



République Algérienne Démocratique et Populaire.

Ministre de l'Enseignement Supérieur et de la Recherche Scientifique.

Université d'Oran 2

Institut de Maintenance et de sécurité Industrielle

MÉMOIRE DE FIN D'ÉTUDES

EN VUE DE L'OBTENTION DU DIPLOME

DE MASTER EN ELECTROMECHANIQUE

INDUSTRIELLE

Thème

Etude et conception d'un bras de robot manipulateur

Présentée et soutenue publiquement par :

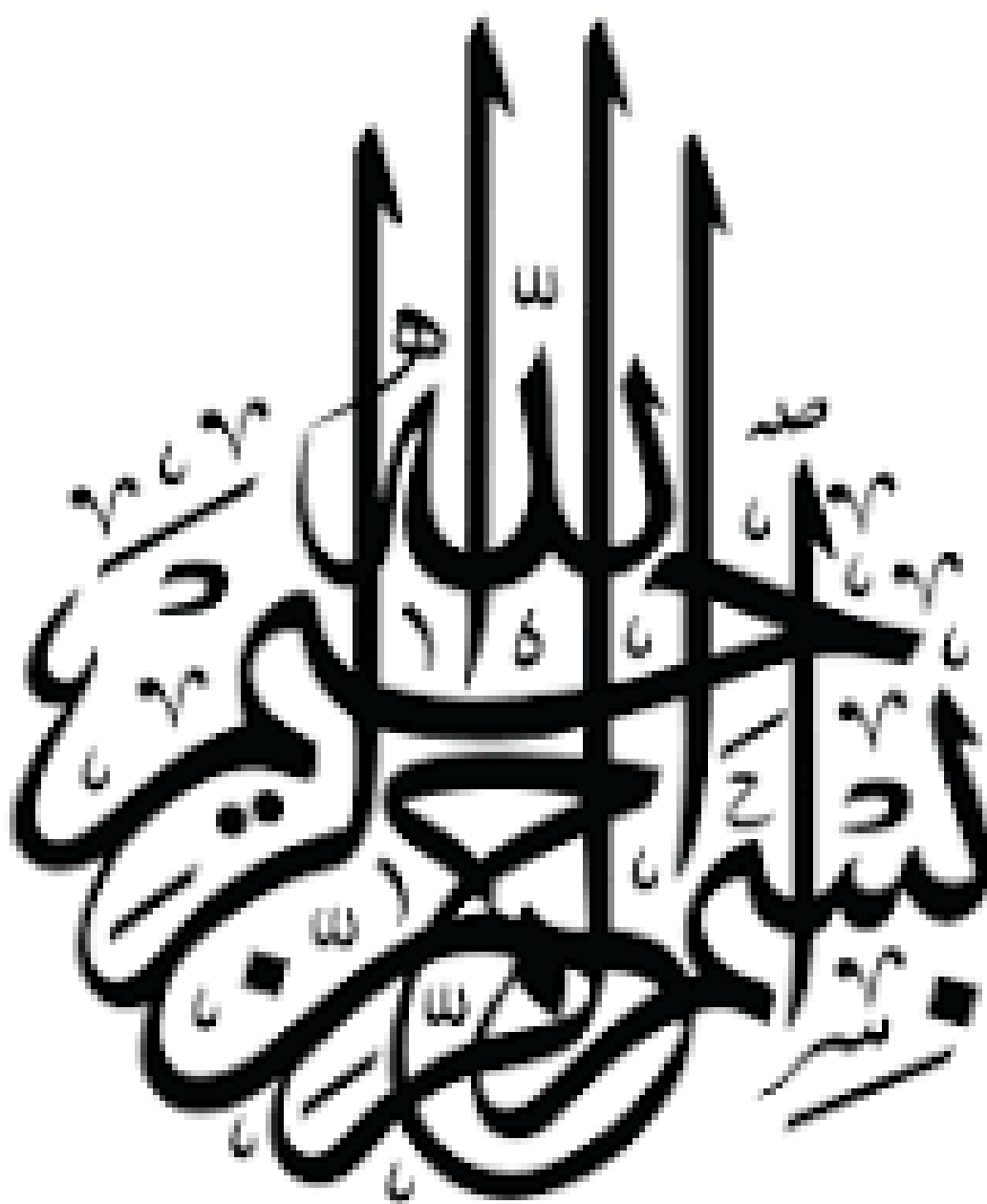
Mr : KLOUCHE Abdelhakim

Mr : HASROUF Walid

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Belabbes Abdallah	MCB	IMSI-Univ. D'Oran2	Président
Adda Neggaz Samir	MAA	IMSI-Univ. D'Oran2	Encadreur
Reguieg Yssaad Sadek	MAA	IMSI-Univ. D'Oran2	Examineur

Année universitaire 2023/2024



REMERCIEMENTS

D'abord nous remercions Dieu notre créateur de nous avoir donné la santé, la force, la volonté et le courage de pouvoir réussir et d'accomplir ce modeste mémoire.

Nous tenons à exprimer nos sincères remerciements et notre reconnaissance à **Mr. Adda Neggaz Samir** pour son encadrement et sa précieuse disponibilité qu'il a bien voulu nous accorder. Son expérience académique, sa rigueur intellectuelle et ses conseils dans le but de réaliser ce travail, nous ont permis de nous développer et de réussir ce projet.

Nous remercions très respectueusement **Mr. Belabbes Abdallah** et **Mr. Reguieg Yssaad Sadek** de nous avoir fait l'honneur de juger notre modeste travail.

On ne saura clôturer ces remerciements sans exprimer toute notre affection pour les enseignants de l'Université d'Oran 2 et Institut de Maintenance et de sécurité Industrielle, qui ont bien voulu partager leur savoir durant nos études.

Enfin, nous remercions tous ceux qui ont contribué de prêt ou de loin à la réalisation de ce projet.

DEDICACES

À mon cher père **KLOUCHE MOHAMMED**, qui a su montrer à ses enfants que le travail est libérateur et qui n'a ménagé aucun effort

À ma chère mère pour ton amour pour moi, pour les sacrifices que tu consens pour rendre tes enfants heureux. Tu as enduré beaucoup de peine pour mon bien-être et à ma réussite. Reçois ceci en guise de ma reconnaissance et que Dieu te garde longtemps afin que tu puisses goûter aux arbres que tu as plantés.

À mon frère aîné **NADHIR KLOUCHE** pour ton dévouement, pour ton écoute et ton soutien. Reçois ici ma profonde gratitude.

À ma sœur Rabiaa et son mari Raouef, votre soutien inconditionnel, et votre présence constante ont été une source inestimable d'inspiration tout au long de ce parcours. Ma profonde gratitude et de mon affection pour vous deux.

Et je tiens à remercier mon binôme **HASROUF Walid**, on a travaillé ensemble depuis notre première année universitaire, et ma profonde gratitude pour son soutien.

Abdelhakim

DEDICACES

Avec l'expression de ma reconnaissance, je dédie ce travail à ceux à qui, quels que soient les termes embrassés, je n'arriverais jamais à exprimer mon amour sincère :

- À l'homme, ma précieuse offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon cher père **Ali**.
- À la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureux : mon adorable mère.
- À mes très chers frères leurs femmes, sœurs et leurs enfants, qui n'ont pas cessé de me conseiller, encourager et soutenir tout au long de mes études, que Dieu les protège et leur offrent la chance et le bonheur.
- À tous les cousin(e)s, les voisin(e)s et les ami(e)s que j'ai connu jusqu'à maintenant. Merci pour leurs encouragements.
- Sans oublier mon compagnon et toutes les personnes qui ont contribué de près ou de loin par leur soutien moral tout au long de ce projet

Walid

Résumé

L'objectif de ce projet est de concevoir et réaliser un bras robotique connecté à une carte électronique Arduino, contrôlé par le développement d'un programme Arduino et d'une application informatique dédiée pour exécuter diverses commandes. La mission de cette machine est de travailler dans des environnements dangereux à la place des humains.

Mots-clés : Contrôle, Conception, Réalisation, Arduino, Bras Robotique

Summary

The objective of this project is to design and implement a robotic arm connected to an Arduino electronic board, controlled by developing an Arduino program and a specialized computer application to execute various commands. The mission of this machine is to operate in hazardous environments in place of humans.

Keywords: Control, Design, Implementation, Arduino, Robotic Arm

ملخص

يهدف هذا المشروع الى تصميم و إنجاز ذراع روبوت يتم توصيلها ببطاقة الكترونية اردوينو والتحكم بها من خلال برنامج اردوينو و التطبيق الخاص الروبوت على مستوى الحاسوب لتنفيذ مختلف الاوامر، تتمثل مهمة هذا الروبوت في العمل بدل الانسان في الاماكن الخطرة.

كلمات مفتاحية : تحكم تصميم انجاز اردوينو ذراع روبوت

SOMMAIRE

Introduction générale	1
Chapitre I : Généralité sur les robots	3
1.1 Introduction	4
1.2 Définition d'un robot	5
1.3 Développement historique des robots manipulateurs	5
1.4 La première machine programmable	6
1.5 Le premier bras télé-opéré	7
1.6 Robot industriel :	8
1.7 Constituants d'un Robot manipulateur :	9
1.8 Les robots manipulateurs	9
1.9 Type de robot manipulateur :	9
1.10 Classification géométrique	13
1.11 Caractéristiques des robots	14
1.12 Caractéristique géométrique.....	15
1.13 Performances en mouvement.....	15
1.14 Bénéfices de leur utilisation	16
1.15 Conclusion	17
Chapitre II : Conception et Architecture des Robots Manipulateurs avec Pince	18
2.1 Introduction	19
2.2 Choix du bras manipulateur étudié :	19
2.3 La carte Arduino :	19
2.3.1 Bref historique de l'Arduino.....	20

2.3.2	Matériel arduino	21
2.3.3	L'utilisation de la carte Arduino UNO :	21
2.3.4	Applications de l'Arduino Uno	23
2.4	Shield Arduino :	25
2.5	Généralités sur les servomoteurs.....	26
2.5.1	Présentation.....	26
2.5.2	Les types de servomoteur :	28
2.6	Le RADAR	29
2.6.1	Introduction radar.....	29
2.6.2	Vue d'ensemble du système	30
2.6.3	Capteurs ultrasoniques HC- SR04	30
2.6.4	Processing	31
2.6.5	Applications :	32
2.7	Source d'énergie	33
2.8	La diode	34
2.8.1	CARACTERISTIQUES	34
2.9	Définition du l'impression 3D	35
2.10	Le prototypage rapide	35
2.11	La modélisation du modèle	36
2.11.1	Définition SOLIDWORKS	36
2.11.2	Historique :	38
2.11.3	Le fonctionnement :	38

2.12	Conclusion	39
	Chapitre III : Cinématique du Bras Manipulateur	40
3.1	Introduction	41
3.2	Modèle géométrique d'un robot en chaine simple :	41
3.2.1	Nécessité d'un modèle :	41
3.2.2	Mécanismes :	42
3.2.3	Cinématiques série :	42
3.2.4	. Chaîne cinématique arborescente	43
3.2.5	Chaîne cinématique fermée.....	44
3.2.6	Morphologie des robots manipulateurs :	44
3.2.7	Position et orientation d'un solide	45
3.3	Modèle géométrique direct	46
3.4	Paramètres Denavit–Hartenberg :	47
3.5	Modèle géométrique indirect	50
3.6	Modélisation cinématique :	51
3.6.1	. Modélisation cinématique direct :	51
3.6.2	Modèle cinématique inverse :	51
3.6.3	Forme générale du modèle cinématique :	52
3.7	Modélisation dynamique :	52
3.8	Application sur le bras robotique :	52
3.8.1	Description du robot à commander	53
3.8.2	Matrice Denavit–Hartenberg :	53

3.8.3	Modèle géométrique direct :	54
3.9	Conclusion	61
	Chapitre IV : Réalisation et Application	62
4.1	Introduction	63
4.2	Simulation	64
4.2.1	Représentation de tinkercad	64
4.2.2	Conception et impression 3D des pièces du bras robotique	67
4.3	Réalisation pratique.....	69
4.3.1	Assemblage du bras manipulateur :	69
4.3.2	Utilisation de la diode	73
4.3.3	Points à vérifier :	74
4.3.4	Assemblage complet du bras manipulateur	74
4.4	Contrôle du bras manipulateur :	76
4.4.1	Programmation d'applications de contrôle :	76
4.4.2	Description de l'application :	77
4.4.3	Programmation Arduino :	79
4.5	Travail du bras manipulateur	80
4.5.1	Planification des Tâches.....	81
4.5.2	Contrôle via la saisie d'angle.....	82
4.6	Conclusion	83
	Conclusion générale	84
	RÉFÉRENCES BIBLIOGRAPHIQUES.....	86
	BIBLIOGRAPHIQUES	87

Sites Web 89

Liste des figures :

Figure 1-1 Boucle de décision	5
Figure 1-2 main robotique a résolu un Rubik's Cube	6
Figure 1-3 Boucle de décision	6
Figure 1-4 Machine à tisser	7
Figure 1-5 Les différents types des robots	8
Figure 1-6 : Vocabulaires du bras de robot.	9
Figure 1-7 Architecture de robot fanuc.....	10
Figure 1-8 le robot qui travaillera dans les usines de BMW aux côtés des humains.....	10
Figure 1-9 L'utilisation de drones pour la maintenance industrielle	11
Figure 1-10 Système de robot de tri de colis largement utilisé utilisant AMR AGV avec rendu basculant 3d	11
Figure 1-11 le robot acrobate hyper agile de Boston Dynamics.....	12
Figure 1-12 Véhicule à Guidage Automatique (AGV) – Les avantages	12
Figure 1-13 L'espace de travail de robot cylindrique	13
Figure 1-14 L'espace de travail de robot rectiligne	13
Figure 1-15 L'espace de travail de robot sphérique.....	14
Figure 1-16 Représentation d'une articulation prismatique.	14
Figure 1-17 Représentation d'une articulation rotoïde	14
Figure 1-18 : Robot dans le secteur de la santé	16
Figure 1-19 : Robots dans le secteur.....	16
Figure 2-1 Pièces constituant le bras manipulateur	19
Figure 2-2 Cartes Arduino	20
Figure 2-3 Différent types des capteurs pour l'arduino	21
Figure 2-4 Différent s actionneurs pour l'arduino	21
Figure 2-5 Schéma de la carte Arduino	23
Figure 2-6 Carte Arduino UNO	23
Figure 2-7 Interface de logiciel Arduino IDE 2.3.2.....	24
Figure 2-8 Détail de barre de bouton	24
Figure 2-9 : Fonctionnalités principales de l'interface Arduino IDE	25
Figure 2-10 les Shields Arduino	26
Figure 2-11 servomoteur SG90.....	27

Figure 2-12schéma d'un servomoteur	27
Figure 2-13 Exemple de signaux en position pour un servomoteur	27
Figure 2-14 Contrôle du servo moteur par signal PWM	27
Figure 2-15 Branchement électrique de servomoteur.....	28
Figure 2-16 Capteur de distance à ultrasons (4 broches).....	30
Figure 2-17 Schéma fonctionnel du système.....	30
Figure 2-18 Capteur à ultrasons.....	31
Figure 2-19 Espace de travail radar	31
Figure 2-20 Software et processing	32
Figure 2-21 Batterie de secours 12 V 7 Ah pour le stockage de l'énergie de secours.....	33
Figure 2-22 Régulateur de tension LM 7805.....	33
Figure 2-23 Fils de raccordement	34
Figure 2-24 La diode 1N4007.....	34
Figure 2-25 imprimante 3D	35
Figure 2-26 L'interface du logiciel	37
Figure 2-27 Les barres d'outils du logiciel Solidworks.....	37
Figure 2-28 La pièce et l'assemblage par Solidworks.....	38
Figure 3-1 Modèle géométrique d'une chaine simple	41
Figure 3-2 Type de chaines cinématiques	42
Figure 3-3 Chaîne cinématique simple ouverte	43
Figure 3-4 Chaîne cinématique simple ouverte	43
Figure 3-5 Chaîne cinématique fermée.....	44
Figure 3-6 des modèles cinématique	45
Figure 3-7 Position d'un solide dans l'espace et son repère associé	46
Figure 3-8 Relation entre le modèle géométrique direct et inverse.....	47
Figure 3-9 : les paramètres de Denavit-Hartenberg.....	48
Figure 3-10 le modèle Denavit-Hertenberg(DH).....	53
Figure 3-11 Représentation selon la convention DH.....	53
Figure 3-12 Calcul de matrices via matlab	55
Figure 4-1 Le logiciel TINKERCAD en mode simulation du projet.....	65
Figure 4-2 Le logiciel TINKERCAD en mode simulation du Vue schéma.....	65
Figure 4-3 Le logiciel TINKERCAD en mode simulation du Liste des composants....	66
Figure 4-4 Le logiciel TINKERCAD en Code de programmation.....	66
Figure 4-5 Pièces de bras via Solidworks.....	67

Figure 4-6 bras robotique en utilisant le logiciel SolidWorks	68
Figure 4-7 imprimante 3D	69
Figure 4-8 les pièces imprimés par l'imprimante 3D	69
Figure 4-9 les pièces électriques	69
Figure 4-10 La première base pour fixer le servomoteur avec des boulons	70
Figure 4-11 Premier corps	70
Figure 4-12 brae inférieur	71
Figure 4-13Le bras supérieur.....	71
Figure 4-14 assemblage à la main.....	72
Figure 4-15 Poignée à main	72
Figure 4-16 Arduino Uno.....	73
Figure 4-17 Sarvomoteur	73
Figure 4-18 : diode soudé	74
Figure 4-19 Assemblage complet du bras manipulateur.....	74
Figure 4-20 Les Différentes Vues du Bras.....	75
Figure 4-21 Code de programmation d'applications en Python.....	76
Figure 4-22 Interface finale de l'application	78
Figure 4-23 Le résultat de l'exécution.	79
Figure 4-24 Arduino Code	79
Figure 4-25 Bras robotique pour effectuer des actions.....	80
Figure 4-26 Quelques manipulations de Pick & Place spécifiques	81
Figure 4-27 Position des mains au démarrage lorsque vous allumez	82
Figure 4-28 Position des mains après avoir entré les coordonnées	82

Liste des tableaux :

Table 1 Caractéristiques de l'Arduino Uno	22
Table 2 Les paramètres de chaque servomoteur	29
Table 3 le tableau de comparaison	50
Table 4 Tableau des paramètres de Denavit-Hertenberg(DH)	54

Introduction générale

Depuis toujours, l'homme a désiré se détacher du travail dans ce qu'il considère comme étant ardu, fatigant et inintéressant. Les avancées technologiques depuis un demi-siècle ont réussi à satisfaire en grande partie ses besoins grâce à deux outils génériques : l'ordinateur pour les tâches mentales et le robot pour les tâches physiques. La découverte des semi-conducteurs a permis au premier de prendre une certaine avance sur le second, car ils sont des matériaux solides capables de trier des informations sans nécessiter de mobilité mécanique. Le deuxième dépend toujours de la mécanique ancienne, des pièces fragiles et mobiles, ainsi que des coûts importants.

Bien que nouvelle, la robotique trouve ses racines dans les civilisations les plus anciennes. Toutes les autres sciences ont peut-être même été créées pour permettre à la robotique de se développer et d'aider l'Homme à créer ce « esclave » qui appliquera ses commandes au doigt et à l'œil et qui le libérera à jamais du travail.

La robotique articulaire a connu des avancées spectaculaires depuis l'Homme préhistorique qui s'est précipité à créer des extensions technologiques de ses membres pour faciliter son travail, comme la massue par exemple, jusqu'au robot manipulateur autonome le plus performant. Effectivement, les robots ont rapidement évolué, offrant de plus en plus de performances en termes de perception, d'autonomie et de capacité de prise de décision. Ces avancées étaient le fruit inévitable des avancées que la mécanique, la microélectronique et l'informatique ont connues. C'est la raison pour laquelle en un temps record, les générations de robots se succèdent afin de voir émerger une branche de la robotique qui vise de plus grandes dimensions : c'est l'ère des robots interactifs.

L'objectif principal de notre travail est de réaliser un robot manipulateur en utilisant une carte électronique Arduino. La tâche principale du robot est de transporter et placer des objets lourds. Ce travail est alors organisé en quatre chapitres de la manière suivante :

Le premier chapitre présente certaines généralités concernant la robotique et les différents types de robots et robots manipulateurs.

Le second chapitre introduit quelques composants du robot comme les capteurs, les actionneurs et d'autres concepts liés à leurs utilisations.

Le troisième chapitre est consacré au microcontrôleur, langages de programmation et en particulier l'IDE de l'Arduino. Le quatrième chapitre traite la partie pratique.

Chapitre I : Généralité sur les robots

1.1 Introduction

Les robots aujourd'hui ont un impact considérable sur de nombreux aspects de la vie moderne, allant de la production industrielle aux soins de santé, le transport et l'exploration de l'espace et des profondeurs de la mer. Demain, les robots seront aussi présents et personnels que les ordinateurs personnels, tout comme maintenant.

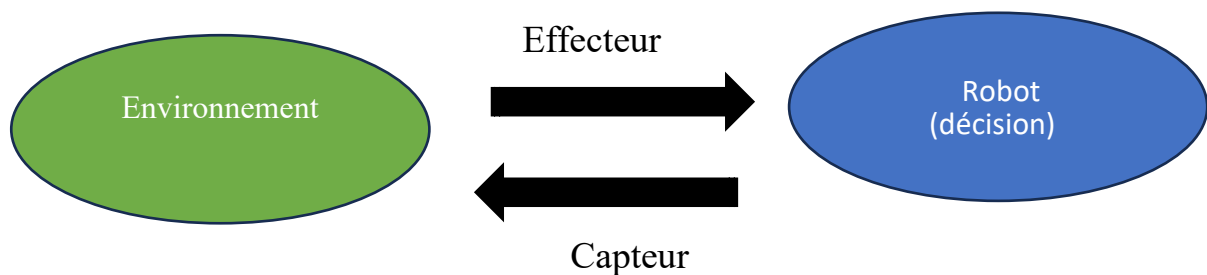
Depuis toujours, l'humanité nourrit le rêve de créer des machines qualifiées et intelligentes. Ce rêve est en train de se concrétiser progressivement dans notre monde contemporain. Dès que les chercheurs ont commencé à conceptualiser des robots, la discipline associée a été nommée "robotique", définie à la fois comme la science et l'art de concevoir et de fabriquer des robots. Les succès rencontrés par les robots industriels, initialement isolés puis intégrés dans des chaînes de production comme des machines parmi d'autres, ont amené le public à élargir la signification du terme "robotique", le rendant presque synonyme d'automatisation. En langage courant, le terme "automatique" aurait dû être l'équivalent de "robotique", mais ce n'est pas le cas. Pour combler cette lacune, les mots "robotisation" et "robotiser" ont été créés pour désigner la plupart des processus d'automatisation et leur mise en œuvre, même en l'absence de véritables robots. De nouveaux termes calqués sur "robotique" ont également émergé dans divers domaines d'application. Par exemple, "domotique" concerne l'automatisation domestique, tandis que "productique" se réfère aux moyens de production automatisés. [21]

De nos jours, la robotique est perçue comme l'art d'automatiser des systèmes de complexité variable, en s'appuyant sur le savoir-faire dérivé des études sur la conception des robots, savoir-faire provenant des avancées dans une branche de l'automatique générale. [22]

En effet, il apparaît rétrospectivement que la structure physique d'une machine ou d'un système n'a pas besoin de correspondre à l'image traditionnelle du robot pour que son contrôle utilise les mêmes composants que ces machines. Ces composants incluent : des capteurs internes pour la régulation, des capteurs externes pour la perception de l'environnement, des actionneurs motorisés, ainsi qu'un système de commande informatique utilisant la programmation et les techniques de l'intelligence artificielle. Ces éléments et leurs actions peuvent être répartis spatialement plutôt que concentrés en une seule entité. De plus, une collaboration entre divers ensembles est possible, aboutissant ainsi à un système robotisé.

1.2 Définition d'un robot

Un robot est un appareil mécanique multi-articulé équipé d'actionneurs et piloté par un contrôleur (calculateur) qui effectue automatiquement une grande diversité de tâches qui sont généralement perçues comme dangereuses, pénibles, répétitives et impossibles pour les êtres humains, ou dans le but d'améliorer son efficacité.



1.3 Développement historique des robots manipulateurs

Les premières solutions techniques des robots se trouvent dans les mécanismes des automates qui comprenaient déjà quelques principes d'autorégulation. Toutefois, la distinction réside dans la programmation : les robots peuvent être reprogrammés et utiliser différentes fonctions en fonction de la tâche assignée, tandis que l'automate est simplement animé par un mécanisme intérieur qui garantit son mouvement et restreint sa capacité de programmation. D'un autre côté, leur but est tout à fait différent, le robot est une machine industrielle, alors que la création de l'automate repose sur une élaboration ludique, voire philosophique : la quête du mouvement constant. [21]

L'histoire des robots a ainsi traversé les étapes suivantes :

En 1947 : L'apparition de premier manipulateur électrique télé-opéré.

En 1970 : les premiers robots industriels.

En 1996 : C'est un robot qui sert à effectuer des mesures et prélever des échantillons.

En 2009 : Cela permet d'aider les animaux et les humains à faire fonctionner leurs membres paralysés ou manquants. [22]

En 2019 : l'objectif était de résoudre un rubik puis induire à faire des tâches plus complexe (open ia)

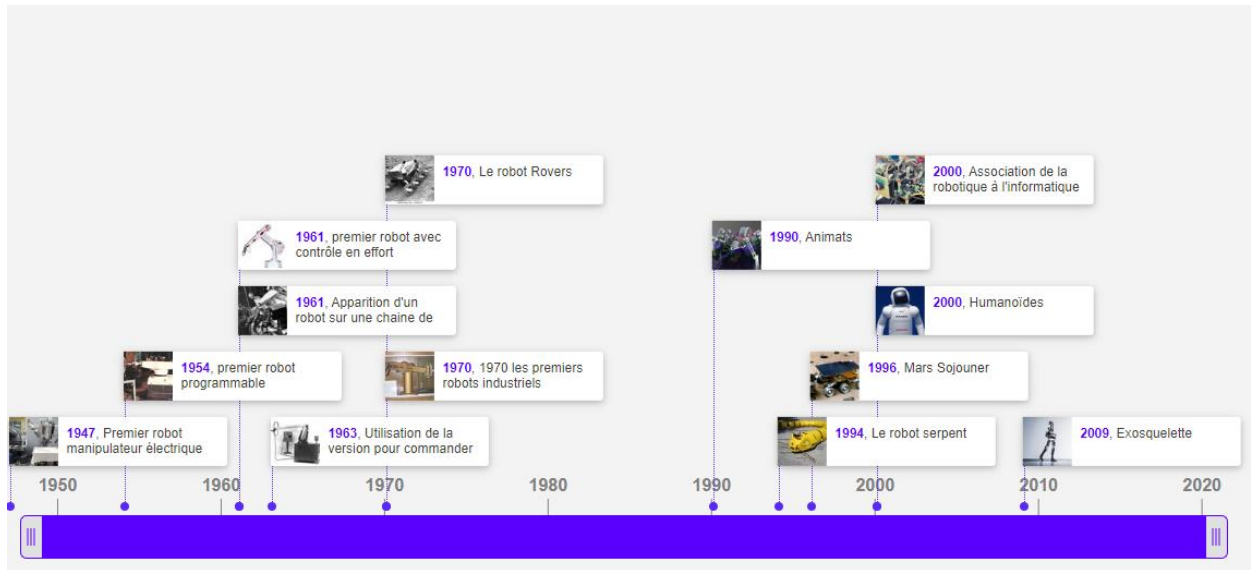


Figure 1-3 Boucle de décision



Figure 1-2 main robotique a résolu un Rubik's Cube

1.4 La première machine programmable

De 1701 jusqu'à la fin, la mode en Europe est influencée par les tendances chinoises, avec des tenues de soie ornées de motifs complexes. Pour les tisserands de Lyon, la fabrication de ces étoffes représente un défi considérable.

Basile Bouchon, en 1725, conçoit une solution innovante : une machine à tisser utilisant une bande de papier perforé pour contrôler le passage des aiguilles à travers le tissu. Ce mécanisme permet de reproduire des motifs complexes de manière plus efficace. En 1745, Jacques de Vaucanson perfectionne cette invention en l'automatisant davantage.

Plus tard, en 1801, Joseph-Marie Jacquard révolutionne cette technologie en séparant les cartes perforées, qui contiennent le modèle à réaliser, de la machine elle-même. Le métier à tisser Jacquard, produit en milliers d'exemplaires, devient ainsi la première machine à

automatiser le traitement de l'information tout en opérant une distinction claire entre la machine et le programme qu'elle utilise

Cette innovation ouvre la voie aux ordinateurs et robots modernes, capables de fonctionner de manière autonome en suivant des instructions programmées. Le métier à tisser Jacquard est donc un précurseur essentiel dans l'histoire de l'informatique et de l'automatisation industrielle. [23]



Figure 1-4 Machine à tisser

1.5 Le premier bras télé-opéré

Ce n'est qu'en Lorraine, en 1954, à Argonne, qu'un chercheur, Raymond Goertz, a l'idée de simplifier les bras de télé-opération en utilisant des moteurs électriques pour les articulations. Un tel dispositif permet à l'opérateur d'être à plusieurs centaines de mètres du lieu où il manipule des éléments **dangereux** : les instructions qu'il indique à la pince sont transmises par des fils électriques. [23]

Les grands constructeurs des robots industriels

Adept Technology Inc : une entreprise américaine basée en Californie, spécialisée dans l'industrie. Créés en 1983, leurs robots sont employés dans les domaines de la fabrication de haute précision et de la vitesse, de l'emballage et de l'automatisation.

Kuka : une entreprise basée à Augsburg, en Allemagne, a lancé son premier robot industriel en 1973, qui est principalement utilisé dans le secteur de l'automobile et de la métallurgie.

