (servo2PPos < servo2Pos) {
    for ( int j = servo2PPos; j <= servo2Pos; j++) {
      servo02.write(j);
      delay(50);
    }
  }
  servo2PPos = servo2Pos;
}

if (dataIn.startsWith("s3")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo3Pos = dataInS.toInt();
  if (servo3PPos > servo3Pos) {
    for ( int j = servo3PPos; j >= servo3Pos; j--) {
      servo03.write(j);
      delay(30);
    }
  }
  if (servo3PPos < servo3Pos) {
    for ( int j = servo3PPos; j <= servo3Pos; j++) {
      servo03.write(j);
      delay(30);
    }
  }
```

```
  servo3PPos = servo3Pos;
 }

 if (dataIn.startsWith("s4")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo4Pos = dataInS.toInt();
  if (servo4PPos > servo4Pos) {
   for ( int j = servo4PPos; j >= servo4Pos; j--) {
    servo04.write(j);
    delay(30);
   }
  }
  if (servo4PPos < servo4Pos) {
   for ( int j = servo4PPos; j <= servo4Pos; j++) {
    servo04.write(j);
    delay(30);
   }
  }
  servo4PPos = servo4Pos;
 }

 if (dataIn.startsWith("s5")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo5Pos = dataInS.toInt();
  if (servo5PPos > servo5Pos) {
   for ( int j = servo5PPos; j >= servo5Pos; j--) {
    servo05.write(j);
    delay(30);
   }
  }
  if (servo5PPos < servo5Pos) {
   for ( int j = servo5PPos; j <= servo5Pos; j++) {
    servo05.write(j);
    delay(30);
   }
  }
  servo5PPos = servo5Pos;
 }

 if (dataIn.startsWith("s6")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo6Pos = dataInS.toInt();
  if (servo6PPos > servo6Pos) {
   for ( int j = servo6PPos; j >= servo6Pos; j--) {
    servo06.write(j);
    delay(30);
   }
  }
  if (servo6PPos < servo6Pos) {
   for ( int j = servo6PPos; j <= servo6Pos; j++) {
    servo06.write(j);
    delay(30);
   }
```

```
    }
    servo6PPos = servo6Pos;
   }

   if (dataIn.startsWith("SAVE")) {
    servo01SP[index] = servo1PPos;
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++;
   }
   // If button "RUN" is pressed
   if (dataIn.startsWith("RUN")) {
    runservo();
   }
   if ( dataIn == "RESET") {
    memset(servo01SP, 0, sizeof(servo01SP));
    memset(servo02SP, 0, sizeof(servo02SP));
    memset(servo03SP, 0, sizeof(servo03SP));
    memset(servo04SP, 0, sizeof(servo04SP));
    memset(servo05SP, 0, sizeof(servo05SP));
    memset(servo06SP, 0, sizeof(servo06SP));
    index = 0;
   }
  }
}


void runservo() {
  while (dataIn != "RESET") {
   for (int i = 0; i <= index - 2; i++) {
    if (Bluetooth.available() > 0) {
     dataIn = Bluetooth.readString();
     if ( dataIn == "PAUSE") {
      while (dataIn != "RUN") {
       if (Bluetooth.available() > 0) {
        dataIn = Bluetooth.readString();
        if ( dataIn == "RESET") {
         break;
        }
       }
      }
     }

     if (dataIn.startsWith("ss")) {
      String dataInS = dataIn.substring(2, dataIn.length());
      speedDelay = dataInS.toInt();
     }
    }

    if (servo01SP[i] == servo01SP[i + 1]) {
```

```
}
if (servo01SP[i] > servo01SP[i + 1]) {
  for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
    servo01.write(j);
    delay(speedDelay);
  }
}
if (servo01SP[i] < servo01SP[i + 1]) {
  for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
    servo01.write(j);
    delay(speedDelay);
  }
}


if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
  for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
    servo02.write(j);
    delay(speedDelay);
  }
}
if (servo02SP[i] < servo02SP[i + 1]) {
  for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
    servo02.write(j);
    delay(speedDelay);
  }
}


if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
  for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
    servo03.write(j);
    delay(speedDelay);
  }
}
if (servo03SP[i] < servo03SP[i + 1]) {
  for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
    servo03.write(j);
    delay(speedDelay);
  }
}


if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
  for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
    servo04.write(j);
    delay(speedDelay);
```

```
      }
    }
    if (servo04SP[i] < servo04SP[i + 1]) {
      for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
        servo04.write(j);
        delay(speedDelay);
      }
    }


    if (servo05SP[i] == servo05SP[i + 1]) {
    }
    if (servo05SP[i] > servo05SP[i + 1]) {
      for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
        servo05.write(j);
        delay(speedDelay);
      }
    }
    if (servo05SP[i] < servo05SP[i + 1]) {
      for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
        servo05.write(j);
        delay(speedDelay);
      }
    }


    if (servo06SP[i] == servo06SP[i + 1]) {
    }
    if (servo06SP[i] > servo06SP[i + 1]) {
      for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
        servo06.write(j);
        delay(speedDelay);
      }
    }
    if (servo06SP[i] < servo06SP[i + 1]) {
      for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {
        servo06.write(j);
        delay(speedDelay);
      }
    }
  }
 }
}
```