ABB : La société multinationale, basée à Zurich en Suisse, a acquis une expertise considérable dans le domaine de l'automatisation et de l'industrie. Elle compte 4600 employés dans 53 pays et a installé plus de 250 000 robots à travers le monde

Stäubli : est une entreprise suisse qui a débuté dans le domaine de la robotique en 1982, avec le SCARA, un robot à 4 même à 6 axes (ddl) qui est utilisé dans divers domaines tels

que les matières plastiques, l'électronique et l'énergie solaire. [24]

1.6 Robot industriel :

Un robot industriel est un ensemble « autonome » composé de trois éléments :

- Le bras :

Il s'agit d'un élément mécanique qui assure le mouvement du robot, et il est classé en 4 grandes familles :

- Robots SCARA 3 ou 4 axes 1
- Robots polyarticulés (4 à 7 axes) 2
- hexapode 3
- double bras.

- L'armoire de commande :

Le mouvement d'un robot est géré par un élément électronique, qu'il s'agisse d'un automate ou d'un robot.

Le programme de travail du robot pour commande est inclus dans une commande numérique. Son voyage ainsi que ses liens avec les actionnaires.

- Le pupitre de programmation :

Il est également connu sous le nom de boîtier d'opérateur, c'est le panneau de commande déporté du robot qui permet de programmer par apprentissage. En règle générale, il est équipé d'un écran d'affichage, de boutons de commande et d'un système de mouvement du robot. [25]



Figure 1-5 Les différents types des robots

1.7 Constituants d'un Robot manipulateur :

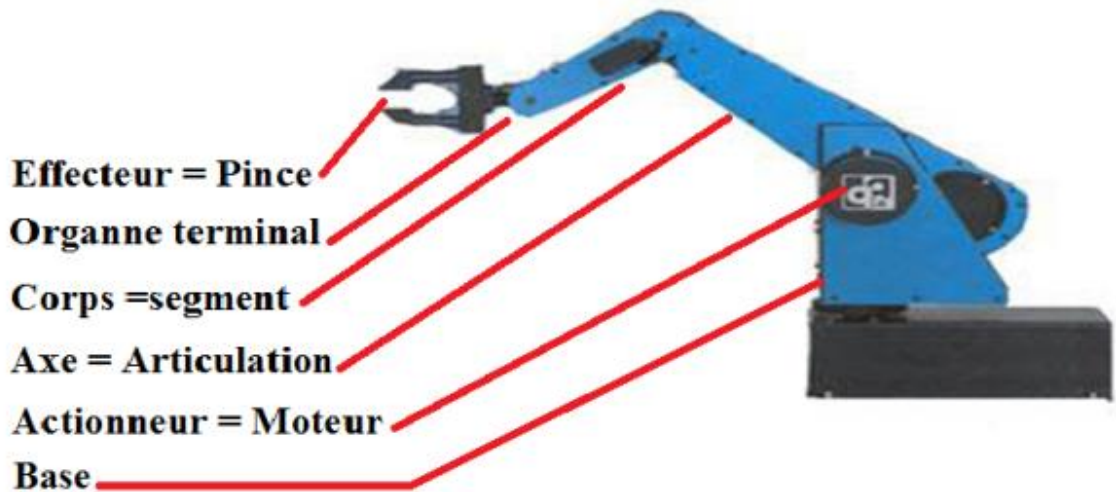


Figure 1-6 : Vocabulaires du bras de robot.

1.8 Les robots manipulateurs

Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour s'effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage. [26]

1.9 Type de robot manipulateur :

Les robots manipulateurs, indispensables dans de multiples domaines industriels, sont divisés en diverses catégories en fonction de leur structure, de leur mécanisme de déplacement et de leurs utilisations particulières. Voici un aperçu des différentes catégories de robots manipulateurs :

- Robots articulés :

Les robots articulés se meuvent à la manière d'un bras humain. Montées sur un joint de torsion, les applications de ce type de robotique sont nombreuses. On les utilise généralement pour l'assemblage, le soudage, la manutention, l'entretien et l'emballage des machines



Figure 1-7 Architecture de robot fanuc

- Robot humanoïde :

Malgré le fait que de nombreux robots humanoïdes mobiles puissent appartenir techniquement à l'AMR, ce terme est employé pour désigner les robots qui remplissent des fonctions axées sur l'homme et qui ont souvent des formes similaires à celles de l'homme d'homme. Ils exploitent de nombreux éléments technologiques similaires à ceux des AMR afin de repérer, planifier et agir dans le cadre de tâches telles que l'indication d'un trajet ou la fourniture de services courants.



Figure 1-8 le robot qui travaillera dans les usines de BMW aux côtés des humains

- Drone :

Les drones ou UAV (Unmanned Aerial Vehicles) sont des véhicules aériens sans pilote capables d'effectuer des tâches autonomes ou semi-autonomes. Leur utilisation principale est les missions militaires de reconnaissance ou de surveillance sans risque de pertes.



Figure 1-9 L'utilisation de drones pour la maintenance industrielle

- Robots mobiles autonomes (AMR) :

Les AMR voyagent à travers le monde et prennent des décisions presque instantanées pendant leurs déplacements. Les capteurs et les caméras sont des technologies qui les permettent d'apprendre sur leur environnement. Les dispositifs de traitement embarqués leur permettent d'analyser ces informations et de prendre une décision éclairée, que ce soit pour se déplacer afin d'éviter un travailleur qui arrive, pour choisir le colis approprié ou pour choisir une surface appropriée à désinfecter. Il s'agit de solutions mobiles qui requièrent une intervention humaine restreinte pour effectuer leur tâche.



Figure 1-10 Système de robot de tri de colis largement utilisé utilisant AMR AGV avec rendu basculant 3d

- Hybride

Les diverses catégories de robots sont fréquemment associées afin de concevoir des solutions hybrides qui peuvent accomplir des tâches plus complexes. Par exemple, il est possible d'associer un AMR à un bras robotique afin de concevoir un robot conçu pour la gestion des colis dans un entrepôt. Au fur et à mesure de la fusion des fonctionnalités dans des solutions uniques, les capacités de calcul sont également Renforcées.



Figure 1-11 le robot acrobate hyper agile de Boston Dynamics

- Véhicules à guidage automatique (AGV)

Si les AMR circulent librement dans les environnements, les AGV utilisent des pistes ou des chemins

Préétablis et sont souvent sous la surveillance d'un opérateur. En général, ils sont employés afin de livrer des matériaux et de déplacer des articles dans des environnements surveillés tels que des entrepôts et des locaux industriels.



Figure 1-12 Véhicule à Guidage Automatique (AGV) – Les avantages

1.10 Classification géométrique

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur.

- Robots cylindriques

Les deux axes de mouvement du robot cylindrique sont le haut et le bas. On effectue la rotation en jonction avec la base. Le bras horizontal a également la capacité de se déplacer à l'intérieur et à l'extérieur, ce qui crée un troisième axe de mouvement restreint. [1]

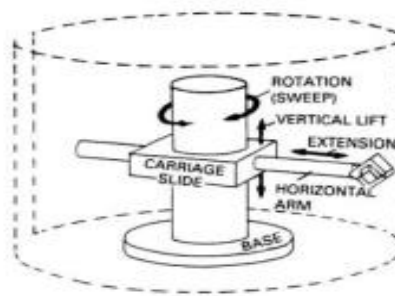


Figure 1-13 L'espace de travail de robot cylindrique

- Robots rectilignes

Il existe trois axes de mouvement pour les robots rectilignes (x, y, z). C'est pourquoi le robot rectiligne est parfois désigné sous le nom de robot cartésien. Ces robots sont utilisés grâce à une pompe pneumatique. [1]

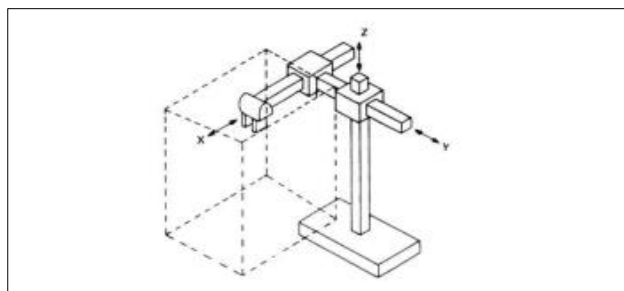


Figure 1-14 L'espace de travail de robot rectiligne

- Robots SCARA

La norme ISO 8373: 1994, No.3.15.6, définit un robot SCARA comme étant constitué de deux liaisons pivots parallèles afin de fournir conformément à un plan spécifique. et peut être perçue comme une situation spécifique d'un robot cylindrique. Le mot «SCARA» désigne «Arme de conformité Sélectionnée». [2]

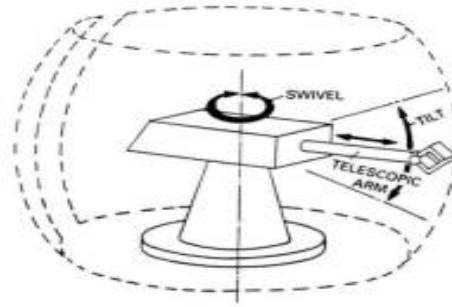


Figure 1-15 L'espace de travail de robot sphérique

1.11 Caractéristiques des robots

- Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée "P" Réduisant le mouvement entre deux corps à une translation le long d'un axe commun. [1]



Figure 1-16 Représentation d'une articulation prismatique.

- Articulation rotoïde :

Il s'agit d'une articulation de type pivot, notée "R" Réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe. [1]



Figure 1-17 Représentation d'une articulation rotoïde

1.12 Caractéristique géométrique

Nombre d'axes (mus par un actionneur).

Architecture (série ou parallèle).

Chaînage des articulations.

Nombre de degrés de liberté.

P : Prismatique (Translation)

R : Rotation

DL : Degrés de liberté

Volume de travail

Le volume de travail accessible par l'outil du robot est le volume qui peut balayer l'outil. Volume dépend de la géométrie du robot, de la longueur des segments et du débattement des articulations

Précision / Répétabilité

Le positionnement absolu est considéré comme imprécis lorsque sa taille dépasse 1 mm, ce qui entraîne l'une des erreurs suivantes :

- Modèle géométrique avec des erreurs.
- Des erreurs dans la mesure de la position.
- Flexibilité. La répétabilité d'un robot correspond à la marge maximale d'erreur de positionnement répétée de l'outil dans n'importe quel point de son espace de travail. La répétabilité ne doit généralement pas dépasser 0,1 mm [3].

1.13 Performances en mouvement

La vitesse maximale correspond à la vitesse à laquelle chaque axe du robot se déplace ou se tourne. Fréquemment, les fabricants indiquent la vitesse maximale de l'outil ou de l'organe terminal [18].

La vitesse maximale dépend de l'inertie, c'est-à-dire de la position du robot. Cette accélération est donnée pour chaque axe dans la configuration la plus défavorable (maximale Inertie, maximale charge) [2]

Charge utile

C'est la charge qui garantit une durée de vie la plus longue possible. Donc, c'est la charge maximale que peut porter le robot sans perturber ni la répétabilité ni les performances dynamiques du robot. Cette charge utile est nettement inférieure à la charge maximale que

peut porter le robot et est directement dépendante des actionneurs [2]

Principales industries utilisant des robots manipulateurs avec pince

La robotique joue un rôle essentiel dans tous les processus productifs à l'échelle industrielle. On retrouve actuellement la robotisation des procédés dans presque tous les domaines industriels, de l'industrie agroalimentaire à l'industrie automobile, pétrochimique ou métallurgique, ainsi que dans les systèmes de gestion des dépôts de centres logistiques, ainsi que dans les procédés de production de médicaments ou de prothèses pour le secteur de la santé. Industrie Manufacturière : Utilisation des robots pour des tâches de fabrication, d'assemblage, de soudure, de peinture, et de manipulation de matériaux.

Logistique et Entrepôt : Les robots autonomes sont utilisés pour le tri, le transport, et la gestion des stocks dans les entrepôts, augmentant l'efficacité et réduisant les erreurs humaines

Secteur Automobile : Robots pour l'assemblage de pièces automobiles, la peinture, et l'inspection des véhicules. Ils permettent une production de masse avec une haute qualité constante.[2]

Santé : Robots pour la chirurgie assistée, la réhabilitation, et la gestion des médicaments dans les **hôpitaux**.



Figure 1-19 : Robots dans le secteur



Figure 1-18 : Robot dans le secteur de la santé

1.14 Bénéfices de leur utilisation

Précision : Les robots peuvent placer des composants minuscules avec une haute précision.

Vitesse : Augmentation significative de la vitesse d'assemblage par rapport au travail manuel.

Réduction des Erreurs : Diminution des défauts de fabrication grâce à une précision constante

Sécurité : Élimine l'exposition des travailleurs aux fumées toxiques et aux hautes températures.

Qualité : Garantit des soudures uniformes et robustes.

Productivité : Fonctionne en continu sans fatigue, augmentant ainsi la production

Flexibilité : Capables de manipuler une grande variété de produits.

Réduction des Coûts : Réduit le besoin de main-d'œuvre pour des tâches répétitives et fatigantes.

Défis et Perspectives

Bien que la robotique offre de nombreux avantages, elle présente également des défis :

Coûts Initiaux : Le coût d'acquisition et d'installation des robots peut être élevé.

Complexité de l'Intégration : Intégrer les robots dans les systèmes existants peut nécessiter des compétences spécialisées.

Évolution Technologique : La rapidité des avancées technologiques nécessite une adaptation continue et des mises à jour régulières.

1.15 Conclusion

Dans ce chapitre, il est possible de constater que la robotique mobile occupe une place distincte dans le domaine de la robotique. À la différence des robots manipulateurs industriels qui opèrent de manière autonome dans de nombreuses usines automatisées, les robots mobiles sont très rares. Le manque d'applications possibles n'explique pas cette situation, mais une fois que la mobilité est disponible, on peut envisager des robots facteurs, nettoyeurs, gardiens, démineurs, explorateurs, jardiniers et bien d'autres encore. La faible diffusion s'explique principalement par le fait que ces tâches sont beaucoup plus complexes que celles réalisées par des robots manipulateurs industriels. La sphère dans laquelle un robot mobile doit se déplacer est souvent extrêmement étendue, partiellement ou complètement inconnue, difficilement définie géométriquement et possédant une dynamique spécifique.

**Chapitre II : Conception et Architecture des
Robots Manipulateurs avec Pince**

2.1 Introduction

Dans ce chapitre, nous exposerons nos différentes options en ce qui concerne la configuration du bras Manipulateurs avec Pince. Nous exposerons comment le bras est contrôlé, en utilisant une carte Arduino programmable, ainsi que les éléments du capteur et de l'actuateur. Finalement, nous exposerons les spécificités de la production des éléments du bras en utilisant la technologie d'impression 3D, dans le but de proposer une méthode précise pour la conception du robot étudié dans nos projets.

2.2 Choix du bras manipulateur étudié :

Aujourd'hui, on observe une généralisation de la robotique dans les pays développés, comme nous l'avons mentionné dans le chapitre précédent. Grâce à cette généralisation, l'achat d'un robot est devenu moins coûteux que par le passé, ce qui nous permet d'éviter de commencer de zéro comme dans notre situation. Des bras manipulateurs sous forme de squelette (en pièces détachées) et une carte de programmation vide sont disponibles, ce qui nous permet de programmer à notre guise pour accomplir les tâches qui nous intéressent. (voir figure 3-1)

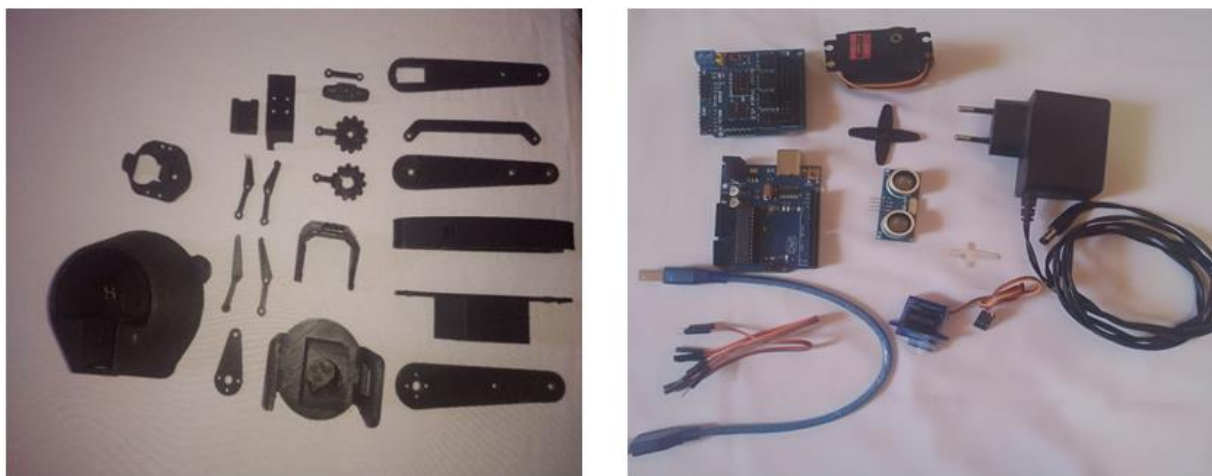


Figure 2-1 Pièces constituant le bras manipulateur

2.3 La carte Arduino :

Le système Arduino est une carte électronique qui repose sur un microcontrôleur et utilise un minimum de composants pour effectuer des fonctions plus ou moins avancées à un prix abordable. Son interface USB permet de la programmer. Il s'agit d'une plateforme open-source qui repose sur une carte simple.

Le programme peut être écrit, compilé et transféré vers la carte à microcontrôleur grâce

à un microcontrôleur (de la famille AVR) et un logiciel, véritable environnement de développement intégré.

Arduino offre la possibilité de concevoir des applications matérielles industrielles légères ou des objets interactifs (comme des créations artistiques), et peut accueillir une grande diversité de capteurs en entrée. Arduino a également la capacité de gérer divers actionneurs tels que les lumières, les moteurs ou tout autre type de sorties matérielles. Les projets Arduino ont la capacité d'être autonomes. On communique avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées préassemblées [27]



Figure 2-2 Cartes Arduino

2.3.1 Bref historique de l'Arduino

Arduino est une plateforme open source créée en 2005 à l'Institut de Design Interaction d'Ivrea en Italie. Le projet a été initié par Massimo Banzi et David Cuartielles. Pour sa thèse de fin d'études, Hernando Barragan avait commencé à développer une carte électronique nommée Wiring, accompagnée d'un environnement de programmation libre et ouvert. [28]

Hernando Barragan utilisait les sources du projet Processing pour ce travail. En mettant en place un langage de programmation simple et adapté aux projets de designers, la carte Wiring a inspiré la création du projet Arduino en 2005. À l'instar de Wiring, l'objectif était de concevoir un dispositif facile à utiliser, peu coûteux, avec des codes et des plans en open source.

2.3.2 Matériel arduino

La carte Arduino repose sur un microcontrôleur intégré, accompagné de diverses entrées et sorties permettant de connecter différents types de composants externes.

Du côté des entrées, on peut brancher des capteurs qui collectent des informations sur leur environnement. Par exemple, une sonde thermique peut mesurer la variation de température, un détecteur de présence ou un accéléromètre peut détecter le mouvement, et un bouton-poussoir peut enregistrer un contact, etc...



Figure 2-3 Différent types des capteurs pour l'arduino

Côtés sortis, des actionneurs qui agissent sur le monde physique telle une petite lampe, un moteur, hautparleur...



Figure 2-4 Différent s actionneurs pour l'arduino

2.3.3 L'utilisation de la carte Arduino UNO :

L'Arduino Uno est une carte de microcontrôleur développée par l'équipe Arduino, qui utilise le microcontrôleur ATmega328P. En raison de sa facilité d'utilisation, de son coût abordable et de sa flexibilité, elle est l'une des cartes les plus appréciées et les plus utilisées dans la communauté des passionnés d'électronique, des makers, des ingénieurs et des étudiants.

Alimentation	-Via port USB -7 à 12 V sur connecteur alim 5.5 x 2.1 mm
Microprocesseur	-ATMega 328 -Mémoire flash : 32 KB -Mémoire SRAM : 2 KB -mémoire EEPROM : 1 KB
Interfaces	-14 broche d'E/S dont 6 PWM -6 entrées analogiques 10 bits -Bus série. 12C et SPI
Intensité par E/S	-40 MA
Cadencement	-16 MHz -Gestion des interruptions fiche USB B
Version	-Rev . 3
Dimensions	-74 x 53 x 15 mm
Reference	-Arduino uno Rev3 A000066 module prêt à l'emploi -Livré avec support en plastique Version d'origine conçue et assemblée en Italie

Table 1 Caractéristiques de l'Arduino Uno

Chapitre II : Conception et Architecture des Robots

Manipulateurs avec Pince

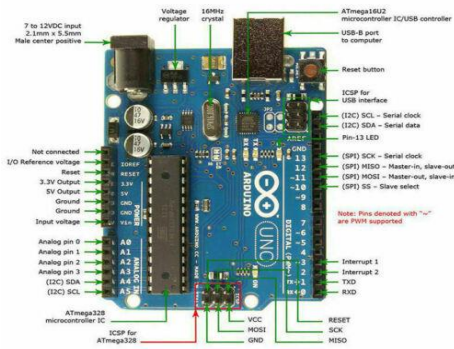


Figure 2-6 Carte Arduino UNO

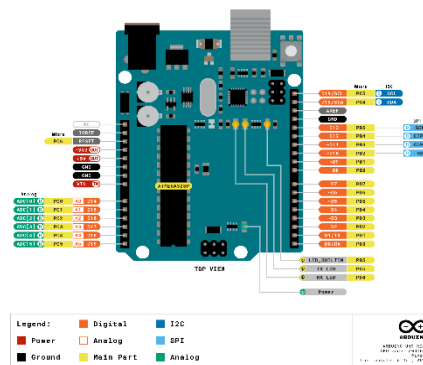


Figure 2-5 Schéma de la carte Arduino

2.3.4 Applications de l'Arduino Uno

L'Arduino Uno est utilisé dans une vaste gamme de projets, allant des simples expériences électroniques aux systèmes complexes d'automatisation et de contrôle. Quelques exemples d'applications incluent :

Prototypes de capteurs et actionneurs : Création rapide de prototypes pour tester divers capteurs (température, humidité, lumière, etc.) et actionneurs (moteurs, relais, etc.).

Systèmes embarqués : Développement de systèmes embarqués pour des projets de domotique, de robotique, et d'Internet des Objets (IoT).

Éducation : Apprentissage de la programmation et de l'électronique grâce à des projets pratiques et interactifs.

Art interactif : Conception de sculptures interactives, installations artistiques et performances utilisant des capteurs et des actionneurs. [28]

Logiciel Arduino : Le logiciel arduino est gratuit (open source) et se télécharger sur le site officiel d'Arduino, à l'adresse

Différentes options de téléchargement sont disponibles, vous devez choisir selon le système d'exploitation de votre ordinateur : Windows, MacOS X, Linux. En général, il est préférable de choisir la dernière version officielle, même si pour les cartes les plus récentes, il est parfois nécessaire de privilégier les versions « beta » du logiciel, c'est-à-dire des versions non testées et à perfectionner, mais tout de même fonctionnelles. [5]

L'IDE (Environnement de Développement Intégré) est un programme spécial que vous pouvez exécuter sur votre ordinateur. Il vous permet d'écrire des croquis pour la carte Arduino dans un langage simple inspiré du langage Processing. La magie opère lorsque vous appuyez sur le bouton pour télécharger le croquis sur la carte : le code que vous avez écrit est traduit en langage C (qui est généralement assez complexe pour un débutant), puis transmis au

compilateur avr-gcc. Ce dernier est un élément crucial du logiciel open source qui effectue la traduction finale dans le langage compréhensible par le microcontrôleur. Cette étape est essentielle car elle illustre comment Arduino simplifie votre vie en masquant autant que possible la complexité de la programmation des microcontrôleurs. [6]

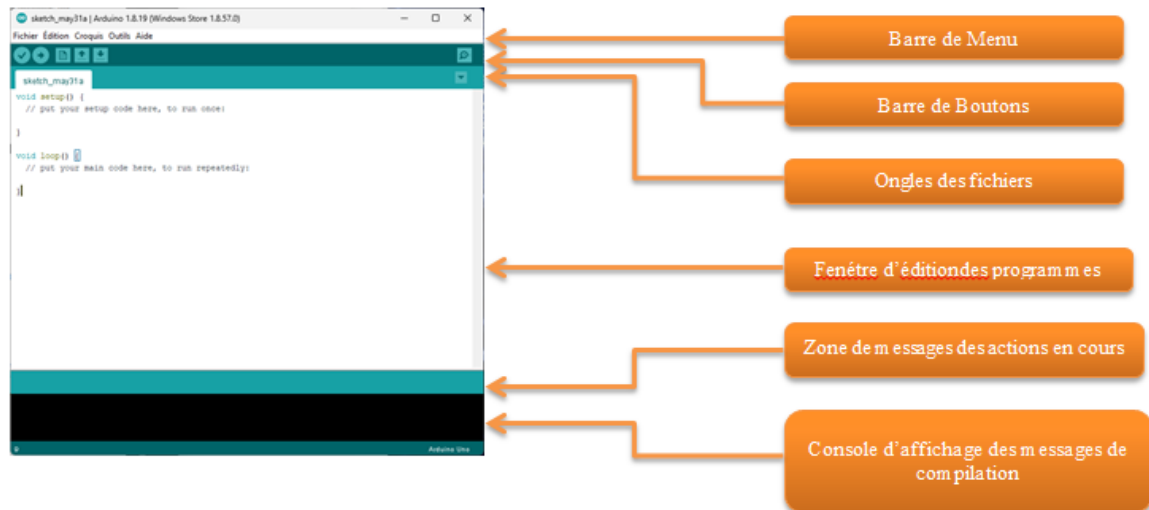


Figure 2-7 Interface de logiciel Arduino IDE 2.3.2

Cette fenêtre vide sera bientôt remplie de mots, de chiffres et d'autres textes. Elle ressemble à tout autre logiciel que vous avez utilisé, avec ses menus, boutons, alertes spéciales et divers contrôles : Ouvrir, Enregistrer, et le bouton du moniteur série à l'extrême droite.

Nouveau : crée une nouvelle esquisse.

Ouvrir : affiche un menu de tous les croquis présents dans votre dossier de croquis. Un clic sur l'un d'eux l'ouvre dans la fenêtre actuelle.

Enregistrer : sauvegarde votre croquis.

Vérifier : ce bouton vous permet de vérifier que votre code est exempt d'erreurs de syntaxe.

Envoyer : ce bouton vérifie et télécharge un croquis sur l'Arduino s'il n'y a aucune erreur de syntaxe ou de mise en forme. moniteur série : ce bouton ouvre le moniteur série et affiche les informations provenant du port série de l'Arduino. Le moniteur série est un outil qui vous permet de "communiquer" avec l'Arduino et de voir les informations importantes. , [7]

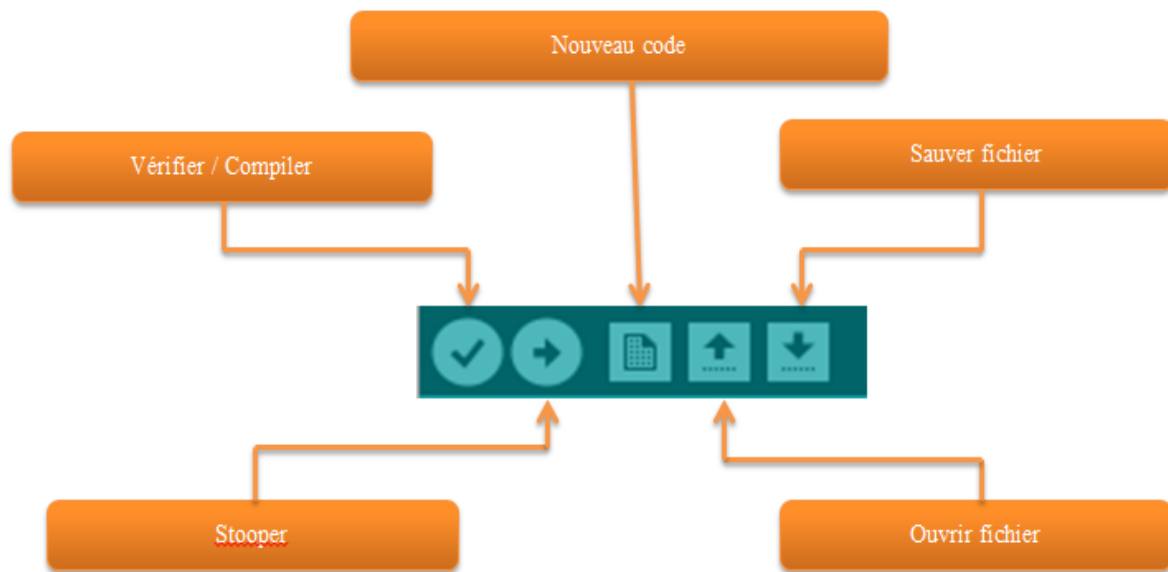


Figure 2-9 : Fonctionnalités principales de l'interface Arduino IDE

Les avantages de l'Arduino sont nombreux :

Prix abordable : Elle est accessible financièrement.

Environnement de programmation simple et clair : L'IDE est facile à utiliser.

Multiplateforme : Fonctionne sous Windows, Mac et Linux.

Large choix de bibliothèques : De nombreuses librairies sont disponibles, offrant diverses fonctions déjà implémentées.

Open source : Le logiciel et le matériel sont open source et extensibles.

Support en ligne : De nombreux conseils, tutoriels et exemples sont disponibles en ligne (forums, sites personnels, etc.).

Shields : Il existe des cartes supplémentaires qui se connectent au module Arduino pour augmenter ses capacités, comme des afficheurs graphiques couleur, des interfaces Ethernet, des modules GPS, etc.

2.4 Shield Arduino :

Les shields Arduino sont des modules qui se connectent sans souder aux cartes Arduino ou à d'autres shields pour étendre leurs fonctionnalités (Wi-Fi, 4G, écran, Bluetooth, etc.). Fidèles à l'esprit d'origine d'Arduino, ces shields sont faciles à produire et à utiliser (Figure 3-10)[8]



Figure 2-10 les Shields Arduino

2.5 Généralités sur les servomoteurs

2.5.1 Présentation

Le servomoteur (souvent abrégé en « servo », du latin servus, « esclave ») est un moteur qui peut faire face à un effort statique et dont la position est réglée en permanence et ajustée en fonction de la mesure. Il s'agit donc d'un système soumis.

Un servomoteur est un dispositif électrique qui peut atteindre des positions préétablies et les maintenir.

De la même manière que les moteurs à courant continu, les servomoteurs ont un axe de rotation qui est bloqué par un système de bridage.

Le servomoteur est un ensemble composé de deux composants mécaniques et électroniques :

Un moteur à courant continu de très petite taille

Un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple

Un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie

Un dispositif électronique d'asservissement

Un axe dépassant hors du boîtier avec différents bras ou roues de fixation

Un axe qui dépasse le boîtier, muni de divers bras ou roues de fixation. Un servomoteur se distingue principalement par :

Sa capacité à tourner pour atteindre la position souhaitée en secondes/degrés.

Son couple, également connu sous le nom de "torque en anglais", correspond à la force que le servomoteur peut fournir pour maintenir la position désirée en kg/cm.

L'unité kg/cm représente le poids que le servomoteur peut supporter en prenant en compte la présence d'une tige de 1 cm. [9]

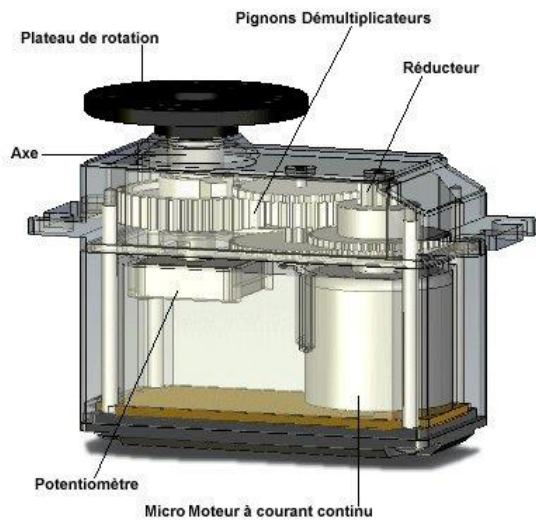


Figure 2-12 schéma d'un servomoteur



Figure 2-11 servomoteur SG90

Le servomoteur possède trois fils de connexion pour pouvoir fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

rouge : pour l'alimentation positive (4.5V à 6V en général) noir ou marron : pour la masse (0V) orange, jaune, blanc, ... : entrée du signal de commande

L'alimentation fournit la puissance absorbée par un servomoteur, tandis que le signal de commande ne consomme presque aucun courant. Un microcontrôleur peut donc être directement commandé par un servomoteur.

L'utilisation d'un signal PWM à fréquence fixe de 50Hz ($T=20\text{ms}$) est utilisée pour gérer un servomoteur. La durée de l'état haut de ce signal détermine la position de l'axe du servomoteur.

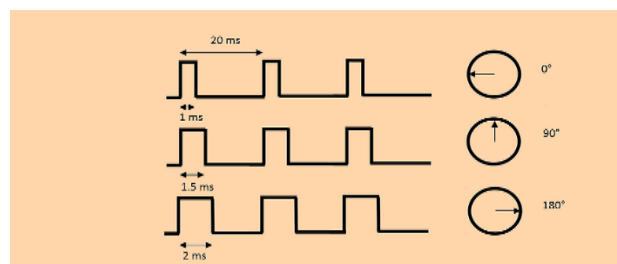


Figure 2-13 Exemple de signaux en position pour un servomoteur

Le servomoteur est contrôlé par un fil de commande et alimenté par deux autres fils.

- * Le fil rouge est connecté à l'alimentation positive (5V ou 6V selon le type de servomoteur).
- * Le fil noir (ou marron) est connecté à la masse (GND).
- * Le fil jaune ou orange est utilisé pour la commande.

Ces fils sont illustrés dans les deux figures ci-dessous :

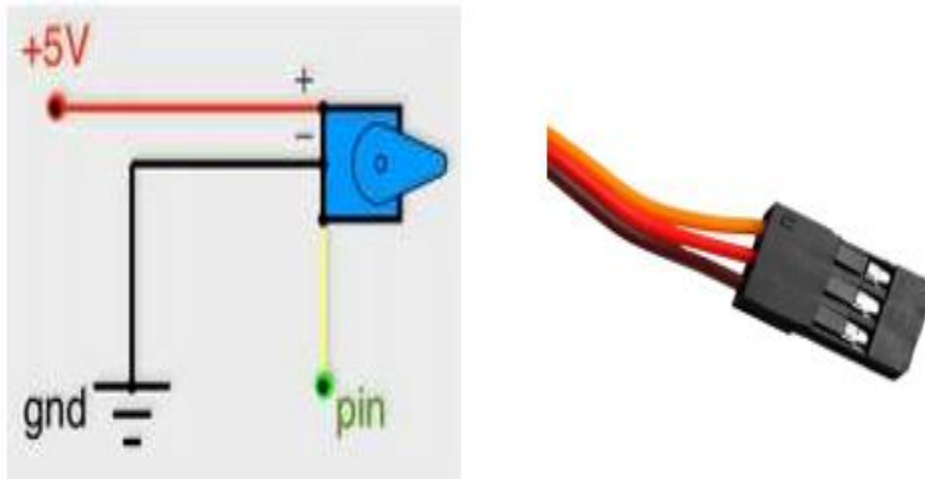


Figure 2-15 Branchement électrique de servomoteur

2.5.2 Les types de servomoteur :

Il y a 2 types 2 servomoteurs

- Servomoteur du type MG996R
- Servomoteur du Type SG90

Les caractéristiques	MG996R	Sg90
		
Poids	65g	9 g
Vitesse	4.8V: 0.17 sec/60°	0,1s/60°
Dimensions	40×19×43mm	32 x 12 x 32 mm
Alimentation	4,8 à 7.2 v	4,8 à 6 v
Angle de rotation	180°	180°
Couple d'arrêt	13kgf.cm	1,8 kgf.cm (4,8V), 2,2 kgf.cm (6V)

Table 2 Les paramètres de chaque servomoteur

2.6 Le RADAR

2.6.1 Introduction radar

Le radar est une méthode de détection d'objets utilisant des ondes radio pour déterminer la taille, la hauteur, la direction ou la vitesse des objets. Les systèmes radar sont disponibles dans différentes tailles avec différentes exigences de performance. Certains systèmes radar sont utilisés dans le contrôle du trafic aérien des aéroports, d'autres dans les systèmes de surveillance à longue distance et des États-Unis a inventé le terme RADAR lui-même comme un acronyme de la détection radio et d'une série d'autres développements en 1940. Les nouvelles utilisations du radar comprennent un large éventail de systèmes de contrôle du trafic aérien, y compris les radars, les systèmes d'astronomie, de défense aérienne, des systèmes antimissiles, des radars maritimes pour l'identification et le repositionnement des sites et d'autres navires, des dispositifs anti-collision d'aéronefs, de surveillance maritime, de contrôle de l'espace extérieur, et de système de rendez-vous. Le traitement du signal numérique est lié aux systèmes radar de haute technologie. [9]



Figure 2-16 Capteur de distance à ultrasons (4 broches)

2.6.2 Vue d'ensemble du système

Le diagramme de bloc de système de radar à courte portée utilisant Arduino est comme indiqué dans (Figure 1). Dans ce travail, la distance de l'objet est mesurée par un capteur de distance par ultrasons, et la sortie du capteur est connectée à l'unité de conditionnement du signal.

Après cela, il est traité par le microcontrôleur Arduino. Les résultats mesurés sont affichés sur l'ordinateur personnel.

Le capteur est attaché au moteur servo pour trouver la distance polaire autour du capteur jusqu'à 180 rotations.[10]

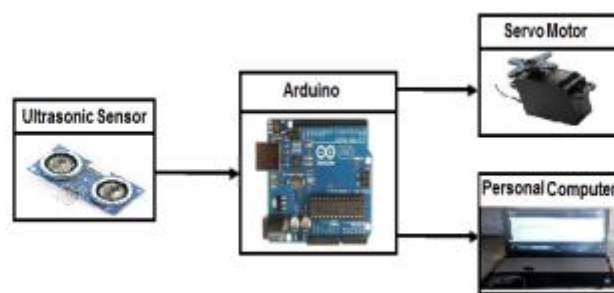


Figure 2-17 Schéma fonctionnel du système.

2.6.3 Capteurs ultrasoniques HC- SR04

Le capteur ultrasonique émet des ultrasons à 40 000 Hz, qui passent par l'air, tandis qu'il revient dans le module s'il y a un objet ou un obstacle sur son chemin. La distance peut être déterminée en fonction du temps de voyage et de la vitesse du son. [9]

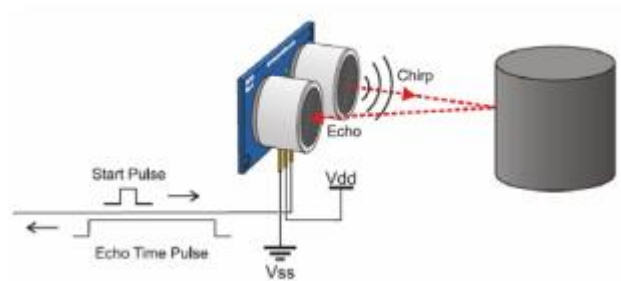


Figure 2-18 Capteur à ultrasons

La vitesse du son est d'environ 341 mètres par seconde (1100 pieds) dans l'air. Cette information et la différence de temps entre la transmission et la réception de l'impulsion sonore sont utilisées par le capteur ultrasonique pour calculer la distance d'un objet.

L'équation mathématique utilisée est la suivante :

$$\text{Distance} = \text{Temps} \times \text{Vitesse du son} / 2$$

Temps = le temps entre quand une vague ultrasonique est transmise et quand elle est reçue.

Vous divisez ce nombre par 2 parce que l'onde sonore doit se rendre à l'objet et retourner

2.6.4 Processing

La programmation avec Processing est un langage de programmation open-source et un environnement de développement intégré (IDE) conçu pour les arts électroniques, l'art des nouveaux médias et les communautés de design visuel. Son objectif est d'enseigner les bases de la programmation informatique dans un contexte visuel (Figure 3-19).[30]

Les spécifications de la programmation :

Téléchargement gratuit et open-source

Programmes interactifs avec sortie 2D, 3D ou PDF

Intégration OpenGL pour une accélération 2D et 3D

Disponible pour GNU/Linux, Mac OS X et Windows

Plus de 100 bibliothèques étendent le logiciel principal

Bien documenté, avec de nombreux livres disponibles

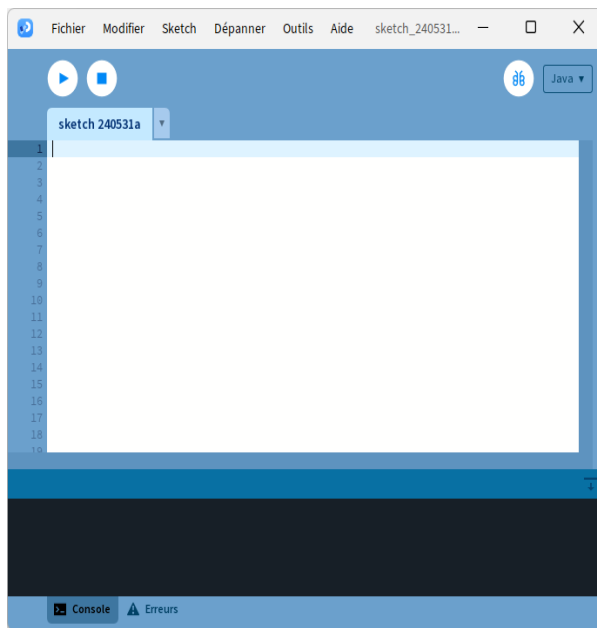
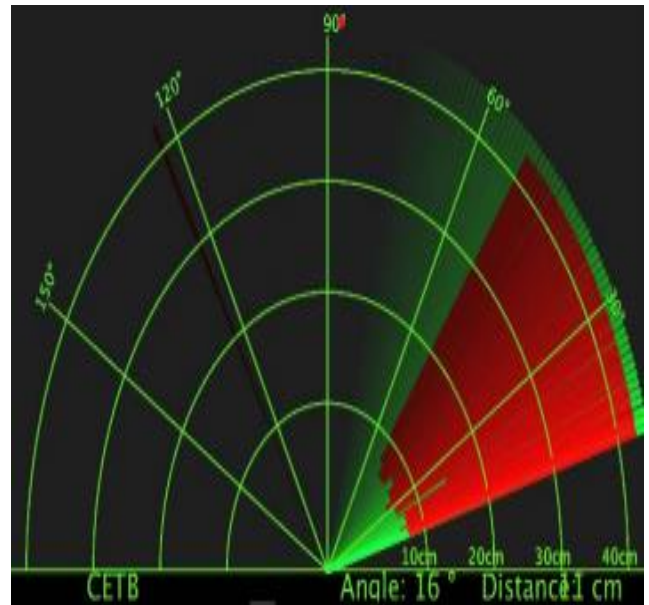


Figure 2-20 Software et processing

2.6.5 Applications :

Le système radar est principalement utilisé pour la cartographie et a plusieurs utilisations à des fins de protection. [10]

- Application dans la Force aérienne : Ceci est utilisé pour l'identification des articles qui entrent par des avions ou des appareils d'aéronefs qui ont un dispositif de radar. Il est souvent utilisé pour le calcul de la mesure de la hauteur.
- Application dans la marine : Il est également utilisé dans les navires ou dans les applications marines. La distance d'autres bateaux ou navires est mesurée sur de grands navires et peut être minimisée en ne pas entrer en collision avec l'aide de cet accident maritime. Il peut également être utilisé dans les ports pour voir la distance des autres navires et suivre ou surveiller les mouvements des navires.
- Application en météorologie : Le suivi ou la surveillance du vent est également effectué avec des systèmes radar. Il est devenu un important équipement de surveillance du climat. Pour commencer, les tempêtes sont utilisées pour détecter les tornades.

2.7 Source d'énergie

Pour l'alimentation du robot on peut utiliser une batterie de 12 volts, et de courant de 7Ampères.



Figure 2-21 Batterie de secours 12 V 7 Ah pour le stockage de l'énergie de secours

Cette section est facile. Il sera nécessaire de générer une tension stable. Il y a des composants spécialement conçus, les 78xx. Ces composants sont composés de trois pattes et leur fonctionnement est très simple. Ils ont entre les deux premières pattes une tension d'alimentation quelle que soit (supérieure à la tension de sortie souhaitée), ce qui leur permet de fournir une tension stabilisée égale à xx (05, 09, 12,...).

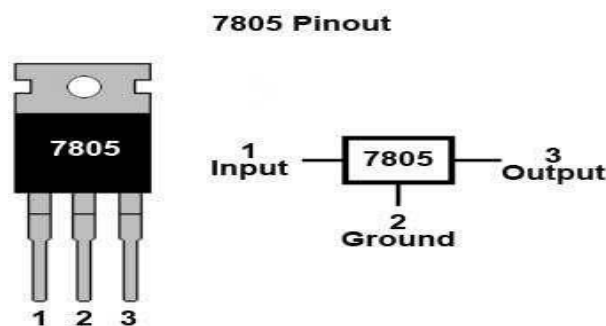


Figure 2-22 Régulateur de tension LM 7805

Les servomoteurs bénéficient d'un fil de commande et sont alimentés par deux fils supplémentaires. En général, ces trois fils sont regroupés dans une prise au format.

Le servo utilise un fil rouge pour connecter l'alimentation positive (+5 ou +6 V), le fil noir pour connecter la masse (GND) et le fil jaune pour la commande. Il existe de nombreuses informations sur le fonctionnement d'un servomoteur, ses éléments, son moteur et le petit potentiomètre qui permet de déterminer sa position, mais cette fiche se concentrera uniquement sur son utilisation avec l'Arduino.



Figure 2-23 Fils de raccordement

2.8 La diode

La diode 1N4007 est une diode standard utilisée pour transformer le courant alternatif en courant continu. Une tension répétitive inverse de 1 kV et un courant direct moyen de 1 A

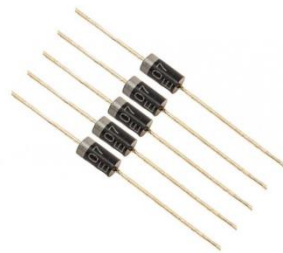


Figure 2-24 La diode 1N4007

sont présents. Sa chute de tension directe est faible (de 1,1 V à 1 A) et sa capacité en dépassement de courant est élevée (30 A). On l'emploie dans des domaines comme les alimentations, les régulateurs de tension ou les circuits de fixation. Le boîtier DO-41 (DO-204AL) de la diode 1N4007 est composé de deux broches et peut supporter des températures allant jusqu'à 150°C.

2.8.1 CARACTERISTIQUES

Diode de puissance
Type de boîtier: DO-41
Courant nominal : 1A
Tension max: 1 000

2.9 Définition du l'impression 3D

La fabrication additive, également connue sous le nom d'impression 3D, est une nouvelle technologie de prototypage rapide qui permet de fabriquer des pièces ou des maquettes avec précision, comme la réalisation d'une pièce de rechange dans le domaine spatial ou la fabrication d'organes dans le domaine biomédical... L'impression 3D consiste à créer un objet en trois dimensions à partir d'un modèle numérique. Des matériaux tels que le plastique, le métal, le béton sont utilisés [30].



Figure 2-25 imprimante 3D

2.10 Le prototypage rapide

Le prototypage rapide permet de fabriquer rapidement des modèles physiques à partir de données conçues par ordinateur en trois dimensions (CAO). Le prototypage rapide est utilisé dans un grand nombre de secteurs et permet aux entreprises de transformer de façon rapide et efficace des idées innovantes en produits finaux réussis

Le principe est de réaliser un prototype pour simuler des pièces séries, permettant de valider les caractéristiques d'une pièce tel que son fonctionnement et son design. Le terme « rapide » est important dans cette appellation. Cela signifie que les pièces sont obtenues sous des délais très courts, présentant l'avantage de valider au plus vite les différentes phases de développement d'un projet avant l'étape d'industrialisation. Le prototypage rapide intègre trois notions essentielles que sont le temps, le coût et la complexité des formes.

- Temps : l'objectif du prototypage rapide est de réaliser rapidement les modèles, dans

un but de réduction des temps de développement des produits.

- Coût : le prototypage rapide permet de réaliser des prototypes sans qu'il soit nécessaire de recourir à des outillages coûteux, tout en garantissant les performances du produit final. On est donc en mesure d'explorer différentes variantes du produit en cours d'élaboration.

- Complexité des formes : les machines procédant par ajout de matière sont capables de réaliser des formes extrêmement complexes (inclusion, cavité...), irréalisables par des procédés tel que l'usinage par exemple.

Logiciel et format de fichier

En prototypage rapide nous utilisons des logiciels dits CAO (Conception Assistée par Ordinateur). Ces derniers supportent des formats de fichier tels que stp, step, et igs et des formats de design comme stl. Ces formats sont développés par des éditeurs de logiciel par exemple, les logiciels dwg et dxf sont des formats industriels d'Auto CAD. Les informations modélisables sur les fichiers 3D dépendent des extensions. Ainsi, les données présentes sur les formats dwg sont relativement restreintes. Les logiciels les plus fréquemment utilisés en prototypage incluent : SketchUp, autoCAD, SolidEdge Catia et Solidworks , dans notre cas nous avons utilisé le SolidWorks. [32]

2.11 La modélisation du modèle

2.11.1 Définition SOLIDWORKS

Solidworks est une plateforme de conception assistée par ordinateur, développée par Dassault Systèmes. Le principe de conception paramétrique est utilisé et il produit trois types de fichiers qui sont interconnectés : la pièce, l'assemblage et la mise en plan. De cette façon, tout changement effectué sur l'un de ces trois fichiers aura un impact sur les deux autres. [35]

Chapitre II : Conception et Architecture des Robots

Manipulateurs avec Pince

L'interface:

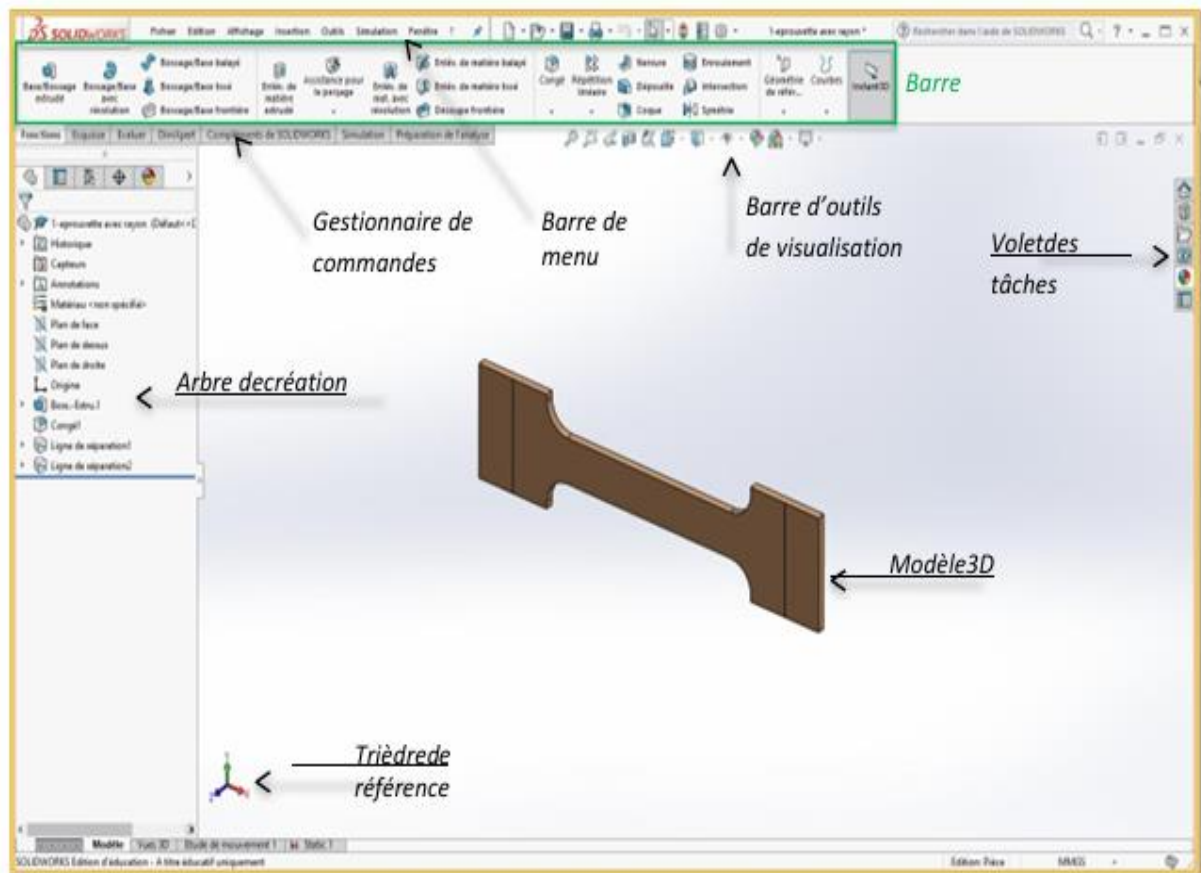


Figure 2-26 L'interface du logiciel

Les barres d'outils:

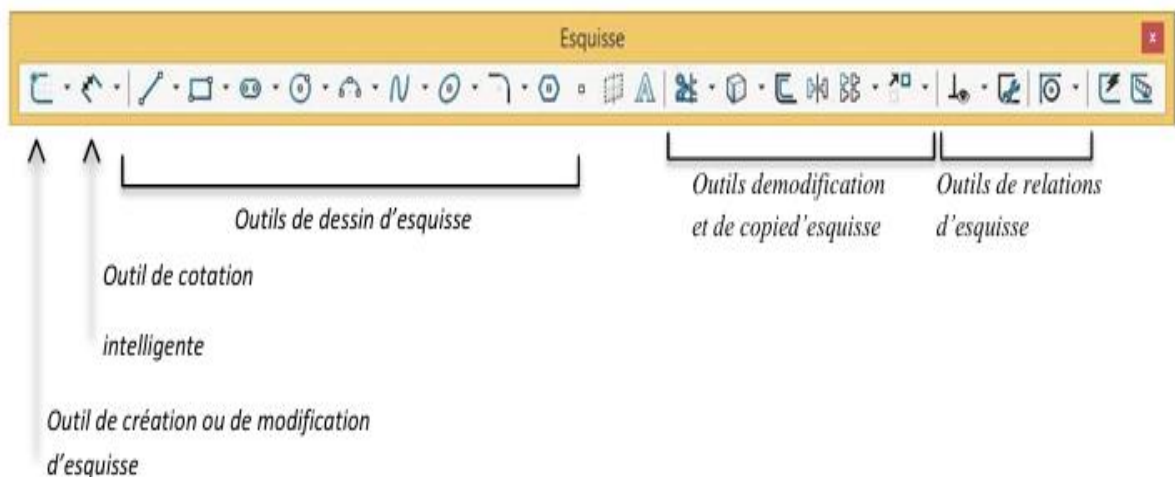


Figure 2-27 Les barres d'outils du logiciel Solidworks

2.11.2 Historique :

Jon Hirschtick, le fondateur, a En 1997, Dassault Systèmes est devenu l'actionnaire principal et a continué à développer les outils du logiciel [35].

2.11.3 Le fonctionnement :

SOLIDWORKS est un logiciel de modélisation 3D basé sur la conception paramétrique. Il crée trois types de fichiers correspondant à trois concepts fondamentaux : la pièce, l'assemblage et le plan. Ces fichiers sont interconnectés, et toute modification à un niveau quelconque est répercutée sur tous les fichiers concernés. Un dossier regroupant tous les fichiers liés à un même système constitue une maquette numérique. De nombreux logiciels viennent compléter l'éditeur SOLIDWORKS, comme des utilitaires spécialisés dans certains domaines (tôlerie, bois, BTP) ainsi que des applications de simulation mécanique ou de création d'images de synthèse qui utilisent les éléments de la maquette virtuelle.

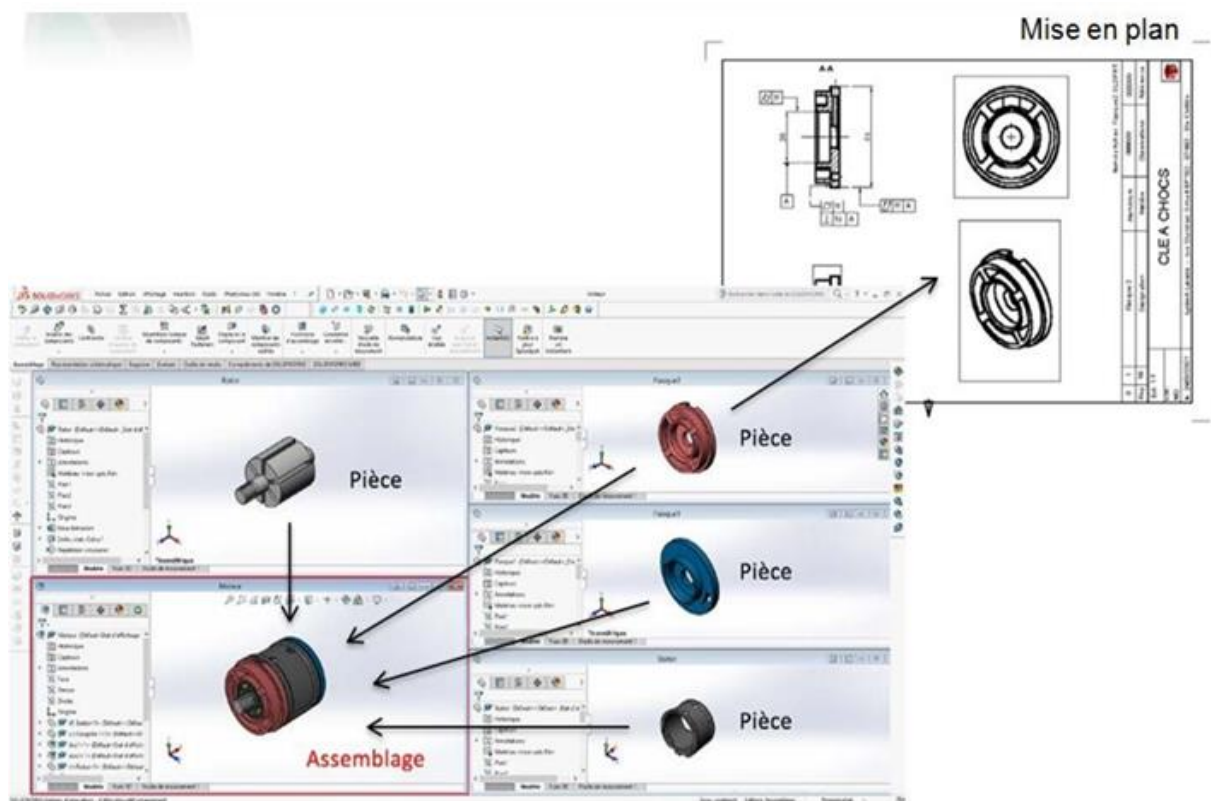


Figure 2-28 La pièce et l'assemblage par Solidworks

2.12 Conclusion

Ce chapitre a permis d'explorer en profondeur les aspects fondamentaux de la conception et de l'architecture des robots manipulateurs équipés d'une pince. À travers l'analyse des différentes configurations structurelles et des choix de conception, nous avons établi les principes clés qui gouvernent la construction d'un système capable de réaliser des tâches de préhension avec précision et efficacité.

Les choix effectués tout au long du processus de conception reflètent un équilibre entre la robustesse, la flexibilité, et l'adaptabilité du robot à diverses situations. Ce travail a démontré que la réussite d'un robot manipulateur dépend non seulement de la conception de ses composants individuels, mais aussi de la manière dont ces composants interagissent pour former un système cohérent et performant.

Chapitre III : Cinématique du Bras Manipulateur

3.1 Introduction

La conception et la commande des robots impliquent le calcul de modèles mathématiques qui décrivent le comportement de la structure robotique en fonction de ses paramètres articulaires. Ces modèles peuvent présenter des niveaux de difficulté très variés selon la structure du robot.

Ce chapitre présentera le modèle géométrique direct, basé sur le modèle de Khalil, qui modifie le modèle de Denavit-Hartenberg (DH), reconnu pour sa simplicité et son aptitude à représenter une vaste gamme de structures robotiques. En conclusion de ce chapitre, le modèle DHM (Denavit-Hartenberg Modifié) nous aidera à définir l'espace de travail du bras manipulateur analysé, afin de développer notre application dans le prochain chapitre.

3.2 Modèle géométrique d'un robot en chaîne simple :

3.2.1 Nécessité d'un modèle :

Les modèles mathématiques nécessaires sont :

- les modèles géométriques directs et inverses qui reflètent la position de l'organe dans sa structure.
- Variable articulaire selon le terminal et inversement.
- les modèles cinématiques direct et inverse qui expriment les vitesses de l'organe terminal.
- En fonction des variables articulaires et inversement.
- Les modèles dynamiques établissent les équations du mouvement du robot, ce qui permet de les rendre réalisables.
- D'établir les liens entre les couples ou les forces qu'exercent les actionneurs et les actionnaires.
- Positions, vitesses et accélérations des articulations. [15]

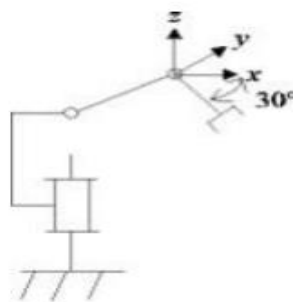


Figure 3-1 Modèle géométrique d'une chaîne simple

3.2.2 Mécanismes :

Un mécanisme, ou système mécanique articulé, est constitué d'un ensemble de solides reliés par des liaisons. On distingue deux types de mécanismes :

Les mécanismes à chaîne simple ouverte (ou en série), où l'on ne rencontre jamais deux fois la même liaison ou le même solide lors du parcours du mécanisme. Ce type de système est le plus répandu.

Les mécanismes à chaîne complexe, soit ceux qui ne sont pas en série (avec au moins un solide ayant plus de deux liaisons). Ces systèmes se classent en deux catégories : les chaînes structurées en arbre et les chaînes fermées, avantageuses pour leur rigidité accrue, leur précision et leur capacité à manipuler des charges lourdes.

Pour représenter un mécanisme, il existe deux méthodes :

Le schéma cinématique, qui emploie une représentation standardisée des liaisons pour décrire le mécanisme, en perspective ou en projection.

Le graphe non normalisé. À titre d'exemple, examinons quelques mécanismes. [30]

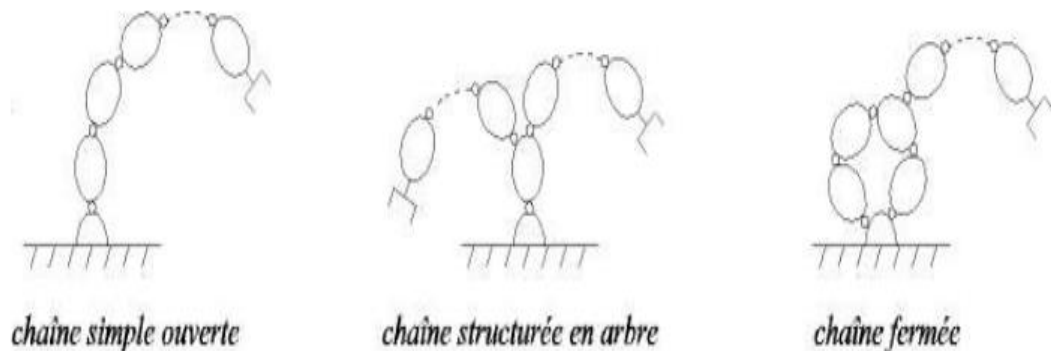


Figure 3-2 Type de chaînes cinématiques

3.2.3 Cinématiques série :

Une chaîne cinématique ouverte simple est composée d'une série de solides connectés par des liaisons simples. Autrement dit, si un élément (n) est connecté en série par $n-1$ éléments et que ces $n-1$ éléments sont eux-mêmes connectés par des liaisons simples, alors ces n éléments et n liaisons constituent une chaîne cinématique ouverte simple ou des liaisons en série (Voir figure 3-3).

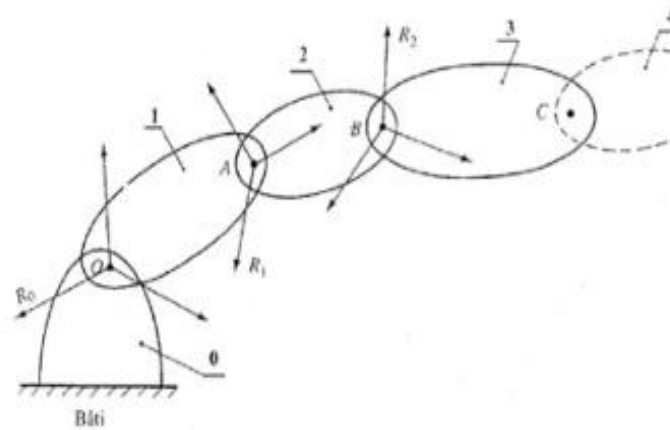


Figure 3-3 Chaîne cinématique simple ouverte

La plupart des robots industriels construits à ce jour sont de type série, c'est-à-dire que leur structure mobile est une chaîne ouverte formée d'une succession de segments reliés entre eux par des liaisons à un degré de liberté. Chaque articulation est commandée par un actionneur situé à l'endroit de l'articulation ou sur un des segments précédents. [34]

3.2.4 . Chaîne cinématique arborescente

Un robot à chaîne cinématique arborescente est constitué de plusieurs chaînes Cinématiques ouvertes reliées à un corps central. En fonctionnement, un robot à chaîne cinématique arborescente peut devenir un robot à Chaîne cinématique fermée par la saisie d'une pièce avec deux bras ou le contact avec le sol de deux ou plusieurs pattes (figure 3-4) [34].

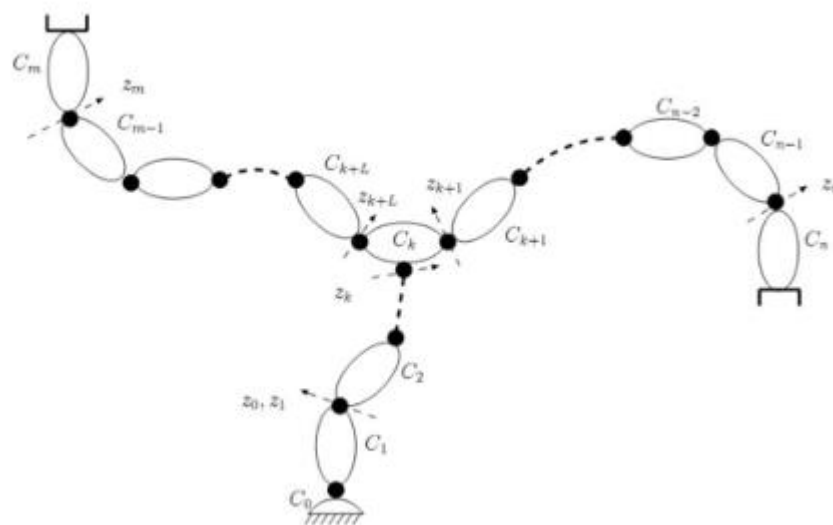


Figure 3-4 Chaîne cinématique simple ouverte

3.2.5 Chaîne cinématique fermée

On appelle chaîne fermée une chaîne ouverte dont les deux solides extrêmes ont une liaison entre eux. Les $n+1$ solides sont reliés par $n+1$ liaisons. Un Mécanisme en chaîne cinématique fermée dont

L'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes (voir figure 3-5). [35]

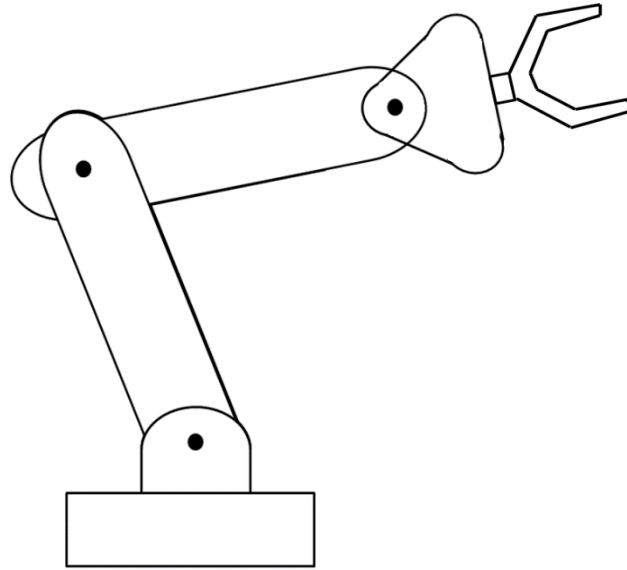


Figure 3-5 Chaîne cinématique fermée

3.2.6 Morphologie des robots manipulateurs :

Les chaînes cinématiques ouvertes simples sont abordées dans ce paragraphe. Deux paramètres sont pris en compte pour énumérer les architectures possibles : le type d'articulation, rotoïde (R) ou prismatique (P), et l'angle formé par deux axes articulaires consécutifs, soit 0° , soit 90° . Les axes successifs d'un robot sont généralement parallèles ou perpendiculaires, à l'exception de cas très spécifiques.

Les trois premiers degrés de liberté (D.D.L) sont désignés comme le porteur du robot. Le poignet est composé de D.D.L résiduels, qui se caractérise par des dimensions considérablement réduites et une masse inférieure. Les douze morphologies possibles pour le porteur sont présentées dans la figure 3-6 Ces structures ne se répètent pas, en l'absence de structures qui limitent les mouvements du porteur à des déplacements linéaires. ou planaires, comme trois liaisons prismatiques aux axes parallèles ou trois liaisons rotoïdes également parallèles. [15]

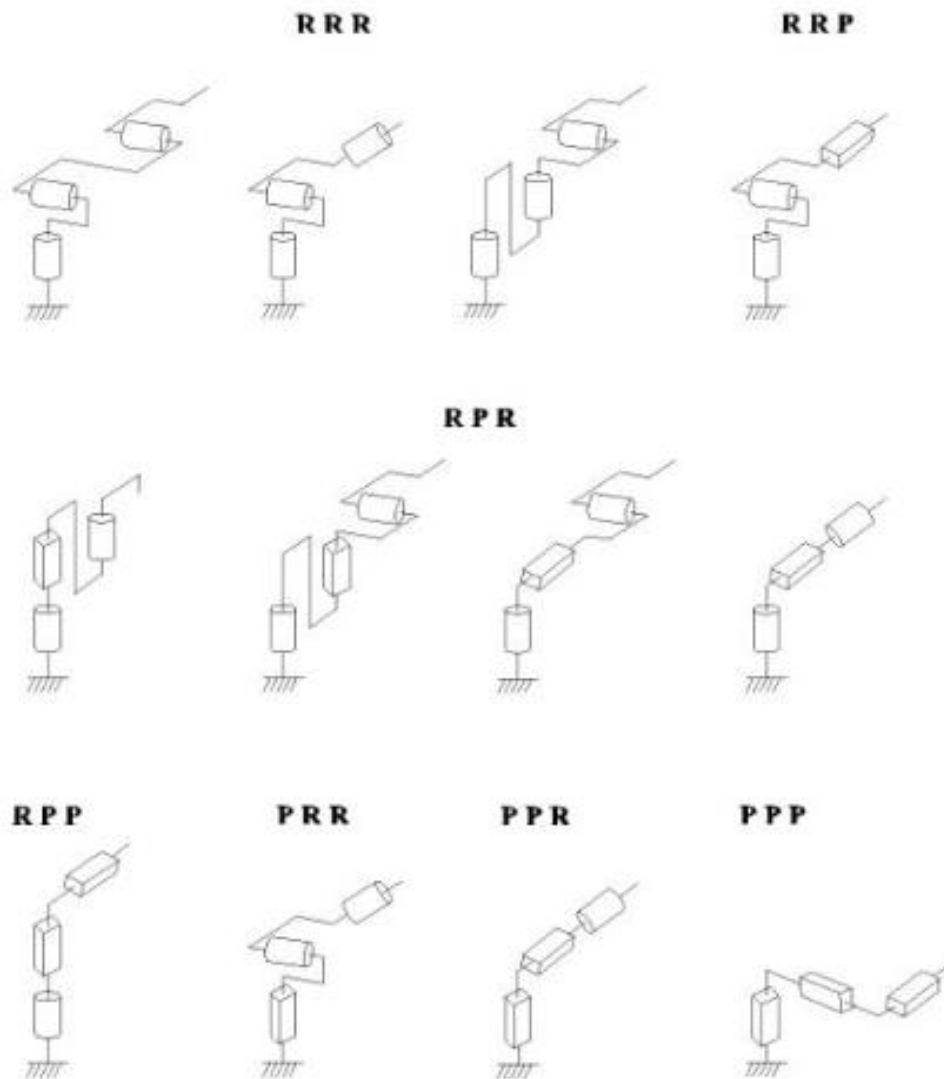


Figure 3-6 des modèles cinématique

3.2.7 Position et orientation d'un solide

Si pour décrire la position d'un point, trois dimensions sont nécessaires et suffisantes, ce n'est plus le cas pour décrire la position d'un solide et son orientation. Il faut trois

Coordonnées pour positionner le centre d'inertie des solides et trois autres coordonnées pour son orientation dans l'espace. Six coordonnées sont donc nécessaires et suffisantes pour placer un objet dans l'espace. [13]

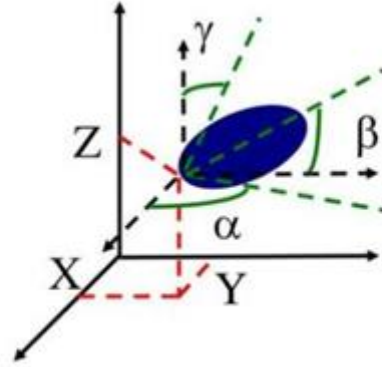


Figure 3-7 Position d'un solide dans l'espace et son repère associé

3.3 Modèle géométrique direct

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

Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation

${}^0T_n$:

$${}^0T_n = {}^0T_1(q_1) * {}^1T_2(q_2) \dots * {}^{n-1}T_n(q_n)$$

Le modèle géométrique direct du robot peut aussi être représenté par la relation :

$$X = f(q)$$

où q est le vecteur des variables articulaires tel que :

$$q = [q_1 q_2 \dots q_n]^T$$

Les coordonnées opérationnelles sont définies par :

$$X = [x_1 x_2 \dots x_m]^T$$

Il existe plusieurs possibilités pour la définition du vecteur X . Par exemple, avec les éléments de la matrice ${}^0T_n$:

$$X = [P_x P_y P_z S_x S_y S_z n_x n_y n_z a_x a_y a_z]^T$$

Ou bien, sachant que $s = n*a$:

$$X = [P_x P_y P_z n_x n_y n_z a_x a_y a_z]^T$$

Pour les rotations, d'autres représentations sont couramment utilisées comme les angles d'Euler, les angles de roulis-tangage-lacet (RTL) ou les paramètres d'Euler (quaternions). On

peut facilement passer des cosinus directeurs S,N,A à l'une quelconque de ces représentations et inversement.

Pour notre étude d'un bras manipulateur de trois degrés de liberté (ROTATION, ROTATION, ROTATION) (RRR) (Figure 4-8), on utilise la méthode de Denavit-Hartenberg pour calculer la matrice de transformation globale.[17]

Pour décrire la cinématique du manipulateur, nous établirons un référentiel pour chaque bras du robot et décrirons leur position relative en utilisant les paramètres de Denavit-Hartenberg. On obtient donc très facilement la description cinématique du manipulateur complet en utilisant la récurrence. [17]

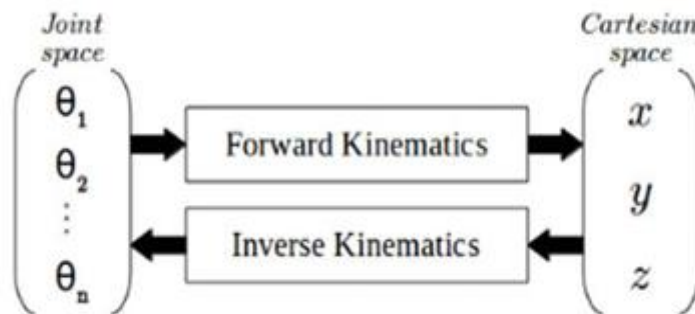


Figure 3-8 Relation entre le modèle géométrique direct et inverse

3.4 Paramètres Denavit–Hartenberg :

Il faut définir les repères de coordonnées sur lesquels ces transformations seront appliquées avant de pouvoir calculer les transformations homogènes entre les articulations adjacentes. Les paramètres de Denavit-Hartenberg (DH) se composent de quatre valeurs qui représentent les mouvements et les déplacements entre les articulations voisines. Ces paramètres créent un système de référence pour assigner des systèmes de coordonnées aux éléments d'un robot manipulateur. La figure ci-dessous illustre la convention modifiée, également appelée convention selon Khalil. [11] et [12]

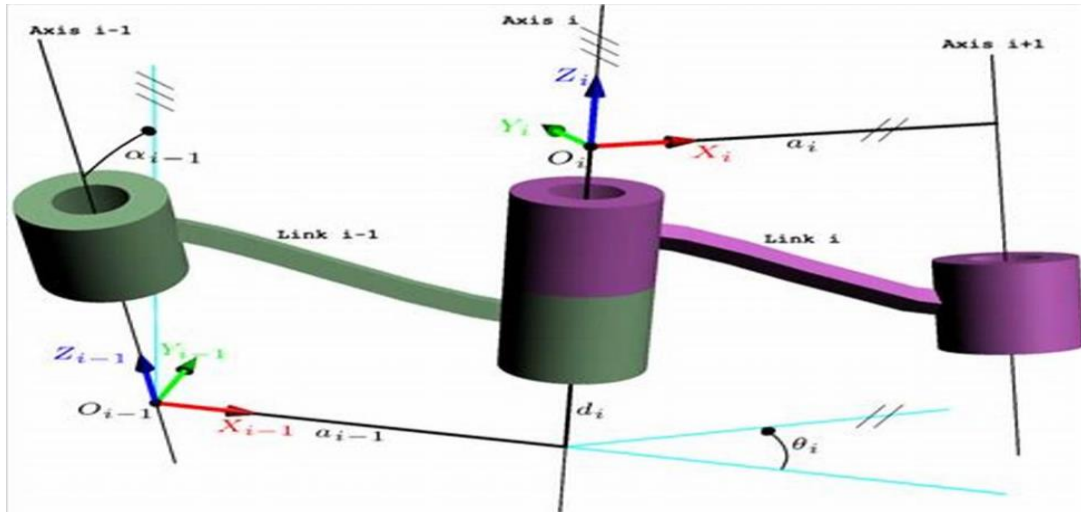


Figure 3-9 : les paramètres de Denavit-Hartenberg.

Les paramètres sont définis comme suit [11] et [12] :

α_{i-1} : angle de torsion entre les axes z des liens $i-1$ et i (mesuré autour de x_{i-1} selon la règle de la main droite).

d_{i-1} : distance de translation entre les axes z des liaisons $i-1$ et i (mesurée selon x_{i-1}).

a_{i-1} : distance de translation entre les axes x des liens $i-1$ et i (mesurée selon z_i).

θ_i : angle de rotation entre les axes x des liens $i-1$ et i (mesuré autour de z_i selon la règle de la main droite).

L'origine d'un repère iii est définie par l'intersection de x_i et z_i .

Les axes x définissent les normales communes entre z_{i-1} et z_i .

Pour calculer la position de l'effecteur terminal par rapport à un repère de référence de base, les transformations entre les liens adjacents sont composées comme suit :

$${}^0_N T = {}^0_1 T {}^1_2 T {}^2_3 T {}^3_4 T \dots \dots \dots {}^{N-1}_N T$$

Le point de départ est marqué par 0 et celui de l'effecteur terminal par N. La transformation homogène qui projette le repère N sur le repère 0 est ainsi définie par ${}^0_N T$. Plus précisément, la référence propose une conversion unique entre les articulations $i-1$ et i . Plus précisément, il y a une seule conversion entre les liens $i-1$ et i fournie par [12].

$${}^{i-1}_i T = R_{x(a_i)} D_{x(d_i)} R_{z(\theta_i)} D_{z(a_{i+1})}$$

De cette façon les matrices de transformation s'écrivent :

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a_i) & -\sin(a_i) & 0 \\ 0 & \sin(a_i) & \cos(a_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finalement, nous obtenons la matrice de transformation du repère $i-1$ au repère i :

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\cos(a_i)\sin(\theta_i) & \sin(a_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(a_i)\cos(\theta_i) & -\sin(a_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(a_i) & \cos(a_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On peut observer cette transformation en deux rotations R de grandeurs α et θ , ainsi que deux déplacements D de grandeurs d et r .

Le processus d'attribution des paramètres pour les chaînes cinématiques ouvertes à n degrés de liberté (les articulations) est résumé comme suit [11] et [12] :

Attribuer toutes les articulations $\{1, 2, \dots, n\}$.

Attribuer tous les repères : $\{0, 1, 2, \dots, n\}$, avec le repère de base numéroté 0.

Tracer des lignes à travers toutes les articulations, définissant les axes des articulations.

Attribuer l'axe Z de chaque repère pour qu'il pointe le long de son axe d'articulation.

Identifier la normale commune** entre chaque deux axes Z_{i-1} et Z_i .

Pour i de 1 à $n-1$, attribuer l'axe x_i :

Pour les axes obliques, le long de la normale entre Z_i et Z_{i-1} et pointant de i à $i+1$

Pour les axes en intersection, normal au plan contenant Z_i et Z_{i-1}

Pour les axes parallèles ou coïncidents, l'affectation est arbitraire.

Pour le repère de base, choisir toujours le repère $\{0\}$ de sorte qu'il coïncide avec le repère $\{1\}$ lorsque la première variable d'articulation θ_1 ou d_1 est égale à zéro.

Pour le repère de l'effecteur terminal, si l'articulation n est rotoïde, choisir x_n dans la direction de x_{n-1} lorsque $\theta_n = 0$ et l'origine du repère $\{n\}$ de sorte que $d_n = 0$.

Des changements sont apportés à la position des caractéristiques par rapport à l'accord DH original, comme le montre le tableau de comparaison Tableau 3

Critère	HD	PAUL
repère i en liaison i	1	1
θ_i et d_i autour de l'axe Z_i	1	0
a_i et r_i autour de l'axe X_i	0	0

Table 3 le tableau de comparaison

3.5 Modèle géométrique indirect

Les coordonnées opérationnelles, c'est-à-dire la position et l'orientation de l'organe terminal, peuvent être calculées en utilisant le modèle géométrique direct d'un robot en fonction des coordonnées articulaires. Le modèle géométrique inverse (MGI) est un problème inverse qui vise à établir les coordonnées articulaires correspondant à une situation spécifique de l'organe terminal.

Une solution générale pour les robots à six articulations est proposée par la méthode générale de Raghavan et Roth, qui consiste à résoudre un polynôme de degré au plus égal à 16. Toutefois, une forme explicite du MGI n'est pas toujours accessible. Dans ces situations, il est possible d'obtenir une solution spécifique en utilisant des procédures numériques telles que celles développées par Pieper, Whitney, Fournier, Featherstone, Wolovich, Goldenberg, et Sciavicco.

La forme explicite qui donne toutes les solutions possibles du MGI (il y a rarement unicité de solution) constitue le modèle géométrique inverse. On peut distinguer trois méthodes de calcul du MGI :

La méthode de Paul : Cette méthode traite séparément chaque cas particulier et est adaptée pour la plupart des robots industriels.[18]

La méthode de Pieper : Cette méthode permet de résoudre le problème pour les robots à six degrés de liberté ayant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques. [20]

La méthode générale de Raghavan et Roth : Cette méthode permet de trouver une solution générale pour les robots à six articulations à partir d'un polynôme de degré au plus égal à 16. [19]

Ces méthodes permettent de calculer les coordonnées articulaires nécessaires pour

atteindre une position et orientation données de l'organe terminal du robot.

3.6 Modélisation cinématique :

La cinématique est la discipline qui étudie le mouvement sans prendre en compte les forces qui le provoquent. La position, la vitesse, l'accélération et toutes les dérivées d'ordre supérieur des variables de position sont étudiées dans cette discipline. Les manipulateurs sont constitués de liens rigides (chaînes) reliés par des articulations qui permettent le mouvement relatif des liens adjacents. En général, ces articulations sont équipées de capteurs de position qui permettent de mesurer la position relative des liaisons voisines. Deux sortes d'articulations existent : rotoïdes et prismatiques. Les mouvements des articulations rotoïdes sont des rotations (angles), tandis que les mouvements des articulations prismatiques sont des translations (déplacements linéaires). [11] et [12].

3.6.1 . Modélisation cinématique direct :

Qui consiste à calculer les vitesses des coordonnées opérationnelles $\dot{X}$, en fonction de vitesses des coordonnées généralisées $\dot{q}$, ce modèle est noté:

$$\dot{X} = J(q)\dot{q}$$

(J est la matrice jacobienne du robot manipulateur).

On peut obtenir la matrice jacobienne par une méthode de calcul direct, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation V_n et ω_n du repère R_n , et les vitesses articulaires $\dot{q}$:

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J_n \dot{q}$$

3.6.2 Modèle cinématique inverse :

Le but du modèle cinématique inverse est de déterminer les vitesses articulaires qui garantissent au repère terminal une vitesse opérationnelle donnée, à partir d'une configuration q spécifique. Le modèle différentiel inverse est similaire à cette définition : il permet de calculer la différentielle articulaire dq correspondant à une différentielle des coordonnées opérationnelles dX spécifiée. Afin d'obtenir le modèle cinématique inverse, on résout un système d'équations linéaires en inversant le modèle cinématique direct. [14]

La réalisation peut être réalisée de manière analytique ou numérique :

- La solution analytique présente l'avantage de réduire considérablement le nombre d'opérations, cependant il est nécessaire de traiter individuellement chaque cas spécifique.

– Les algorithmes numériques sont plus larges, la plus courante reposant sur la notion de pseudo-inverse : les méthodes algorithmiques traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important.

Nous présentons dans ce paragraphe les techniques à mettre en œuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants. [17]

3.6.3 Forme générale du modèle cinématique :

Soit $X = [X_p^T X_r^T]^T$ une représentation quelconque dans le repère R_0 de la situation du repère R_n fixé à un solide, les éléments X_p et X_r désignant respectivement la position et l'orientation opérationnelles du solide. Les relations entre les vitesses $\dot{X}_p$ et $\dot{X}_r$ et les vecteurs vitesses ${}^0V_n$ et ${}^0\omega_n$ du repère R_n sont telles que :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \begin{bmatrix} \Omega_p & \mathbf{0}_3 \\ \mathbf{0}_3 & \Omega_r \end{bmatrix} \begin{bmatrix} {}^0V_n \\ {}^0\omega_n \end{bmatrix} = \Omega \begin{bmatrix} {}^0V_n \\ {}^0\omega_n \end{bmatrix}$$

Les matrices Ω_p et Ω_r dépendant de la représentation choisie respectivement pour la position et pour l'orientation. [17]

3.7 Modélisation dynamique :

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et aux positions, vitesses et accélérations articulaires. [14]

On représente le modèle dynamique par une relation de la forme :

$$\Gamma = F(q, \dot{q}, \ddot{q}, f_e)$$

Avec, Γ : Vecteur des couples/forces des actionneurs, selon le type d'articulation (rotoïde ou prismatique).

q : Vecteur des positions articulaires.

$\dot{q}$: Vecteur des vitesses articulaires.

$\ddot{q}$: Vecteur des accélérations articulaires

f_e : Vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement. Le modèle dynamique direct (MDD) exprime les accélérations articulaires en fonction des positions, des vitesses et des couples articulaires :

$$\ddot{q} = F(q, \dot{q}, \Gamma, f_e)$$

3.8 Application sur le bras robotique :

3.8.1 Description du robot à commander

Le robot à contrôler est un bras manipulateur composé principalement de six parties pouvant tourner successivement autour des axes Z_1 , Z_2 , Z_3 , Z_4 , Z_5 et Z_6 . La sixième partie contient l'élément terminal, qui est une pince à deux doigts. La rotation de θ_1 , θ_2 , θ_3 , θ_4 et θ_5 ainsi que l'ouverture et la fermeture de la pince sont assurées par des actionneurs électriques (servomoteurs)

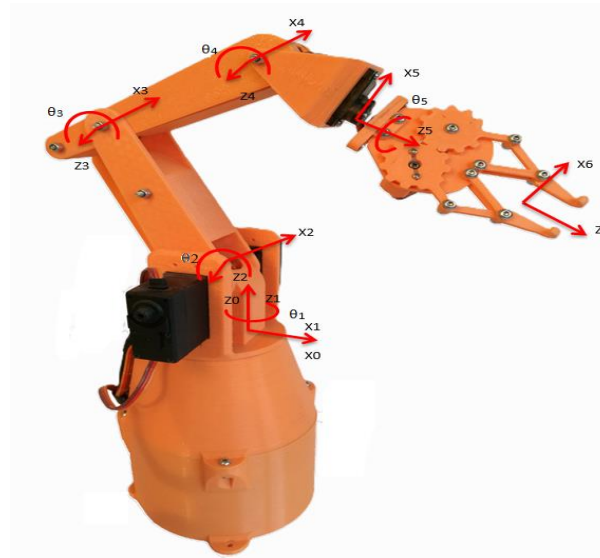


Figure 3-10 le modèle Denavit-Hartenberg(DH)

Le bras robotique à 5 degrés de liberté, La figure suivante montre la configuration des axes pour le modèle géométrique direct selon la convention de DH.

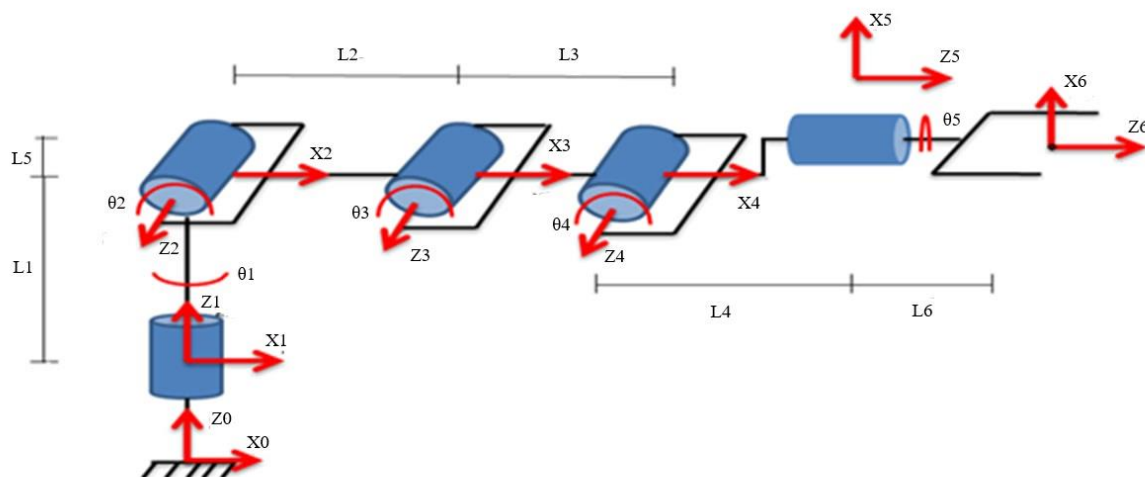


Figure 3-11 Représentation selon la convention DH

Les paramètres géométriques de Denavit-Hertenberg(DH) du robot à commander sont donnés dans le tableau suivant :

	θ_i	d_i	a_i	α_i
1	θ_1	0	0	0
2	θ_2	L1	0	$-\frac{\pi}{2}$
3	θ_3	0	L2	0
4	θ_4	0	L3	0
5	θ_5	L4	L5	$\frac{\pi}{2}$
6	0	L6	0	0

Table 4 Tableau des paramètres de Denavit-Hertenberg(DH)

3.8.3 Modèle géométrique direct :

En utilisant les valeurs du tableau précédent de la matrice DH, on obtient les matrices suivantes de transformation homogène pour chaque axe de coordonnées :

Les matrices de transféré :

Les matrices de transformations homogènes sont données comme suit :

Le calcul des matrices a été fait sous Matlab

```

matric2.m x matric3.m x +
1  disp('This program is used to get the transformation matrix of robot')
2  syms t1 t2 t3 t4 t5 t6 l1 l2 l3 l4 l5 l6 ;
3
4  %% DH parameters
5  alpha = [0, pi/2, 0, 0, -pi/2, 0];
6  a = [0, 0, l2, l3, l5, 0];
7  d = [0, l1, 0, 0, l4, l6];
8  t = [t1, t2, t3, t4, t5, 0];
9
10 %% Transformation matrices
11 M = sym(zeros(4, 4, 6));
12 for i = 1:6
13     M(:, :, i) = [cos(t(i)), -sin(t(i))*cos(alpha(i)), sin(t(i))*sin(alpha(i)), a(i)*cos(t(i));
14                  sin(t(i)), cos(t(i))*cos(alpha(i)), -cos(t(i))*sin(alpha(i)), a(i)*sin(t(i));
15                  0, sin(alpha(i)), cos(alpha(i)), d(i);
16                  0, 0, 0, 1];
17 end
18
19 %% Multiplying transformation matrices
20 M10 = M(:, :, 1);
21 M21 = M(:, :, 2);
22 M32 = M(:, :, 3);
23 M43 = M(:, :, 4);
24 M54 = M(:, :, 5);
25 M65 = M(:, :, 6);
26
27 M20 = M10 * M21;
28 M30 = M20 * M32;
29 M40 = M30 * M43;
30 M50 = M40 * M54;
31 M60 = M50 * M65;
32
33 %% Simplify the final transformation matrix
34 M60 = simplify(M60);
35
36 %% Display final transformation matrix from base to end-effector
37 disp('Transformation matrix from base to end-effector:');
38 disp(M60);
39

```

Figure 3-12 Calcul de matrices via matlab

$$t1= \theta_1, t2= \theta_2, t3= \theta_3, t4= \theta_4, t5= \theta_5, t6= \theta_6$$

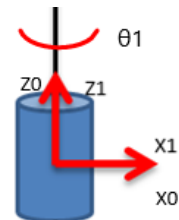
Après exécution du programme nous trouvons :

La matrice de transformation homogène de l'organe terminal par rapport à la base du robot est donnée par :

$$\mathbf{M}^{0,6} = \mathbf{M}^{0,1} * \mathbf{M}^{1,2} * \mathbf{M}^{2,3} * \mathbf{M}^{3,4} * \mathbf{M}^{4,5} * \mathbf{M}^{5,6}, \text{ ce qui donne :}$$

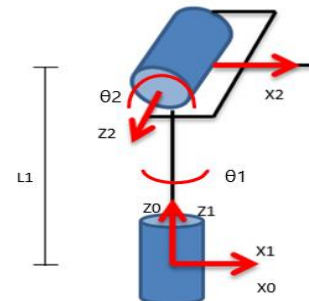
1-Pour le repère de l'origine (R0) à la liaison de rotation(R1) :

$$\mathbf{M}^{0,1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



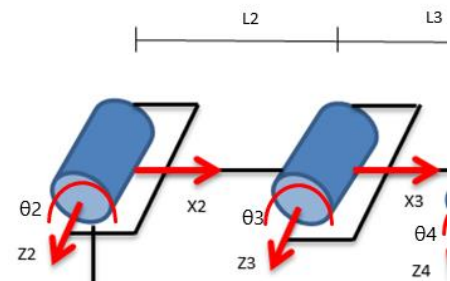
2- Pour le repère de l'origine (R1) à la liaison de rotation(R2) :

$$\mathbf{M}^{1,2} = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



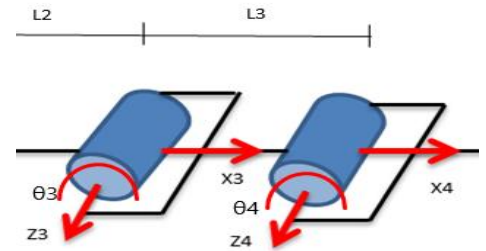
3-Pour le repère de l'origine (R2) à la liaison de rotation(R3) :

$$\mathbf{M}^{2,3} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_2 * \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_2 * \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



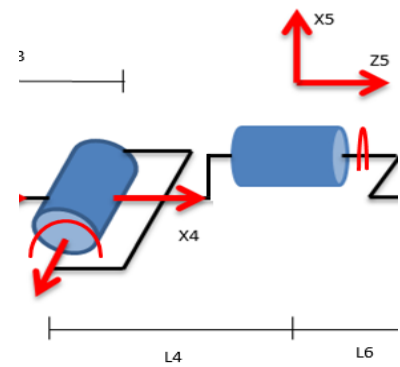
4-Pour le repère de l'origine (R3) à la liaison de rotation(R4) :

$$M^{3,4} = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & l_3 * \cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & l_3 * \sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



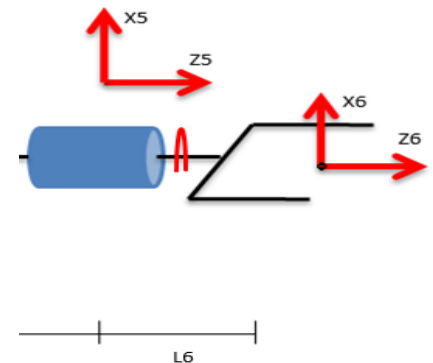
5-Pour le repère de l'origine (R4) à la liaison de rotation (R5) :

$$M^{4,5} = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) & l_5 * \cos(\theta_5) \\ \sin(\theta_5) & 0 & -\cos(\theta_5) & l_5 * \sin(\theta_5) \\ 0 & 1 & 0 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



6- Pour le repère (R5) à la liaison de Translation(P6) :

$$M^{5,6} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$M^{0,2} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & 0 & -\sin(\theta_1 + \theta_2) & 0 \\ \sin(\theta_1 + \theta_2) & 0 & \cos(\theta_1 + \theta_2) & 0 \\ 0 & -1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M^{0,3} =$$

$$\begin{bmatrix} \cos(\theta_3) * \cos(\theta_1 + \theta_2) & -(\cos(\theta_1 + \theta_2) \sin(\theta_3)) & -\sin(\theta_1 + \theta_2) & l_2 * (\cos(\theta_1 + \theta_2) * \cos(\theta_3)) \\ \cos(\theta_3) * (\sin(\theta_1 + \theta_2)) & -\sin(\theta_1 + \theta_2) * \sin(\theta_3) & \cos(\theta_1 + \theta_2) & l_2 * \sin(\theta_1 + \theta_2) * \cos(\theta_3) \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & l_1 - l_2 * \sin(\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M^{0,4} =$$

$$\begin{bmatrix} \cos(\theta_3 + \theta_4) * \cos(\theta_1 + \theta_2) & -(\cos(\theta_1 + \theta_2) \sin(\theta_3 + \theta_4)) & -\sin(\theta_1 + \theta_2) & l_2 * (\cos(\theta_1 + \theta_2) * (l_3 * \cos(\theta_3 + \theta_4) + l_2 * \cos(\theta_3))) \\ \cos(\theta_3 + \theta_4) * (\sin(\theta_1 + \theta_2)) & -\sin(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) & \cos(\theta_1 + \theta_2) & l_2 * \sin(\theta_1 + \theta_2) * (l_3 * \cos(\theta_3 + \theta_4) + l_2 * \cos(\theta_3)) \\ -\sin(\theta_3 + \theta_4) & -\cos(\theta_3 + \theta_4) & 0 & l_1 - l_2 * \sin(\theta_3 + \theta_4) - l_2 * \sin(\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M^{0,5} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$$

$$T_{11} = \cos(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2)$$

$$T_{12} = -\sin(\theta_1 + \theta_2)$$

$$T_{13} = \sin(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2)$$

$$T_{14} = \cos(\theta_1 + \theta_2) * (l_3 * \cos(\theta_3 + \theta_4) + l_2 * \cos(\theta_3)) - l_4 * \sin(\theta_1 + \theta_2) - l_5 * \cos(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5) + l_5 * \cos(\theta_1 + \theta_2) * \cos(\theta_3 + \theta_4) * \cos(\theta_5)$$

$$T_{21} = \cos(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2)$$

$$T_{22} = \cos(\theta_1 + \theta_2)$$

$$T_{23} = \sin(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2)$$

$$T_{24} = l_4 * \cos(\theta_1 + \theta_2) + \sin(\theta_1 + \theta_2) * (l_3 * \cos(\theta_3 + \theta_4) + l_2 * \cos(\theta_3)) + l_5 * \cos(\theta_3 + \theta_4) * \sin(\theta_1 + \theta_2) * \cos(\theta_5) - l_5 * \sin(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5)$$

$$T_{31} = -\sin(\theta_3 + \theta_4 + \theta_5)$$

$$T_{32} = 0$$

$$T_{33} = \cos(\theta_3 + \theta_4 + \theta_5)$$

$$T_{34} = l_1 - l_3 * \sin(\theta_3 + \theta_4) - l_2 * \sin(\theta_3) - l_5 * \sin(\theta_3 + \theta_4 + \theta_5)$$

$$T_{41} = 0$$

$$T_{42} = 0$$

$$T_{43} = 0$$

$$T_{44} = 1$$

$$M^{0,6} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$$

$$T_{11} = \cos(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2)$$

$$T_{12} = -\sin(\theta_1 + \theta_2)$$

$$T_{13} = \sin(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2)$$

$$T_{14} = \cos(\theta_1 + \theta_2) * (13 * \cos(\theta_3 + \theta_4) + 12 * \cos(\theta_5)) - 14 * \sin(\theta_1 + \theta_2) + 16 * \sin(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2) - 15 * \cos(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5) + 15 * \cos(\theta_1 + \theta_2) * \cos(\theta_3 + \theta_4) * \cos(\theta_5)$$

$$T_{21} = \cos(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2)$$

$$T_{22} = \cos(\theta_1 + \theta_2)$$

$$T_{23} = \sin(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2)$$

$$T_{24} = 14 * \cos(\theta_1 + \theta_2) + \sin(\theta_1 + \theta_2) * (13 * \cos(\theta_3 + \theta_4) + 12 * \cos(\theta_5)) + 16 * \sin(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2) + 15 * \cos(\theta_3 + \theta_4) * \sin(\theta_1 + \theta_2) * \cos(\theta_5) - 15 * \sin(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5)$$

$$T_{31} = -\sin(\theta_3 + \theta_4 + \theta_5)$$

$$T_{32} = 0$$

$$T_{33} = \cos(\theta_3 + \theta_4 + \theta_5)$$

$$T_{34} = 11 - 13 * \sin(\theta_3 + \theta_4) - 12 * \sin(\theta_5) + 16 * \cos(\theta_3 + \theta_4 + \theta_5) - 15 * \sin(\theta_3 + \theta_4 + \theta_5)$$

$$T_{41} = 0$$

$$T_{42} = 0$$

$$T_{43} = 0$$

$$T_{44} = 1$$

Modèle Géométrique Inverse

Lorsqu'on génère des mouvements dans l'espace opérationnel d'une position X1 (X1, Y1, Z1, Rx1, Ry1, Rz1) à une position X2 (X2, Y2, Z2, Rx2, Ry2, Rz2), il est nécessaire de connaître les coordonnées articulaires qqq correspondant à ces positions. Cela nécessite d'inverser le modèle géométrique direct. Pour déterminer le modèle géométrique d'un bras manipulateur avec pince, nous implémenterons la méthode de Paul, proposée par Richard P. Paul en 1980, qui se base sur une série d'équations successives.

Étapes de Calcul

Extraction des valeurs X, Y et Z de la matrice $M^{0,6}$

À partir de la matrice finale $M^{0,6}$:

$$M^{0,6} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} \rightarrow M^{0,6} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & X \\ m_{21} & m_{22} & m_{23} & Y \\ m_{31} & m_{32} & m_{33} & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nous avons donc les valeurs suivantes :

$$X = \cos(\theta_1 + \theta_2) * (13 * \cos(\theta_3 + \theta_4) + 12 * \cos(\theta_5)) - 14 * \sin(\theta_1 + \theta_2) + 16 * \sin(\theta_3 + \theta_4 + \theta_5) * \cos(\theta_1 + \theta_2) - 15 * \cos(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5) + 15 * \cos(\theta_1 + \theta_2) * \cos(\theta_3 + \theta_4) * \cos(\theta_5)$$

$$Y = 14 * \cos(\theta_1 + \theta_2) + \sin(\theta_1 + \theta_2) * (13 * \cos(\theta_3 + \theta_4) + 12 * \cos(\theta_5)) + 16 * \sin(\theta_3 + \theta_4 + \theta_5) * \sin(\theta_1 + \theta_2) + 15 * \cos(\theta_3 + \theta_4) * \sin(\theta_1 + \theta_2) * \cos(\theta_5) - 15 * \sin(\theta_1 + \theta_2) * \sin(\theta_3 + \theta_4) * \sin(\theta_5)$$

$$Z = 11 - 13 * \sin(\theta_3 + \theta_4) - 12 * \sin(\theta_5) + 16 * \cos(\theta_3 + \theta_4 + \theta_5) - 15 * \sin(\theta_3 + \theta_4 + \theta_5)$$

Calcul des Angles θ

Matrice de Rotation Finale $R^{0,6}$

La matrice de rotation représente la rotation de l'effecteur final et est constituée des parties 3×3 de la matrice de transformation :

$$R^{0,6} = \begin{bmatrix} \cos(\theta_3 + \theta_4 + \theta_5) \cos(\theta_1 + \theta_2) & \cos(\theta_3 + \theta_4 + \theta_5) \sin(\theta_1 + \theta_2) & -\sin(\theta_3 + \theta_4 + \theta_5) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 \\ \sin(\theta_3 + \theta_4 + \theta_5) \cos(\theta_1 + \theta_2) & \sin(\theta_3 + \theta_4 + \theta_5) \sin(\theta_1 + \theta_2) & \cos(\theta_3 + \theta_4 + \theta_5) \end{bmatrix}$$

Pour extraire les angles θ à partir de ces éléments, nous pouvons suivre les étapes suivantes :

Calcul de $\theta_1 + \theta_2$

De T_{12} et T_{22} , nous obtenons :

$$\theta_1 + \theta_2 = \text{atan2}(-T_{12}, T_{22})$$

Calcul de $\theta_3 + \theta_4 + \theta_5$

$$\theta_3 + \theta_4 + \theta_5 = \text{atan2}(-T_{31}, T_{33})$$

Calcul de $\theta_3 + \theta_4$

Pour $\theta_3 + \theta_4$, nous devons utiliser des éléments supplémentaires de T_{14} ou T_{24} .

Nous pouvons écrire des équations en termes de T_{14} et T_{24} puis résoudre pour $\theta_3 + \theta_4$

Calcul de θ_3

Une fois que nous avons $\theta_3 + \theta_4 + \theta_5$ et $\theta_3 + \theta_4$, nous pouvons trouver θ_5 par soustraction :

$$\theta_5 = (\theta_3 + \theta_4 + \theta_5) - (\theta_3 + \theta_4)$$

Calcul de θ_4

Avec θ_5 trouvé, nous pouvons ensuite isoler θ_3 et θ_4 en utilisant des équations supplémentaires des termes T_{14} et T_{24}

Résumé

$$\theta_1 + \theta_2 = \text{atan2}(-T_{12}, T_{22})$$

$$\theta_3 + \theta_4 + \theta_5 = \text{atan2}(-T_{31}, T_{33})$$

$\theta_3 + \theta_4$ se trouve à partir des termes T_{14} et T_{24}

θ_5 est la différence entre $\theta_3 + \theta_4 + \theta_5$ et $\theta_3 + \theta_4$

Les valeurs restantes de θ_3 et θ_4 peuvent être résolues en utilisant les autres termes de la matrice.

Avec ces étapes, nous avons une méthode systématique pour calculer les angles θ à partir de la matrice $M^{0,6}$

3.9 Conclusion

Dans ce chapitre, nous avons exploré en profondeur les principes fondamentaux de la cinématique appliqués à la conception d'un bras de robot manipulateur. Nous avons analysé les différents mouvements permis par les articulations et les segments du bras, en nous concentrant sur les aspects cinématiques directs et inverses. Cette étude nous a permis de comprendre comment les mouvements souhaités dans l'espace de travail sont traduits en commandes spécifiques pour les moteurs et les actionneurs.

La maîtrise de la cinématique est cruciale pour la conception de robots manipulateurs efficaces, car elle assure non seulement la précision des mouvements, mais aussi l'optimisation des trajectoires pour minimiser les efforts et maximiser la vitesse d'exécution.

Chapitre IV : Réalisation et Application

4.1 Introduction

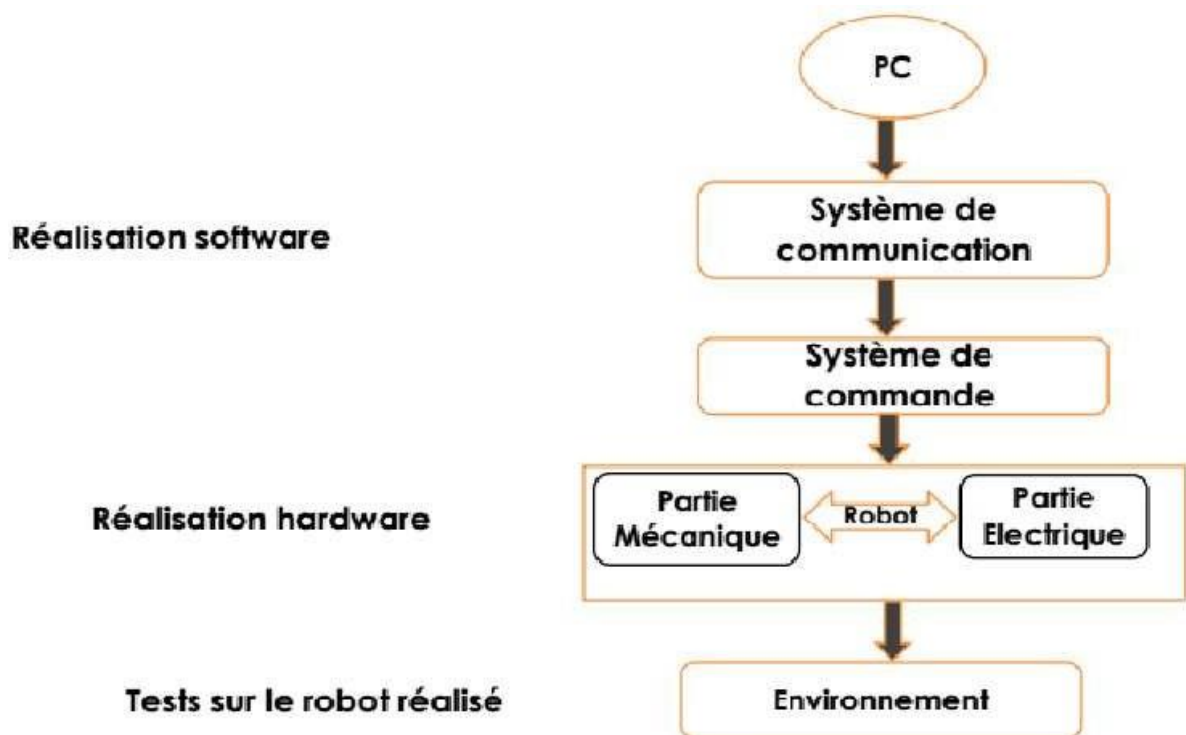
Les chapitres précédents se sont concentrés sur l'analyse des éléments essentiels d'un robot qui constituent les fondements de la construction de notre bras.

Au cours de ce chapitre, nous allons exposer la composante mécanique de notre bras manipulateur, ainsi que sa structure et ses différents segments.

La fusion de la conception et de la mise en œuvre dans un seul chapitre n'est pas un hasard.

La création de tout système, y compris un robot dans notre situation, est conçue dans le but de le concrétiser. Il est donc essentiel qu'elle prenne en considération les contraintes du terrain et les ressources disponibles, sinon tout le processus de réalisation risque d'être échoué.

Après avoir été réalisé, le robot sera mis en marche et devra accomplir différentes tâches, qui sont souvent très difficiles. Il sera donc nécessaire que la durée de vie du robot soit la plus longue possible afin de pouvoir amortir son coût de revient. Les étapes de planification et de mise en œuvre abordées dans ce chapitre revêtent donc une grande importance et doivent être réalisées de manière à réduire au minimum les imperfections de la structure et à optimiser ses performances.



Différents logiciels peuvent être employés pour concevoir, programmer et superviser un bras manipulateur, en fonction des exigences particulières du projet, tels que la modélisation 3D, la simulation, le contrôle en temps réel et la programmation des microcontrôleurs. Voici un certain nombre de logiciels fréquemment employés pour l'utilisation de bras manipulateurs CAD et Modélisation 3D

Le terme CAD, qui signifie Computer Aided Design ou CAO, fait référence à tous les logiciels informatiques de conception 3D qui sont conçus pour faciliter la création, la modification, l'analyse ou l'optimisation d'un design. Les techniques d'ingénierie assistée par ordinateur (IAO) incluent donc la CAO.

La conception assistée par ordinateur (CAO) joue un rôle essentiel dans les domaines industriels, offrant la possibilité de concevoir des pièces avec une précision inégalée grâce à des logiciels spécialisés tels que CATIA, SOLIDWORKS, Autocad, Créo ou encore Autodesk. La conception de robot :

4.2 Simulation

4.2.1 Représentation de tinkercad

TINKERCAD (Système d'Entrée Schématique Intelligent TINKERCAD) est un outil logiciel avancé pour la conception et la simulation de circuits électroniques, et fait partie de la suite logicielle Proteus. Il permet aux utilisateurs de dessiner des schémas électroniques détaillés, de tester leur comportement dans un environnement virtuel avant la mise en œuvre pratique, accélérant ainsi le processus de prototypage et réduisant les coûts. TINKERCAD s'intègre parfaitement avec le module ARES pour la conception de circuits imprimés, facilitant ainsi le flux de travail du schéma à la réalisation.

La simulation dans TINKERCAD aide à ajuster et identifier les erreurs avant de créer des prototypes physiques, réduisant ainsi les risques d'erreurs coûteuses, tout en prenant en compte que les résultats de la simulation peuvent légèrement différer des résultats réels.

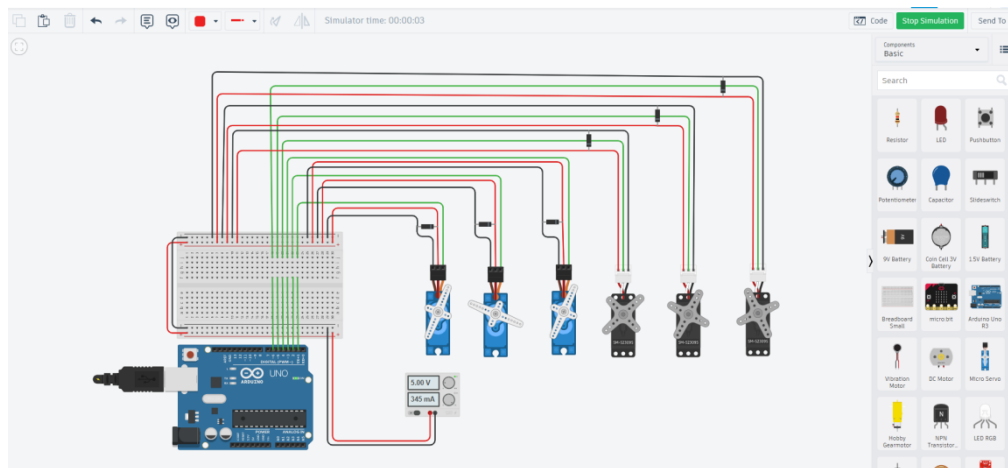


Figure 4-1 Le logiciel TINKERCAD en mode simulation du projet

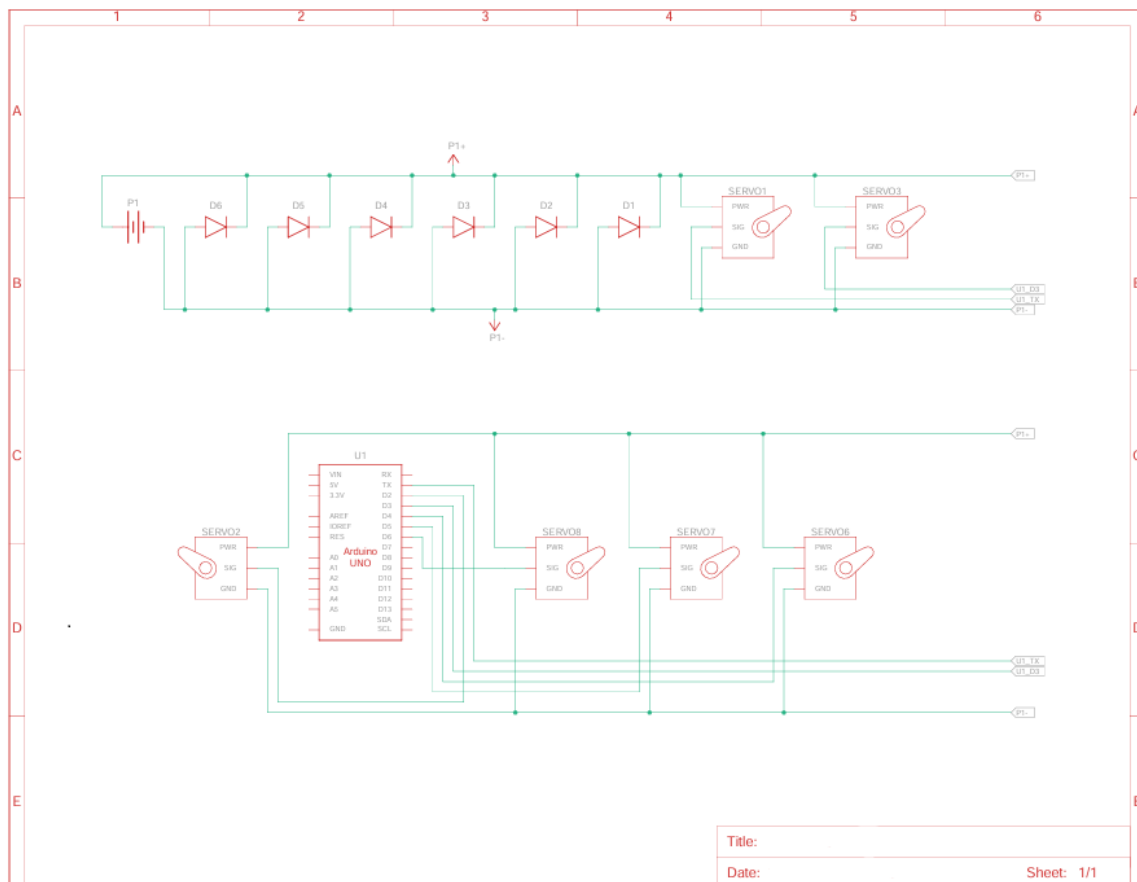


Figure 4-2 Le logiciel TINKERCAD en mode simulation du Vue schéma

Name	Quantity	Component
U1	1	Arduino Uno R3
SERVO2 SERVO3 SERVO1	3	Positional Micro Servo
SERVO6 SERVO7 SERVO8	3	Micro Servo
D1 D2 D3 D4 D5 D6	6	Diode
P1	1	5 , 5 Power Supply

Figure 4-3 Le logiciel TINKERCAD en mode simulation du Liste des composants

Après avoir connecté tous les moteurs et capteurs à l'Arduino et avant d'y connecter l'alimentation, nous programmons le code et le modifions pour qu'il fonctionne comme requis dans le programme Tinkecad afin d'éviter des problèmes dans le vrai robot.

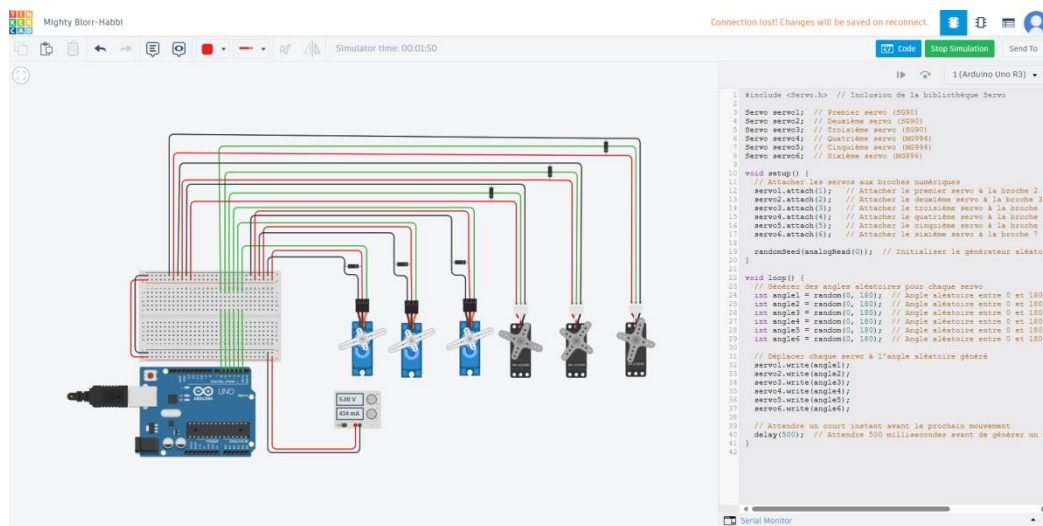


Figure 4-4 Le logiciel TINKERCAD en Code de programmation

4.2.2 Conception et impression 3D des pièces du bras robotique

Nous avons conçu les pièces du bras robotique en utilisant le logiciel SolidWorks, où chaque pièce a été soigneusement élaborée pour garantir sa compatibilité avec les autres éléments du système. Une fois la conception numérique terminée, on a transféré les dessins à l'étape de production en utilisant l'impression 3D. Cette technologie nous a permis de produire des pièces précises et conformes aux spécifications définies dans la conception originale. Chaque pièce a été fabriquée avec une grande précision, assurant ainsi une qualité supérieure et une performance optimale lors de l'assemblage complet du bras robotique

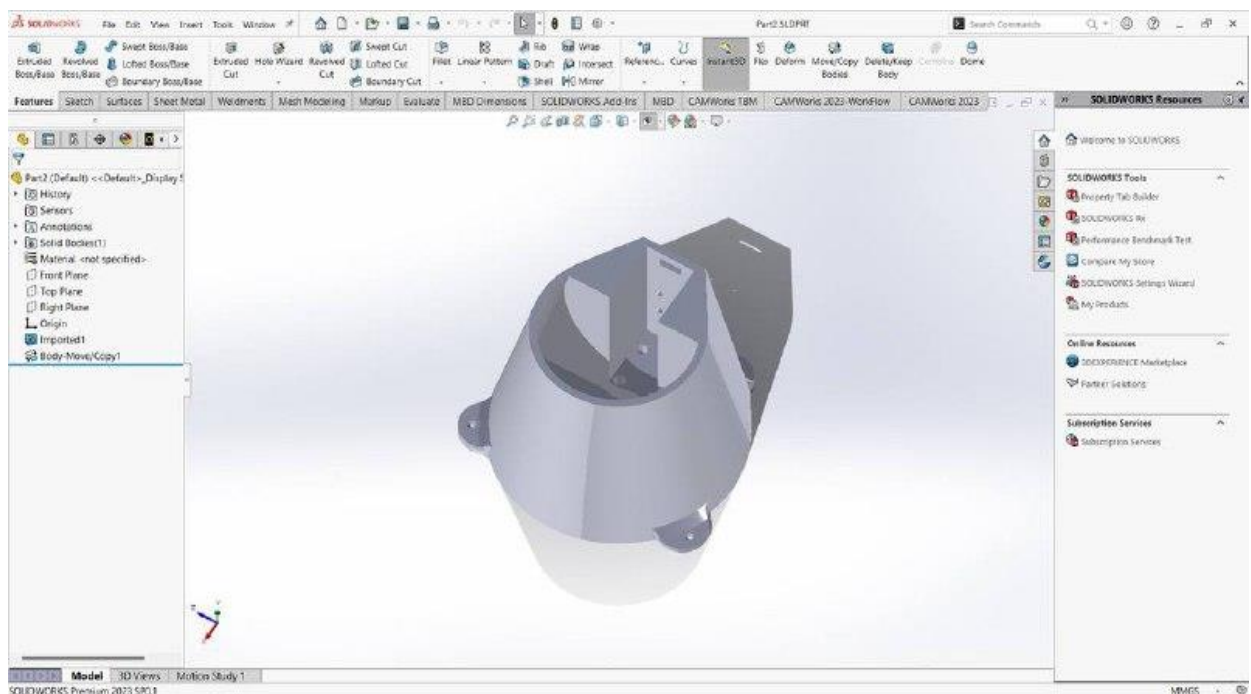


Figure 4-5 Pièces de bras via Solidworks

Après avoir conçu chaque pièce du bras robotique, nous les avons assemblées à l'aide du logiciel SolidWorks. Ensuite, nous avons effectué des tests de mouvement pour vérifier la compatibilité des pièces mobiles et leur performance fluide. Nous avons également réalisé une analyse de contrainte sur les pièces critiques pour garantir leur capacité à supporter les forces exercées lors du fonctionnement.



Figure 4-6 bras robotique en utilisant le logiciel SolidWorks

4.3 Réalisation pratique

Dans cette partie de notre étude, nous examinerons les différents composants d'un bras manipulateur et comment les assembler. Nous effectuerons également des tests pour évaluer ses performances.

Structure mécanique du bras de lucarne

Nous avons choisi le plastique en raison de sa disponibilité, de sa faible densité et de sa capacité à le former et à l'imprimer dans la machine d'impression 3D.

Concevoir une structure mécanique selon un modèle 5 degrés de liberté.



Figure 4-7 imprimante 3D

4.3.1 Assemblage du bras manipulateur :

Dans les paragraphes précédents, nous avons examiné la structure de la main en mouvement et le modèle mathématique qui décrit son mouvement.

Nous allons maintenant commencer à pratiquer le processus, en commençant par assembler des pièces à main mixtes et souder tous les composants sur la bande Arduino.



Figure 4-8 les pièces imprimés par l'imprimante 3D

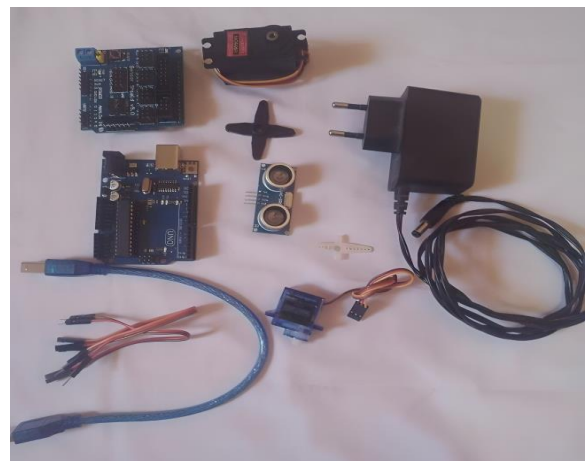


Figure 4-9 les pièces électriques

Pour simplifier l'assemblage des composants mécaniques, le modèle CAO réalisé sur Solidworks a servi de référence et pour faciliter la vérification des degrés de liberté du bras du processeur. Après avoir assemblé les éléments présentés dans les figures VI.1 et VI.2, des tests fonctionnels ont été effectués pour éviter tout problème lors de la programmation sur Arduino Uno.

- La première base

La première base est conçue comme un cylindre ouvert en haut, ce qui permet de fixer le servomoteur avec des boulons.



Figure 4-10 La première base pour fixer le servomoteur avec des boulons

- Premier corps

Il s'agit d'une pièce en plastique en forme de U, reliée à un servomoteur ancré sur la base. Une cavité est ménagée pour accueillir le servomoteur, garantissant la stabilité de la structure.



Figure 4-11 Premier corps

- Bras inférieur

Il est conçu à partir de cellules en plastique et se compose de deux parties égales de $10 * 2.5 * 2.5$ cm, reliées par un servomoteur dans la partie précédente.



Figure 4-12 bras inférieur

- Bras supérieur

Il s'agit d'une pièce en plastique, composée de deux parties identiques de $13 * 2.5 * 2.5$ cm, dont une extrémité est reliée à un servomoteur relié au deuxième corps, et l'autre extrémité est reliée à l'articulation de la main,

L'emballage a été découpé en une seule partie pour créer un trou pour l'entrée du servomoteur, assurant l'articulation manuelle



Figure 4-13 Le bras supérieur

- Assemblage à la main

Il s'agit d'une pièce en plastique en forme de U de dimensions 3*4*3 cm. L'emballage a été découpé en une seule partie pour créer un trou pour l'entrée du servomoteur, assurant l'articulation manuelle



Figure 4-14 assemblage à la main

- Poignée à main

C'est une poignée en plastique constituée de deux doigts, dont un fixe et l'autre articulé par un servomoteur TowerPro SG90, conçue pour maintenir un objet.



Figure 4-15 Poignée à main

- Arduino Uno

Programmation de l'Arduino Uno après avoir connecté les pièces électronique

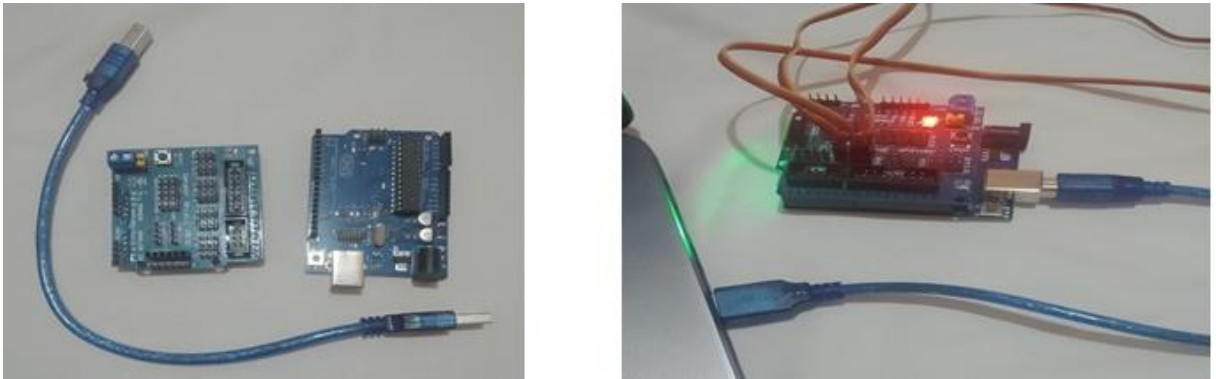


Figure 4-16 Arduino Uno

- Servomoteur

Installer des servomoteurs dans chaque joint et les connecter à l'Arduino et à la source d'alimentation



Figure 4-17 Servomoteur

4.3.2 Utilisation de la diode

Pour protéger les servomoteurs contre les surtensions ou les retours de courant (notamment lors de la coupure de l'alimentation ou des mouvements brusques)

Où placer la diode ?

La diode doit être placée en **parallèle avec les bornes d'alimentation du servomoteur**, avec la cathode (marquée par une bande) connectée au positif de l'alimentation et l'anode au négatif (masse). Cela permettra à la diode de conduire lorsque le courant inverse apparaît, protégeant ainsi le circuit.



Figure 4-18 : diode soudé

4.3.3 Points à vérifier :

- **Tension inverse maximale (VR)** : Assurez-vous que la diode peut supporter la tension inverse maximale de votre circuit d'alimentation.
- **Courant nominal (IF)** : La diode doit pouvoir gérer le courant nominal du servomoteur.

4.3.4 Assemblage complet du bras manipulateur

Après avoir énuméré les différentes composantes du bras manipulateur, nous procéderons à leur assemblage afin d'obtenir un bras manipulateur complet, comme l'illustre la figure ci-dessous.



Figure 4-19 Assemblage complet du bras manipulateur



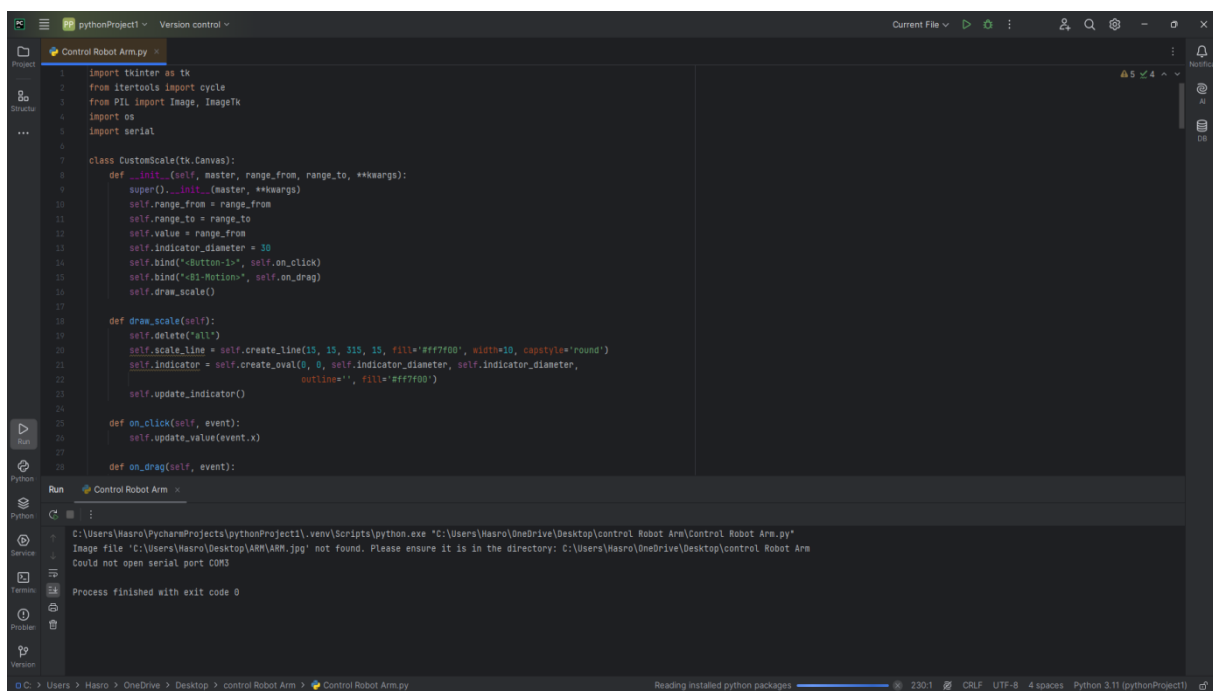
Figure 4-20 Les Différentes Vues du Bras

4.4 Contrôle du bras manipulateur :

Dans cette section, l'accent sera mis sur le contrôle du bras manipulateur, y compris les logiciels et les composants matériels utilisés pour garantir un contrôle précis des mouvements du bras. Les différentes méthodes de contrôle, telles que le contrôle manuel et le contrôle programmé à l'aide d'interfaces utilisateur, seront expliquées, ainsi que les approches utilisées pour assurer la stabilité et la sécurité des opérations. Les défis potentiels liés au contrôle du bras seront également abordés, ainsi que les solutions pour garantir une performance optimale.

4.4.1 Programmation d'applications de contrôle :

J'ai développé une application pour contrôler un bras robotique en utilisant le langage de programmation Python via l'environnement de développement PyCharm 2024.1.4. Cette application permet à l'utilisateur de contrôler les mouvements du bras avec une grande précision grâce à une interface utilisateur simple et efficace. L'application utilise des bibliothèques puissantes comme Tkinter pour l'interface utilisateur, ce qui facilite le contrôle et offre un retour instantané sur les mouvements effectués. L'application a été conçue pour être flexible et personnalisable, permettant ainsi de l'adapter à différents besoins et modes de fonctionnement



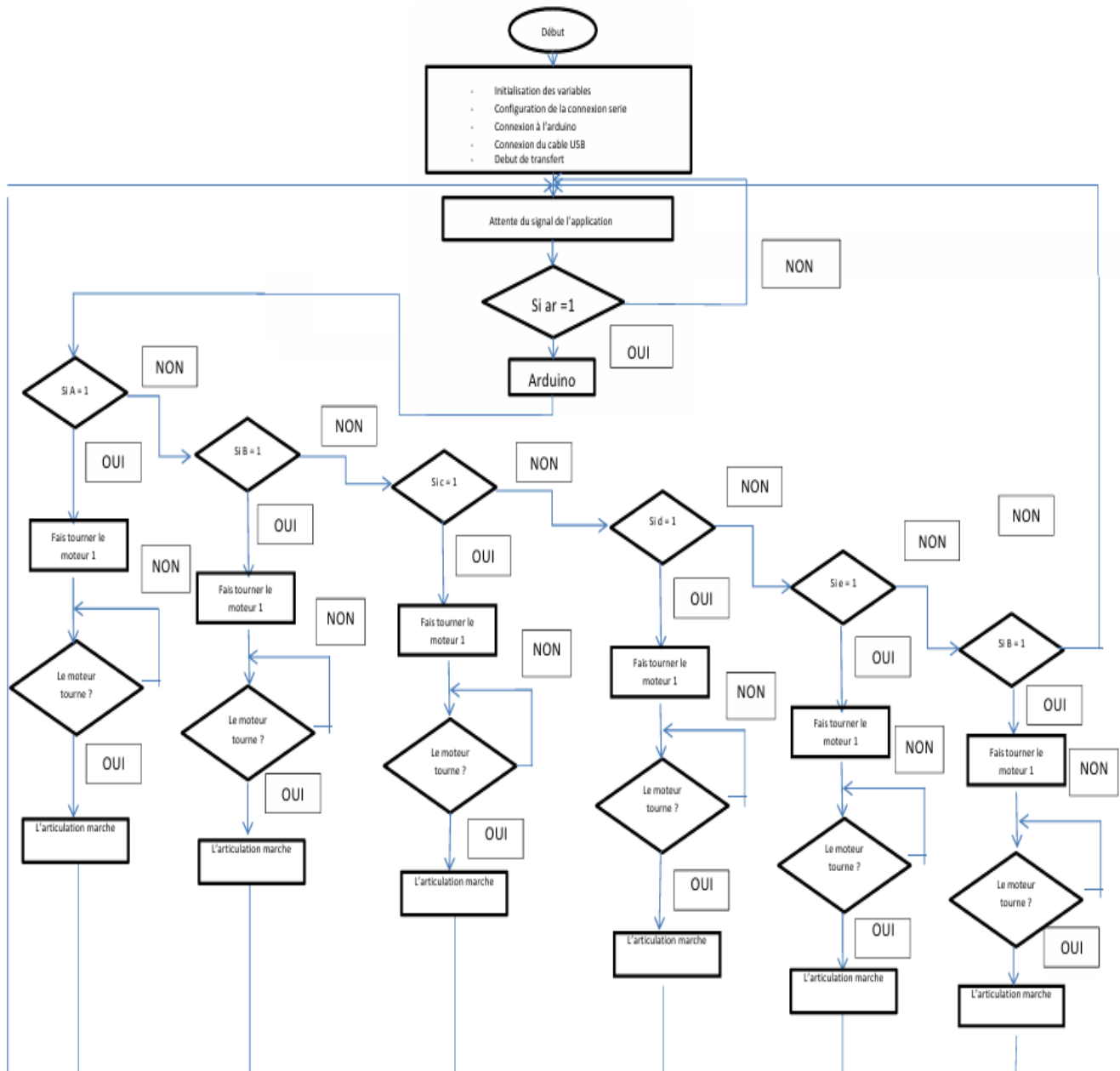
```
1 import tkinter as tk
2 from itertools import cycle
3 from PIL import Image, ImageTk
4 import os
5 import serial
6
7 class CustomScale(tk.Canvas):
8     def __init__(self, master, range_from, range_to, **kwargs):
9         super().__init__(master, **kwargs)
10        self.range_from = range_from
11        self.range_to = range_to
12        self.value = range_from
13        self.indicator_diameter = 10
14        self.bind("<Button-1>", self.on_click)
15        self.bind("<B1-Motion>", self.on_drag)
16        self.draw_scale()
17
18    def draw_scale(self):
19        self.delete('all')
20        self.scale_line = self.create_line(15, 15, 315, 15, fill='black', width=2, capstyle='round')
21        self.indicator = self.create_oval(0, 0, self.indicator_diameter, self.indicator_diameter,
22                                           fill='black', outline='black')
23        self.update_indicator()
24
25    def on_click(self, event):
26        self.update_value(event.x)
27
28    def on_drag(self, event):
```

Run Console Output:

```
C:\Users\Hassan\PycharmProjects\pythonProject1\.venv\Scripts\python.exe "C:\Users\Hassan\OneDrive\Desktop\control Robot Arm\Control Robot Arm.py"
Image file 'C:\Users\Hassan\Desktop\ARM\ARM.jpg' not found. Please ensure it is in the directory: C:\Users\Hassan\Desktop\control Robot Arm
Could not open serial port COM3
Process finished with exit code 0
```

Figure 4-21 Code de programmation d'applications en Python

Ce logigramme représente une explication du fonctionnement d'une application codée en Python, qui a pour but de faire fonctionner un bras robot manipulateur



4.4.2 Description de l'application :

- Connexion via USB :

L'application communique avec le bras robotique via une connexion USB, permettant ainsi de transmettre les commandes et de contrôler les servomoteurs avec précision et rapidité.

- 6 curseurs pour contrôler les servomoteurs :

L'application comprend 6 curseurs dédiés à chaque servomoteur. Chaque moteur peut tourner à des angles spécifiques selon les commandes entrées par l'utilisateur.

- **Bouton Exit :**

Le bouton de sortie permet à l'utilisateur de fermer l'application en toute sécurité après avoir fini de contrôler le bras robotique.

- **Curseur et champ d'affichage de l'angle de rotation :**

L'application comprend un curseur pour afficher l'angle de rotation actuel de chaque moteur, et l'utilisateur peut également entrer manuellement un angle de rotation spécifique pour l'appliquer au moteur.

- **Connected et Disconnected :**

L'application dispose d'un champ d'affichage qui montre l'état de la connexion avec le bras robotique, permettant à l'utilisateur de savoir si le robot est correctement connecté ou non.

- **Bouton Reset :**

L'application possède un bouton spécifique pour réinitialiser toutes les valeurs d'angle de rotation des moteurs à zéro, facilitant ainsi le retour rapide du bras à sa position par défaut.

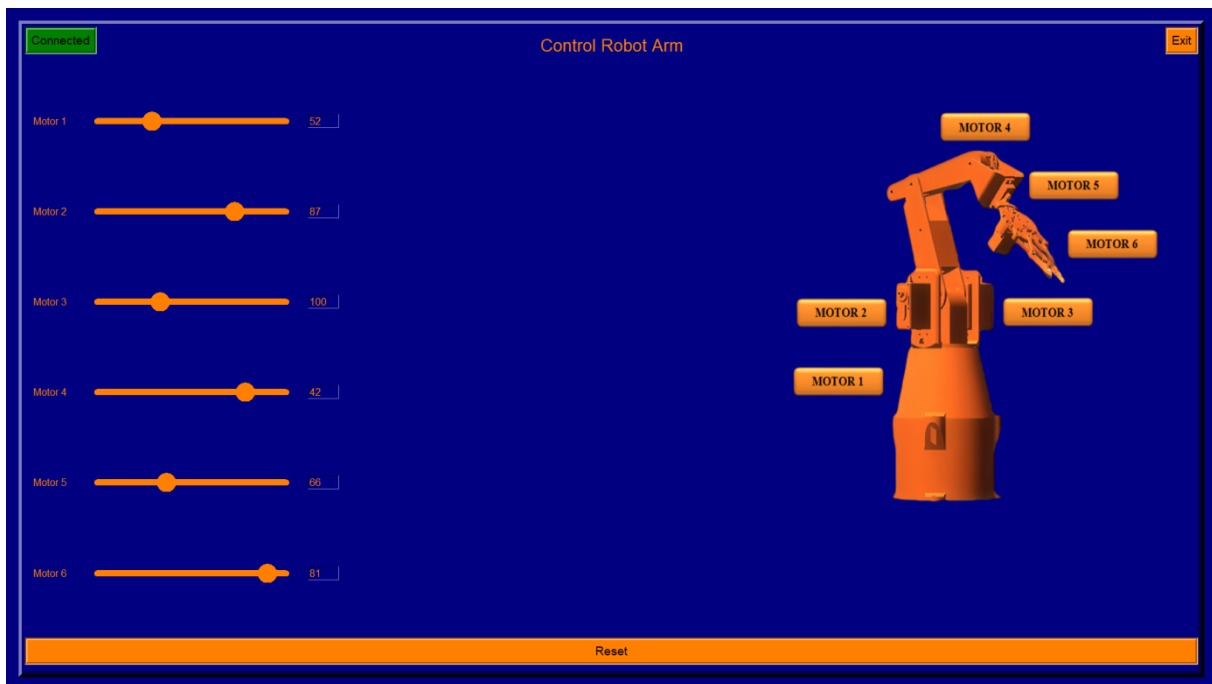
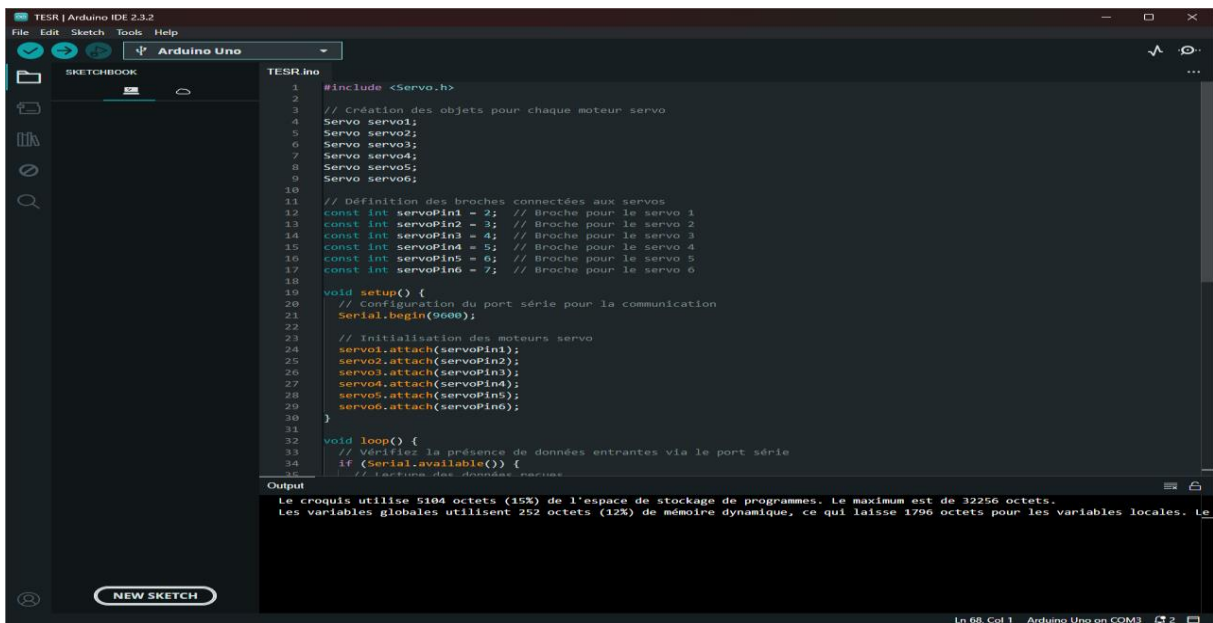


Figure 4-22 Interface finale de l'application

4.4.3 Programmation Arduino :

Nous avons programmé une carte Arduino pour contrôler les moteurs servo du bras robotique. Le processus a commencé par le raccordement des moteurs aux câbles appropriés et la configuration des circuits nécessaires sur la carte Arduino. Ensuite, on a écrit le code qui gère le mouvement des moteurs en fonction des commandes envoyées par l'application que on a développée. Nous avons utilisé la bibliothèque `Servo` d'Arduino pour faciliter le contrôle des moteurs, en définissant avec précision les angles de mouvement et les vitesses pour assurer un fonctionnement cohérent et efficace du bras. Après avoir terminé la programmation, on a effectué plusieurs tests pour m'assurer que les moteurs réagissaient correctement aux commandes et que le système fonctionnait de manière optimale et sans



```
1 #include <Servo.h>
2
3 // Création des objets pour chaque moteur servo
4 Servo servo1;
5 Servo servo2;
6 Servo servo3;
7 Servo servo4;
8 Servo servo5;
9 Servo servo6;
10
11 // Définition des broches connectées aux servos
12 const int servoPin1 = 2; // Broche pour le servo 1
13 const int servoPin2 = 3; // Broche pour le servo 2
14 const int servoPin3 = 4; // Broche pour le servo 3
15 const int servoPin4 = 5; // Broche pour le servo 4
16 const int servoPin5 = 6; // Broche pour le servo 5
17 const int servoPin6 = 7; // Broche pour le servo 6
18
19 void setup() {
20   // Configuration du port série pour la communication
21   Serial.begin(9600);
22
23   // Initialisation des moteurs servo
24   servo1.attach(servoPin1);
25   servo2.attach(servoPin2);
26   servo3.attach(servoPin3);
27   servo4.attach(servoPin4);
28   servo5.attach(servoPin5);
29   servo6.attach(servoPin6);
30 }
31
32 void loop() {
33   // Vérifier la présence de données entrantes via le port série
34   if (Serial.available()) {
35     // Lire les données entrantes
36     int data = Serial.read();
37     // Traitement des données entrantes
38     // ...
39   }
40 }
```

Output

Le croquis utilise 5104 octets (15%) de l'espace de stockage de programmes. Le maximum est de 32256 octets. Les variables globales utilisent 252 octets (12%) de mémoire dynamique, ce qui laisse 1796 octets pour les variables locales. Le

Figure 4-24 Arduino Code

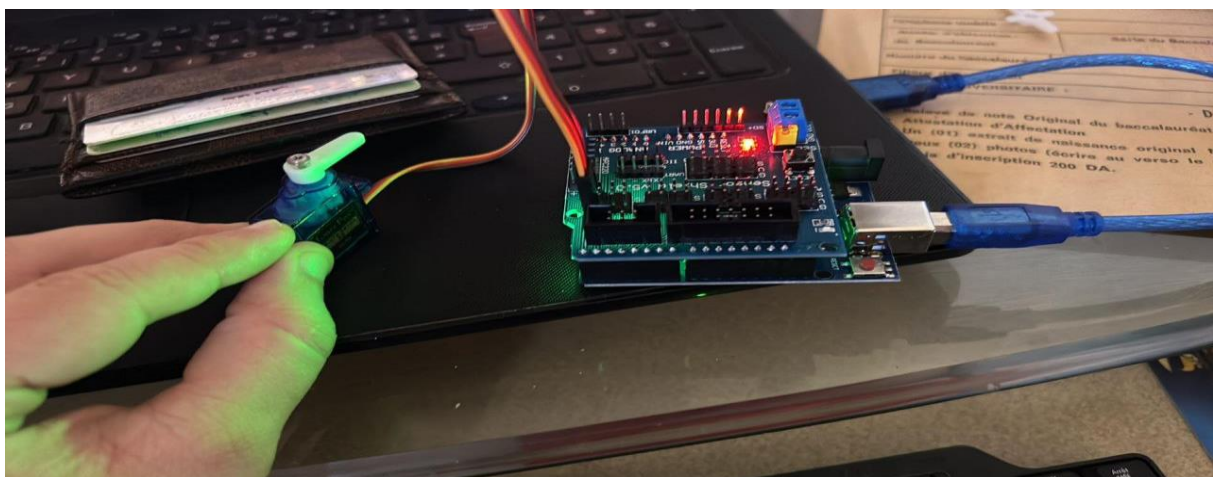
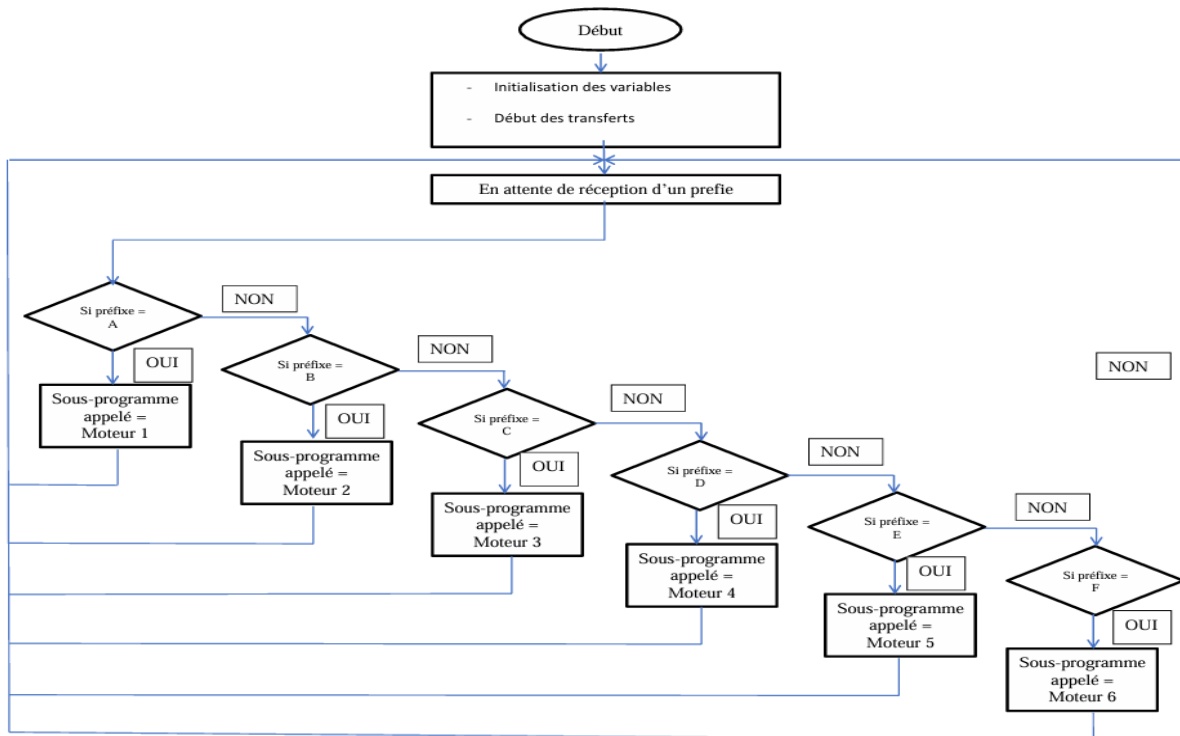


Figure 4-23 Le résultat de l'exécution.

Ce logigramme représente une explication pour le fonctionnement des servomoteurs grâce à l'Arduino



4.5 Travail du bras manipulateur

Nous avons programmé différentes positions de coordonnées (x, y, z) pour le bras robotique afin de réaliser des actions de préhension et de placement, comme le montrent les figures ci-dessous. A Choisir et placer



Figure 4-25 Bras robotique pour effectuer des actions

4.5.1 Planification des Tâches

Tâche de préhension et de placement

L'une des tâches de programmation les plus élémentaires est le "Pick and place" – un terme anglais signifiant "saisir et déposer". Il s'agit d'une opération de positionnement durant laquelle le robot doit atteindre plusieurs points définis dans son espace de travail. L'objectif est de déplacer des objets d'un endroit à un autre. La trajectoire entre ces deux points peut être arbitraire si aucun obstacle n'est présent.

Pour ce genre de tâche, l'outil utilisé est un dispositif de préhension d'objets (comme une pince ou un électroaimant, etc.). Pour la tâche que nous envisageons, nous allons supposer l'absence d'obstacles afin de simplifier le processus. Le schéma explicatif de la tâche est illustré par la figure 4.16.

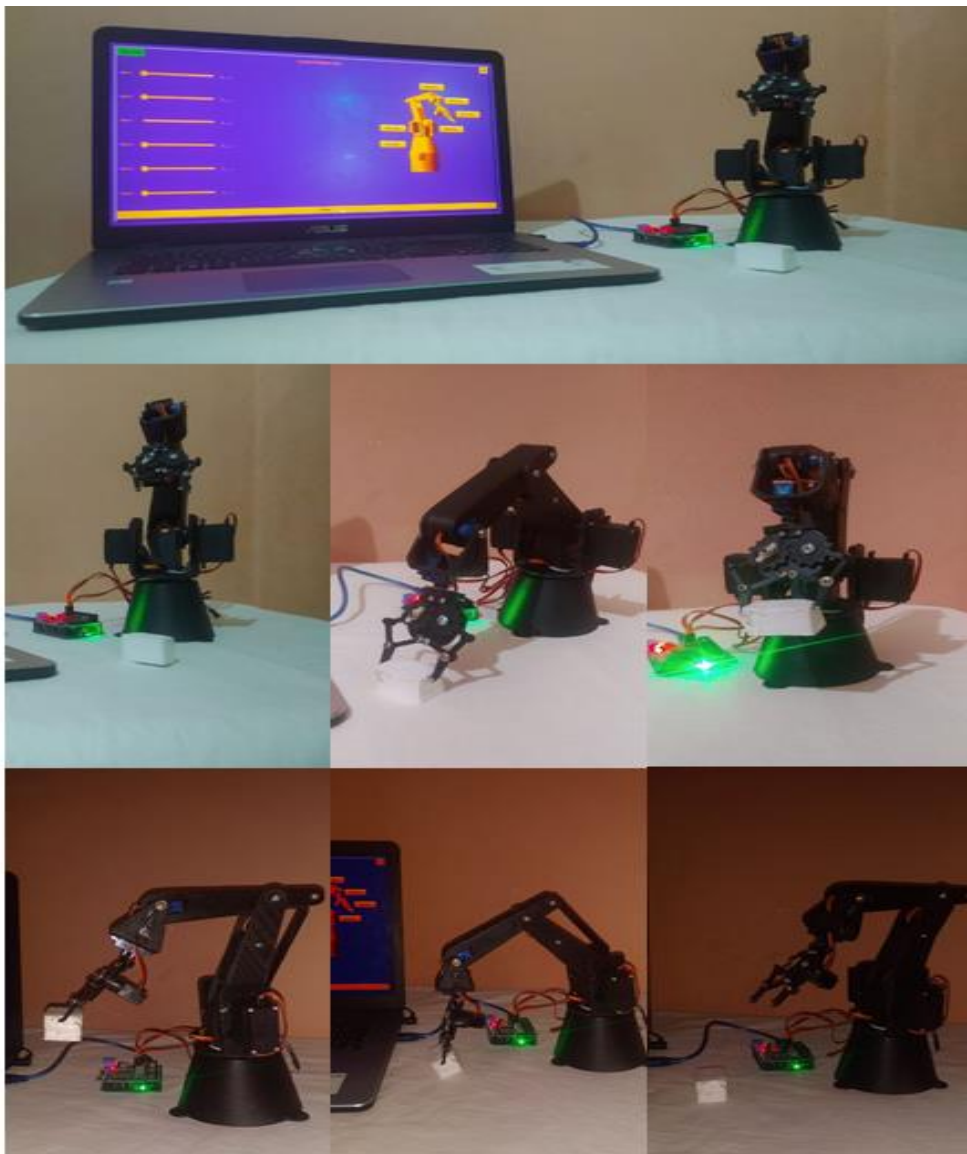


Figure 4-26 Quelques manipulations de Pick & Place spécifiques

Le robot doit déplacer un objet d'une position initiale sur un plan horizontal à une position finale située sur le même plan. Le déplacement du robot se déroulera en deux phases

- La première phase consiste à soulever l'objet du plan horizontal à une position intermédiaire.
- La seconde phase, où le robot devra atteindre sa position finale pour y déposer l'objet.

4.5.2 Contrôle via la saisie d'angle

Le contrôle des servomoteurs via l'entrée d'angles spécifiques dans des cases de saisie permet à l'utilisateur d'ajuster avec précision les mouvements du robot. L'utilisateur entre l'angle souhaité pour chaque servomoteur dans la case qui lui est dédiée, et cet angle est ensuite transmis au servomoteur qui se déplace en conséquence. Ce type de contrôle offre une grande facilité et flexibilité dans la gestion du robot, permettant ainsi d'exécuter des tâches complexes avec une grande précision, que ce soit dans des applications éducatives ou des projets robotiques avancés.

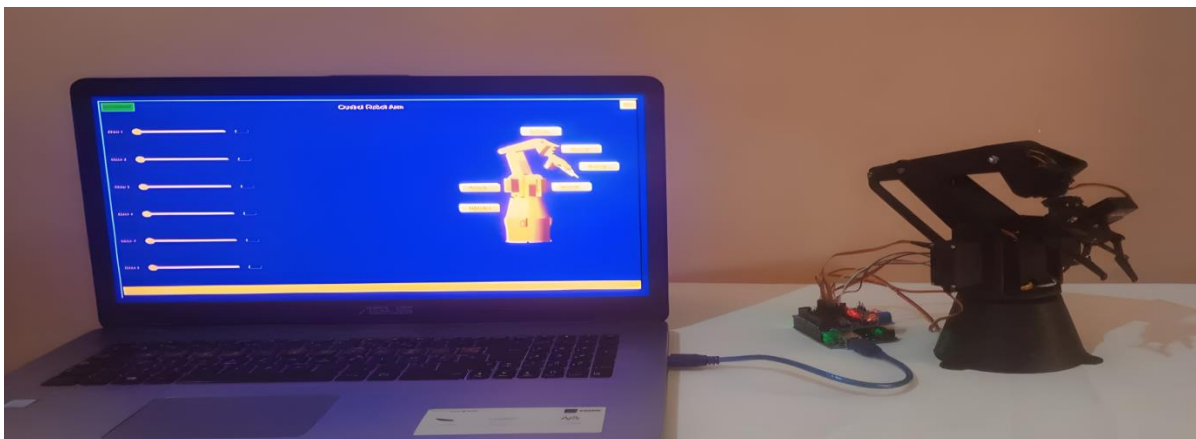


Figure 4-27 Position des mains au démarrage lorsque vous allumez

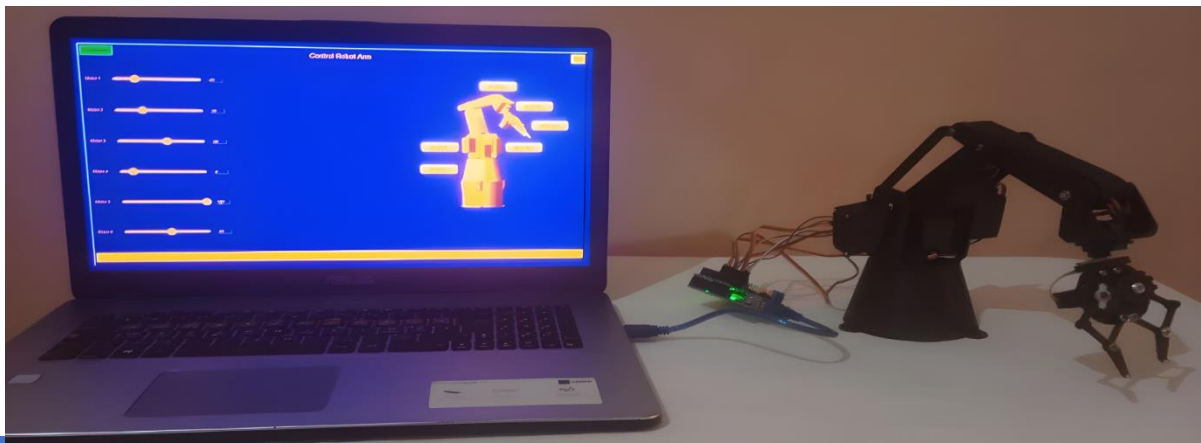


Figure 4-28 Position des mains après avoir entré les coordonnées

4.6 Conclusion

Cette phase de réalisation marque une étape clé dans l'évolution du projet, validant les approches adoptées et ouvrant la voie à des améliorations futures. Les résultats obtenus constituent une base solide pour envisager des applications industrielles ou des développements plus poussés dans le cadre de projets de recherche.

La phase d'application a été l'occasion de tester le fonctionnement du système dans des conditions réelles, démontrant la robustesse et la fiabilité du design. Les expérimentations réalisées ont non seulement confirmé la validité des choix techniques, mais ont également mis en lumière des pistes d'amélioration pour optimiser davantage l'efficacité et la précision du robot.

Conclusion générale

Grâce à ce projet de fin d'études, qui a été consacré à l'analyse et à la création d'un bras de robot manipulateur, nous avons pu nous immerger dans les défis et les avancées du domaine de la robotique. Au fil des étapes du projet, de la conception théorique à la mise en œuvre concrète, nous avons examiné les éléments mécaniques, électroniques et informatiques indispensables à la conception d'un robot manipulateur fonctionnel et performant.

Les simulations effectuées ont été essentielles pour améliorer les performances du bras robotique, nous donnant la possibilité de tester diverses configurations et d'apporter les ajustements requis avant la mise en production physique.

Ce projet ne s'est pas limité à une occasion d'appliquer les connaissances acquises pendant notre parcours, mais a également ouvert des perspectives d'innovation et de recherche à venir. L'importance et la pertinence de la robotique dans notre société sont confirmées par les différentes applications de notre travail, qu'elles soient dans l'industrie, la médecine ou les services.

En résumé, la mise en place de ce bras de robot manipulateur représente une étape importante dans notre carrière d'ingénieurs et constitue une fondation solide pour nos futures carrières. Les compétences et l'expérience que nous avons acquises lors de ce projet seront des atouts précieux pour nos futures aspirations professionnelles.

RÉFÉRENCES BIBLIOGRAPHIQUES

BIBLIOGRAPHIQUES

- [1] Bagad, V.S. *Mechatronics*. ISBN 9788184314908, 2008
- [2] Elanchezhian, C., & Sundar, G. Shanmuga. *Computer Aided Manufacturing*. First Edition 2005, Second Edition 2007.
- [3] Gangloff, Jacques. *Cours de Robotique*. Université de Strasbourg, 2010.
- [4] Ramos Melgar, Enrique, & Castro Diez, Ciriaco. *Arduino and Kinect Projects: Design, Build, Blow Their Minds*.
- [5] Affagard, Bruno, Géri-dan, Jean-Michel, & Lafargue, Jean-Noël. *Projets créatifs avec Arduino*. ISBN 978-2-7440 2617-1.
- [6] Banzi, Massimo. *Getting Started with Arduino*. ISBN 978-1-449 309879, First Edition.
- [7] Kelly, James Floyd, & Timmis, Harold. *Arduino Adventures: Escape from Gemini Station*.
- [8] Banzi, Massimo, & Shiloh, Michael. *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*. Maker Media, Inc., 2014.
- [9] Dutt, Anuj. *Arduino Based RADAR System*. United States: GRIN Verlag, Jun 25, 2014.
- [10] Benet, G., Albaladejo, J., Rodas, A., & Gil, P.J. "An intelligent ultrasonic sensor for ranging in an industrial distributed control system." In *Proceedings of the IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Malaga, Spain, May 1992, pp. 299–303.
- [11] Craig, John J. *Introduction to Robotics, Mechanics and Control*. Third Edition. Pearson Education, Inc., Upper Saddle River, New Jersey, 2005, 408p. ISBN 0-13-123629-6, 2001.
- [12] Khalil, W., & Dombre, E. *Modeling, Identification and Control of Robots*. Kogan Page Science, 2004, 483p. ISBN 1-903996-66-X.
- [13] Pasquier, Claude. *Mécanique*. Cours, Polytech Paris-Sud, 2012.
- [14] Mendil, B. *Cours de Robotique*. Cours, Université Abderrahmane Mira, Béjaia, 2017.
- [15] Boimond, Jean-Louis. *Cours de Robotique*. Université d'Angers, pages 10-88.
- [16] *Étude de cas : Bras manipulateur RPP*. Mémoire de master 2, Université Ibn-Khaldoun de Tiaret, 2015-2016, page 14, chapitre 2.
- [17] *Étude cinématique d'un bras manipulateur et simulation des mouvements sur logiciel SOLIDWORKS* par Bokreta, Manal, & Guelmame, Djamila
- [18] Paul R.C.P., *Robot manipulators: mathematics, programming and control*, MIT Press, Cambridge, 1981.

- [19] Raghavan M., Roth B., "Inverse kinematics of the general 6R manipulator and related linkages", Trans. of the ASME, J. of Mechanical Design, Vol. 115, 1990, p. 502-508.
- [20] Pieper D.L., "The kinematics of manipulators under computer control", Ph. D. Thesis, Stanford University, 1968.

Sites Web

- [21] https://www.persee.fr/doc/rei_0154-3229_1992_num_61_1_1436
- [22] <https://www.timetoast.com/timelines/histoire-de-la-robotique-42b54aaa-e621-4a97-9591-94d786eb7bf1>
- [23] <https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch4/history-histoire/5214792-fra.htm>
- [24] <https://www.staubli.com/fr/fr/robotics.html>
- [25] <https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/techniques/8692/8692-197-p46.pdf>
- [26] <https://www.evsint.com/fr/what-is-a-robotic-manipulator-a-guide/>
- [27] https://www.academia.edu/8239651/1_Dossier_ARDUINO
- [28] [Software | Arduino](#)
- [29] <http://ericaeromodelisme974.unblog.fr/2011/06/04/le-servomoteur/>
- [30] <http://howtomechatronics.com/projects/arduino-radar-project>
- [31] <https://www.techniques-ingenieur.fr/base-documentaire/archives-th12/archives-travail-des-metallurgistes-assembly-tiabt/archive-1/prototypage-rapide-bm7017/definition-du-prototypage-rapide-bm7017niv10002.html>
- [32] <https://www.futura-sciences.com/tech/definitions/informatique-cao-4453/>
- [33] <https://www.sculpteo.com/fr/glossaire/solidworks-definition-fr/>
- [34] <https://docer.pl/doc/x0xs01>
- [35] <https://sciences-indus-cpge.papanicola.info/MEC2-Chaines-cinematiques>

ANNEXE A :
Programme de l'application

Bibliothèque utilisée dans le programme :

```
import tkinter as tk
from itertools import cycle
from PIL import Image, ImageTk
import os
import serial
```

Bibliothèque tkinter (alias tk) :

Description: tkinter C'est la librairie classique pour concevoir des interfaces graphiques en Python. Son offre comprend des outils permettant de concevoir des fenêtres, des boutons, des étiquettes, des entrées, et bien d'autres encore.

Utilisation dans le code : L'utilisation de cette bibliothèque permet de concevoir la fenêtre principale de l'application, les divers widgets (tels que les boutons et les entrées), ainsi que de gérer les événements tels que les clics de souris et les raccourcis au clavier.

Bibliothèque itertools

Description : La fonction cycle de la bibliothèque itertools permet de générer un itérateur qui renvoie les éléments d'une séquence de manière cyclique.

Utilisation dans le code Son utilisation consiste à faire défiler les couleurs de fond du cadre principal de l'application, en alternant entre deux teintes

Bibliothèque PIL :

Description : PIL est une bibliothèque Python qui permet de traiter des images. Elle offre des options permettant d'accéder, de manipuler et de sauvegarder des images dans divers formats.

Utilisation dans le code: PIL est utilisée pour charger et afficher les images de fond et d'icône dans l'application. Plus précisément, Image et ImageTk sont utilisés pour ouvrir des images et les convertir en objets que tkinter peut afficher

Bibliothèque serial

Description : La bibliothèque serial (une composante de pyserial) offre la possibilité de dialoguer avec des périphériques à travers des ports série, tels que des microcontrôleurs.

Utilisation dans le code: Elle est utilisée pour établir une connexion série avec un Arduino, envoyer des commandes à l'Arduino et vérifier si la connexion est ouvert.

Bibliothèque OS

Description : Les fonctionnalités de la bibliothèque OS permettent d'interagir avec le système d'exploitation, telles que la gestion des fichiers et des répertoires.

Utilisation dans le code: Elle est utilisée pour vérifier l'existence des fichiers d'icônes et d'images avant de les charger. Cela aide à éviter les erreurs si les fichiers ne sont pas trouvés.

Détails sur le code

`reset_values()`, `update_entry()`, `update_scale()`, `check_connection()`, `update_connection_status()`, `on_close()`, `validate_entry()`, `increase_value()`, `decrease_value()`, `on_key_press()`, et `on_resize()`

Assurent la gestion de différents aspects de l'application, comme la réinitialisation des valeurs, la mise à jour des entrées, la vérification de la connexion série, ainsi que la gestion des événements de redimensionnement et de touches.

```
def reset_values():
    for entry, scale in zip(entries, scales):
        entry.delete(0, tk.END)
        entry.insert(0, '0')
        scale.value = 0
        scale.update_indicator()
```

def : Sa fonction permet de rassembler un ensemble d'instructions qui peuvent être exécutées en l'appelant par son nom. Une fonction peut être sollicitée et fournir des valeurs.

La boucle for

```
for entry, scale in zip(entries, scales):
    entry.delete(0, tk.END)
    entry.insert(0, '0')
    scale.value = 0
    scale.update_indicator()
```

Une boucle "for" On l'emploie pour faire des itérations sur une séquence (liste, tuple, dictionnaire, etc). Elle offre la possibilité de consulter chaque élément d'une séquence et de lancer du code pour chacun d'entre eux.

La boucle if

```
if ser and ser.is_open:
    connected = True
else:
    connected = False
```

Le code peut être exécuté en fonction d'une condition (qu'elle soit vraie ou fausse) grâce aux instructions conditionnelles. En Python, les termes `if`, `elif` (sinon si) et `else` (sinon si) sont utilisés.

```
root.after(5000, check_connection)
```

`root.after()` effectue régulièrement des fonctions. On utilise fréquemment cette méthode avec `tkinter` afin de gérer les événements de manière asynchrone (par exemple, vérifier la connexion toutes les 5 secondes).

```
root.bind('<KeyPress>', on_key_press)
```

`Bind()` établit une connexion entre une action et un événement (par exemple, une pression sur une touche du clavier ou un clic de souris).

Récapitulatif :

`def` : Définit une fonction, qui est un bloc de code réutilisable.

`for` : Crée une boucle pour parcourir une séquence (liste, tuple, etc.).

`if-elif-else` : Crée une structure conditionnelle pour exécuter du code en fonction d'une condition.

`root.after()` : Planifie l'exécution d'une fonction après un délai, généralement utilisé pour créer une boucle d'événements avec `tkinter`.

`bind()` : Associe un événement à une fonction.

Méthodes (`self.methode()`) : Appelle une fonction associée à un objet ou une classe.

ANNEXE B :

Programmes Arduino

Bibliothèque utilisée dans le programme :

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

Bibliothèque Servo : On utilise cette bibliothèque pour superviser des servomoteurs. Elle offre la possibilité de déplacer un servomoteur à un angle précis (de 0° à 180°).

Fonctionnalités principales :

Servo.attach(pin) : Associe un servomoteur à une broche de l'Arduino.

Servo.write(angle) : Déplace le servomoteur à un angle donné (compris entre 0 et 180 degrés).

Bibliothèque Math : C'est une bibliothèque standard de mathématiques en C++ utilisée en Arduino pour les opérations mathématiques avancées.

Explication des Commandes Utilisées :

Création d'objets Servo :

```
// Création des objets pour chaque moteur servo
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
```

Tu declares six objets de type Servo, chacun correspondant à un moteur servo que tu souhaites contrôler.

Définition des broches des servos :

```
// Définition des broches connectées aux servos
const int servoPin1 = 2;
const int servoPin2 = 3;
const int servoPin3 = 4;
const int servoPin4 = 7;
const int servoPin5 = 9;
const int servoPin6 = 10;
```

Ces constantes définissent les broches Arduino auxquelles chaque servomoteur est connecté.

Fonction setup()

```
void setup() {  
  // Configuration du port série pour la communication  
  Serial.begin(9600);  
  
  // Initialisation des moteurs servo  
  servo1.attach(servoPin1);  
  servo2.attach(servoPin2);  
  servo3.attach(servoPin3);  
  servo4.attach(servoPin4);  
  servo5.attach(servoPin5);  
  servo6.attach(servoPin6);  
}
```

`Serial.begin(9600)` : Initialise la communication série avec une vitesse de transmission de 9600 bauds (pour envoyer et recevoir des données via le port série).

`servoX.attach(servoPinX)` : Associe chaque servomoteur à une broche Arduino correspondante pour permettre de les contrôler.

Fonction loop() :

La fonction `loop()` tourne en permanence pendant l'exécution du programme.

Lecture des données via le port série :

```
void loop() {  
  // Vérifiez la présence de données entrantes via le port série  
  if (Serial.available()) {  
    // Lecture des données reçues  
    String input = Serial.readStringUntil('\n');  
  }  
}
```

`Serial.available()` : Vérifie si des données sont disponibles sur le port série.

`Serial.readStringUntil('\n')` : Lit les données reçues jusqu'au caractère de nouvelle ligne ('\n').

Traitement des données reçues :

```
// Analyse des données en numéro d'index et angle du moteur
int index = input.substring(0, input.indexOf(' ')).toInt();
int angle = input.substring(input.indexOf(' ') + 1).toInt();
```

`input.substring(0, input.indexOf(' '))` : Extrait la partie de la chaîne avant l'espace (correspond au numéro du moteur).

`input.substring(input.indexOf(' ') + 1)` : Extrait la partie de la chaîne après l'espace (correspond à l'angle du moteur).

`toInt()` : Convertit la chaîne en un entier.

Transformation mathématique de l'angle :

```
float transformedAngle = cos(radians(angle)) * 90;
```

`radians(angle)` : Convertit l'angle en degrés vers des radians.

`cos(radians(angle))` : Calcule le cosinus de l'angle en radians.

90 : Multiplie le cosinus par 90 pour obtenir un angle transformé.

Contrôle des limites de l'angle :

```
transformedAngle = constrain(transformedAngle, 0, 180);
```

`constrain(x, min, max)` : Restreint la valeur de `x` pour qu'elle soit comprise entre **min** et **max**.

Ici, cela assure que l'angle est compris entre 0° et 180°.

Commande des servos :

```
switch (index) {  
    case 1:  
        servo1.write(transformedAngle);  
        break;  
    ...  
}
```

`switch(index)` : Selon l'index reçu, cette structure contrôle quel servomoteur sera déplacé à l'angle transformé.

`servoX.write(transformedAngle)` : Déplace le servomoteur correspondant à **transformedAngle**.

Affichage sur la console série :

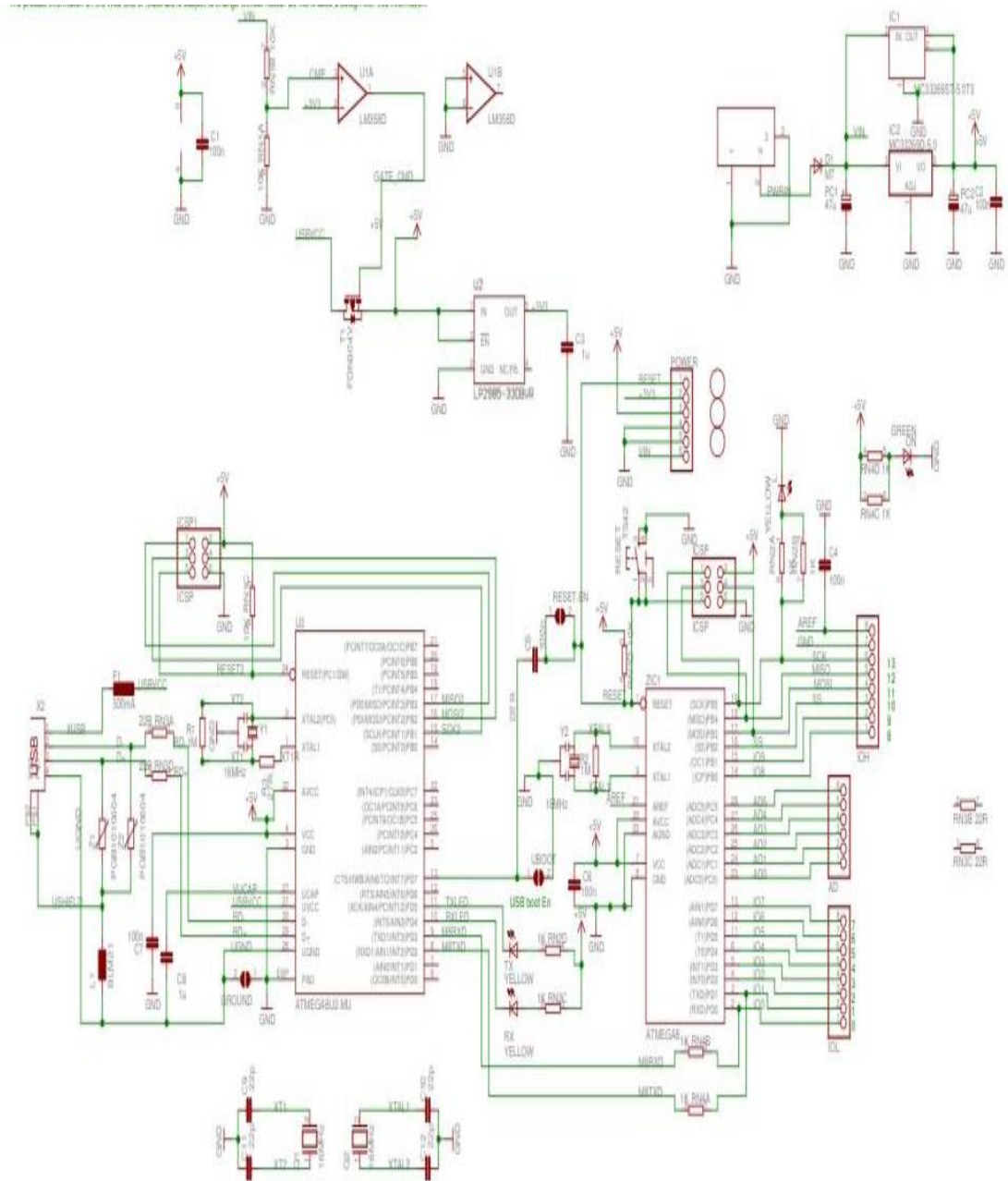
```
Serial.print("Moteur ");  
Serial.print(index);  
Serial.print(": Angle transformé = ");  
Serial.println(transformedAngle);
```

`Serial.print()` et `Serial.println()` : Ces fonctions envoient des messages au moniteur série pour afficher des informations (ici l'index du moteur et l'angle transformé).

ANNEXE C :

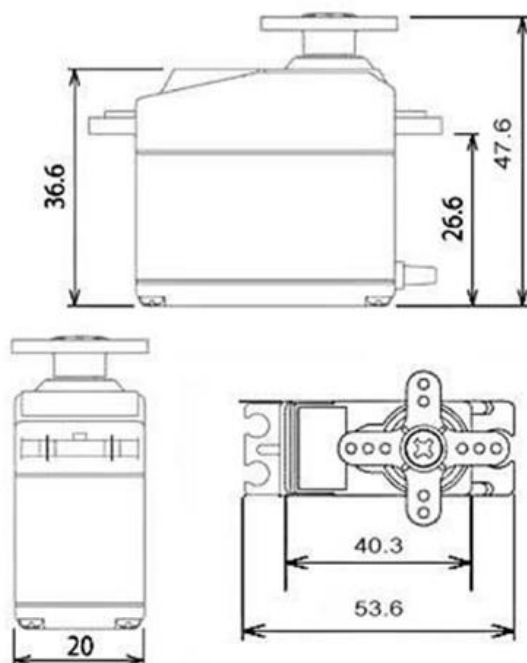
Les schémas

Arduino™ UNO



Les dimensions d'un Servomoteur

Servomoteur MG996R



Servomoteur Sg90

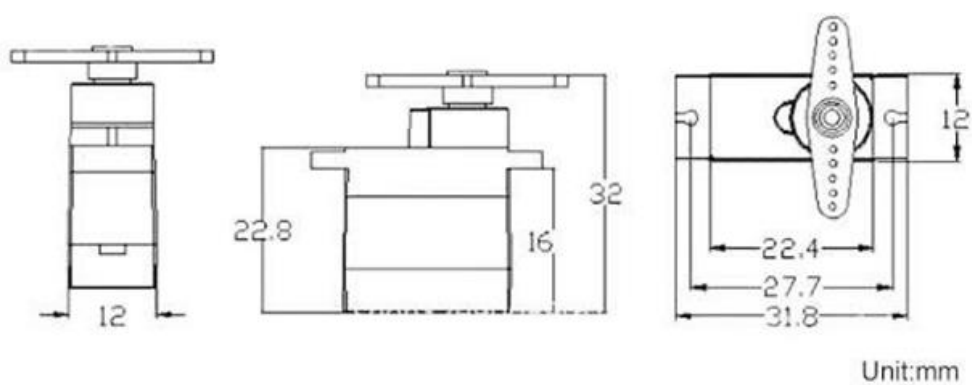
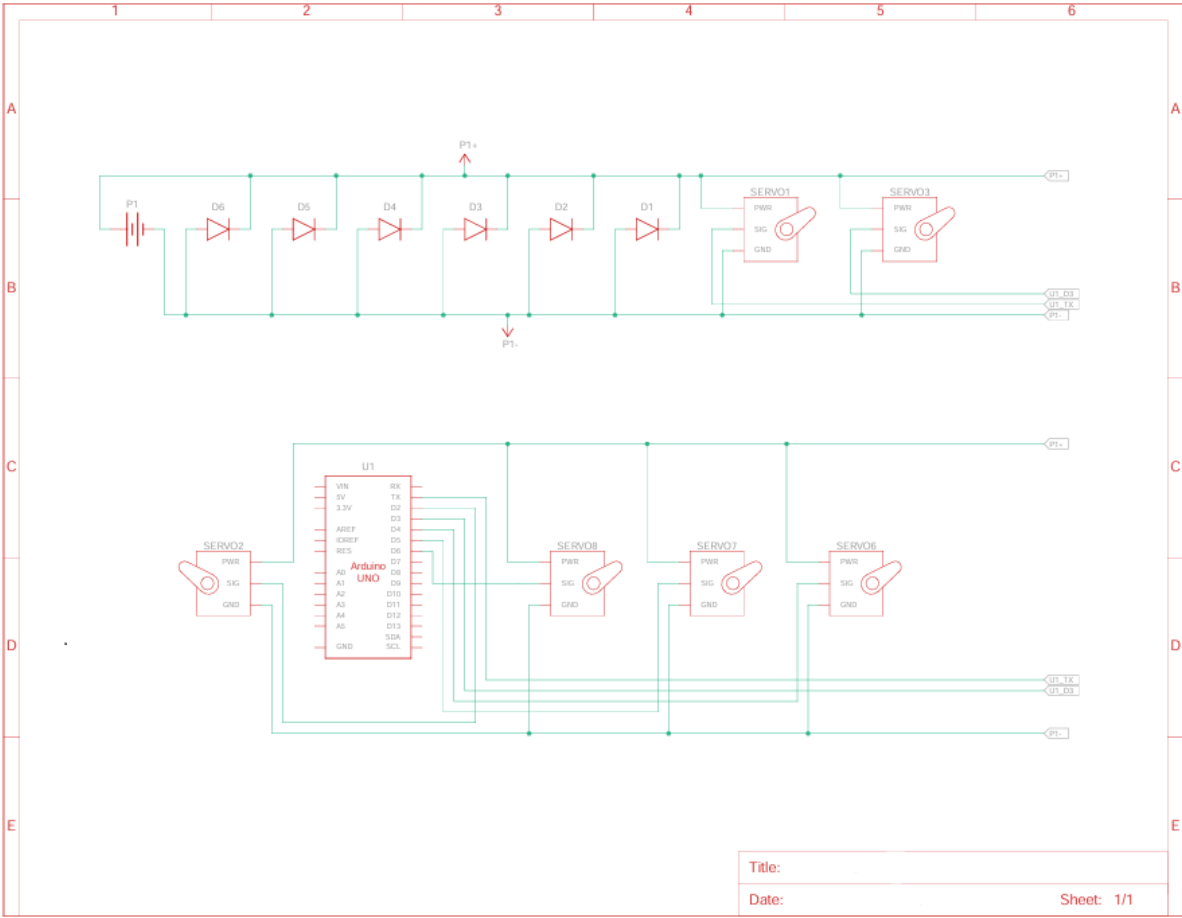
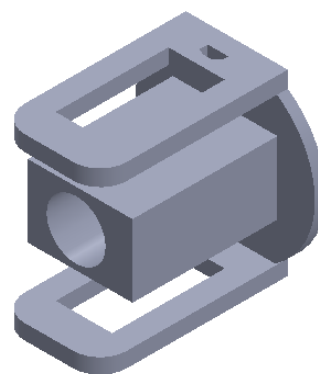
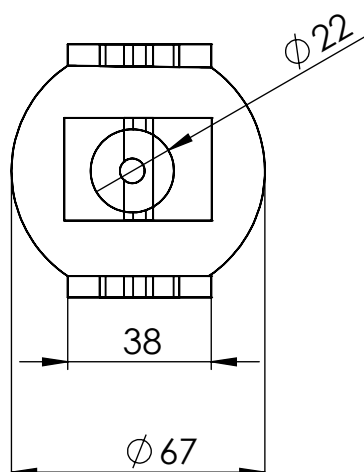
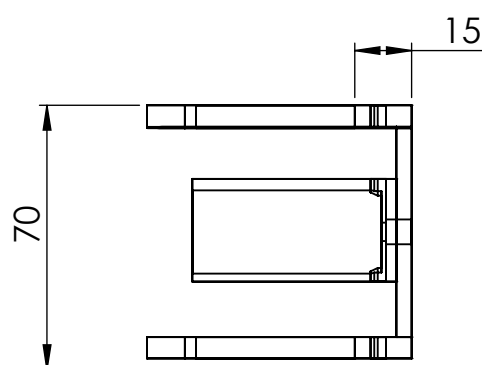


Schéma de la carte de puissance

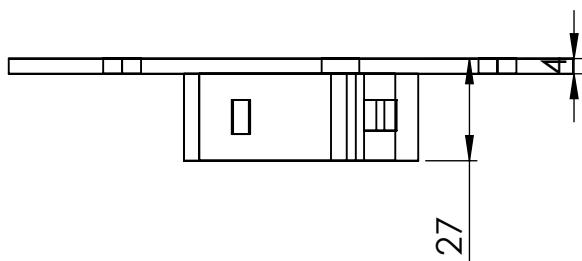
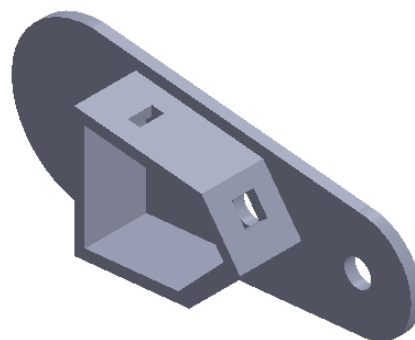
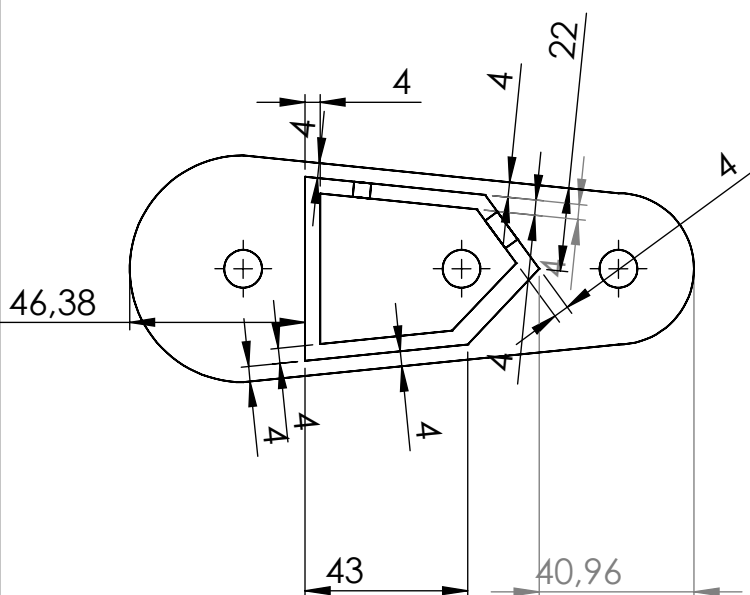
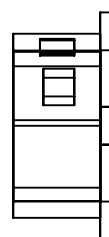


ANNEXE D:

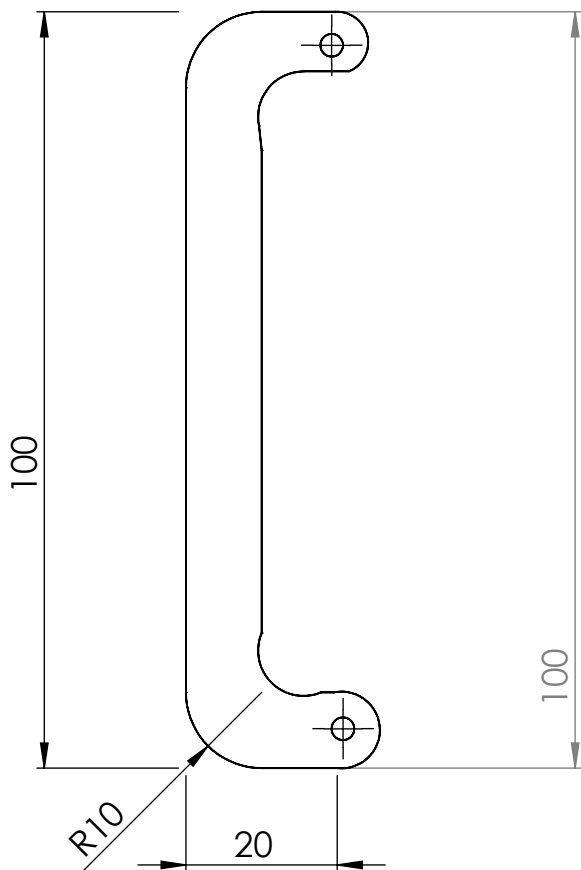
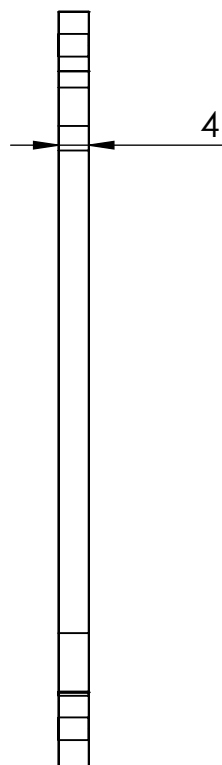
Mises en plan des pièces du bras manipulateur



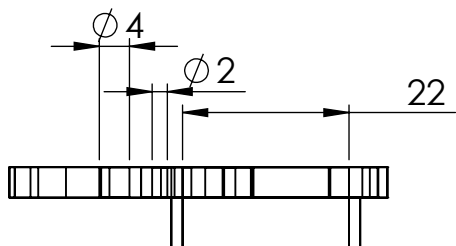
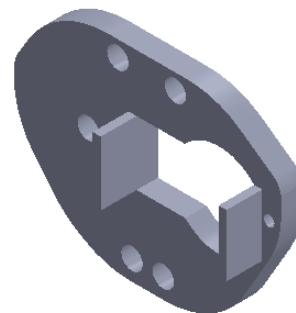
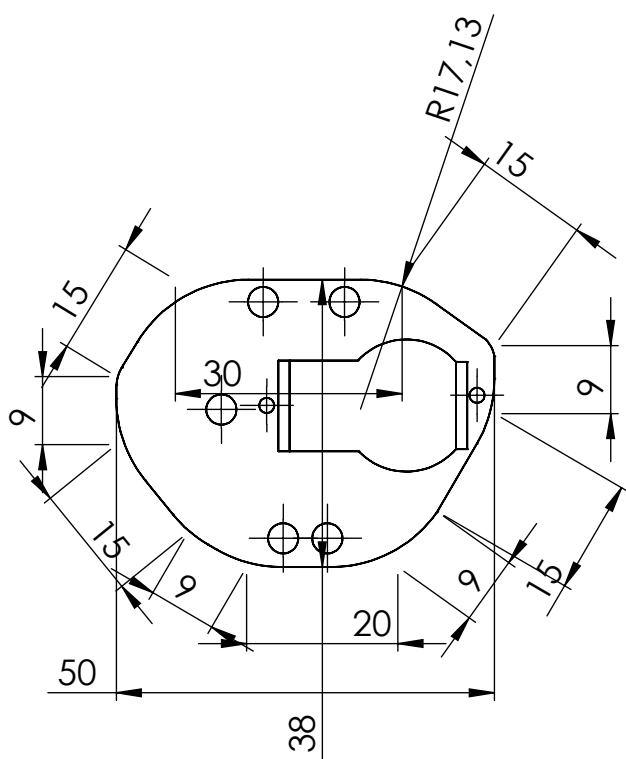
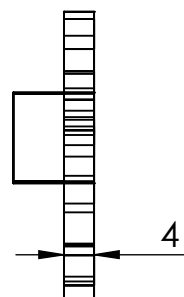
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:				FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION			
	NOM	SIGNATURE	DATE				TITRE:						
AUTEUR													
VERIF.													
APPR.													
FAB.													
QUAL.				MATERIAU:			No. DE PLAN		Pièce 01		A4		
				MASSE:			ECHELLE:1:2		FEUILLE 1 SUR 1				



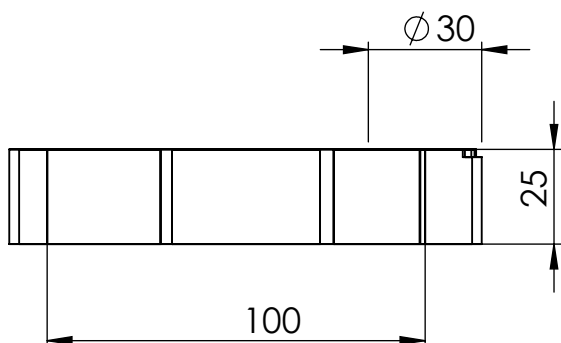
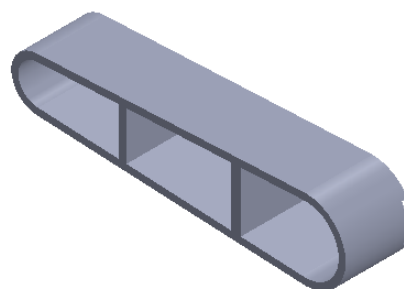
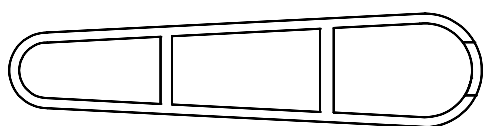
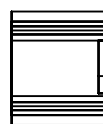
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
								Pièce 02	
								A4	
						MASSE:		ECHELLE:1:2	
								FEUILLE 1 SUR 1	



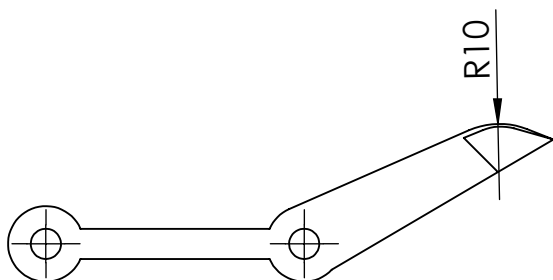
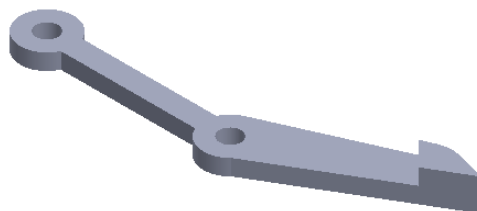
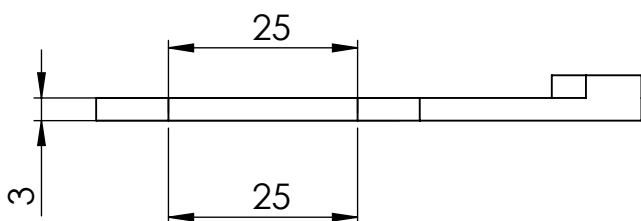
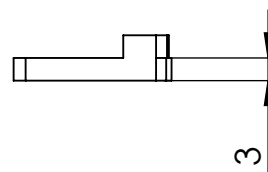
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAU:		No. DE PLAN		Pièce 03	
				MASSE:		ECHELLE:1:1		FEUILLE 1 SUR 1	



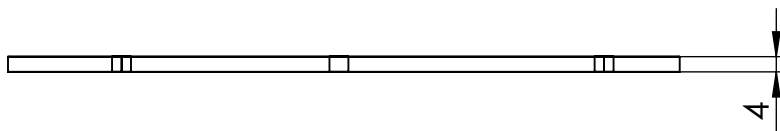
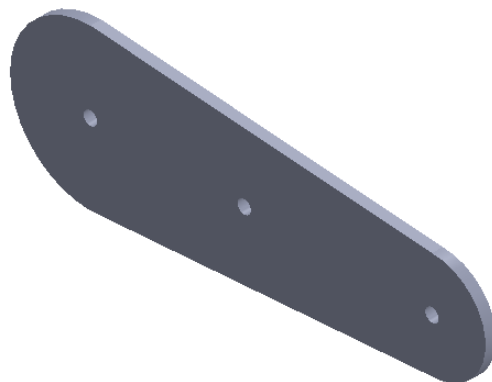
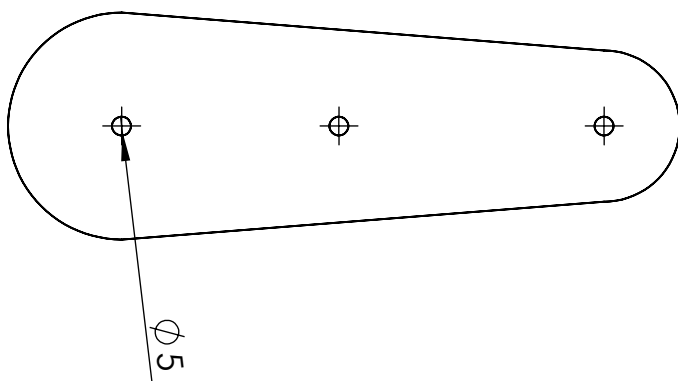
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
								Pièce 04	
								A4	
						MASSE:		ECHELLE:1:1	
								FEUILLE 1 SUR 1	



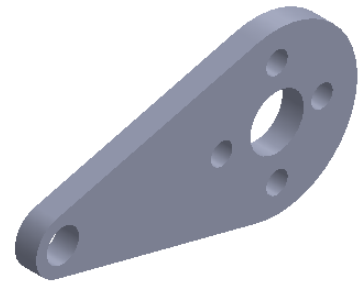
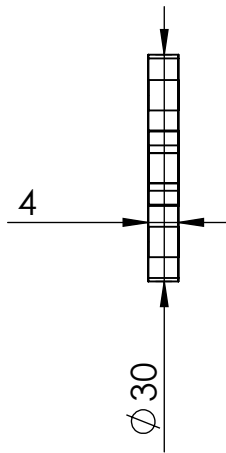
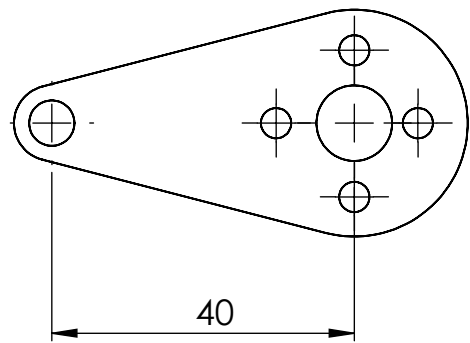
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
	NOM	SIGNATURE	DATE			TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAU:		No. DE PLAN		Pièce 05	
								A4	
				MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1	



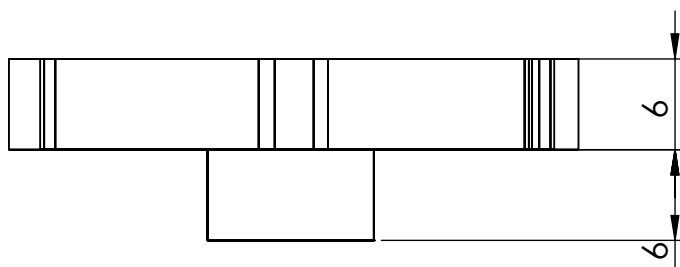
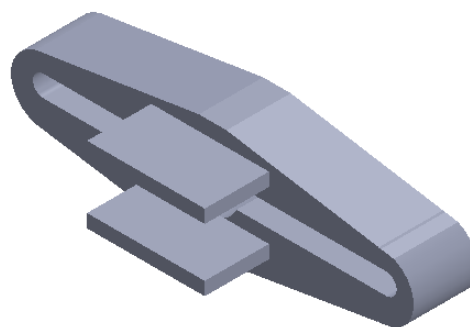
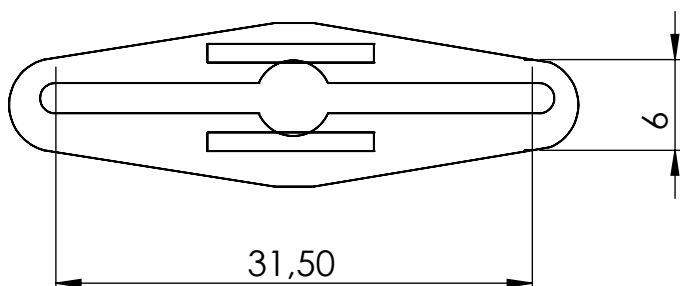
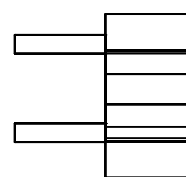
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:				FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE				TITRE:			
AUTEUR											
VERIF.											
APPR.											
FAB.											
QUAL.						MATERIAU:		No. DE PLAN		Pièce 06	
										A4	
						MASSE:		ECHELLE:1:1		FEUILLE 1 SUR 1	



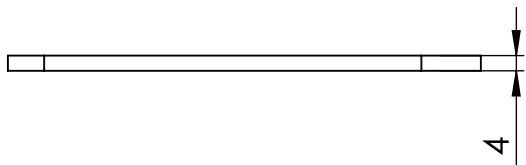
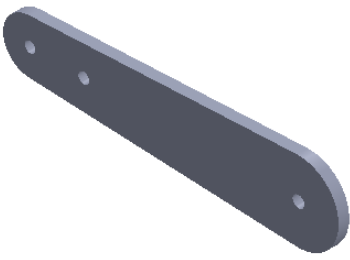
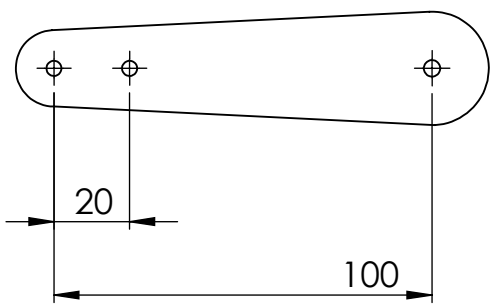
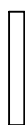
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:				CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
	NOM	SIGNATURE	DATE				TITRE:				
AUTEUR											
VERIF.											
APPR.											
FAB.											
QUAL.					MATERIAU:		No. DE PLAN		Pièce 07		A4
					MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1		



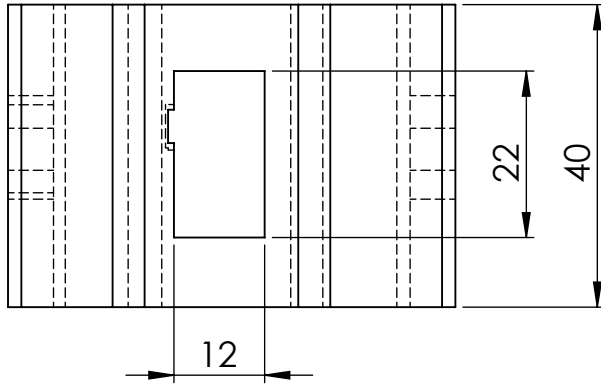
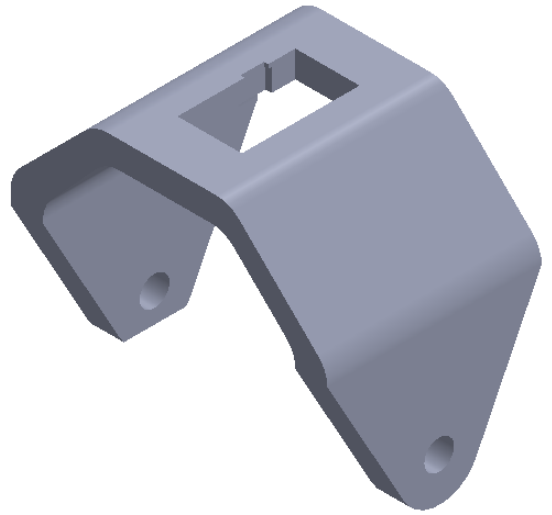
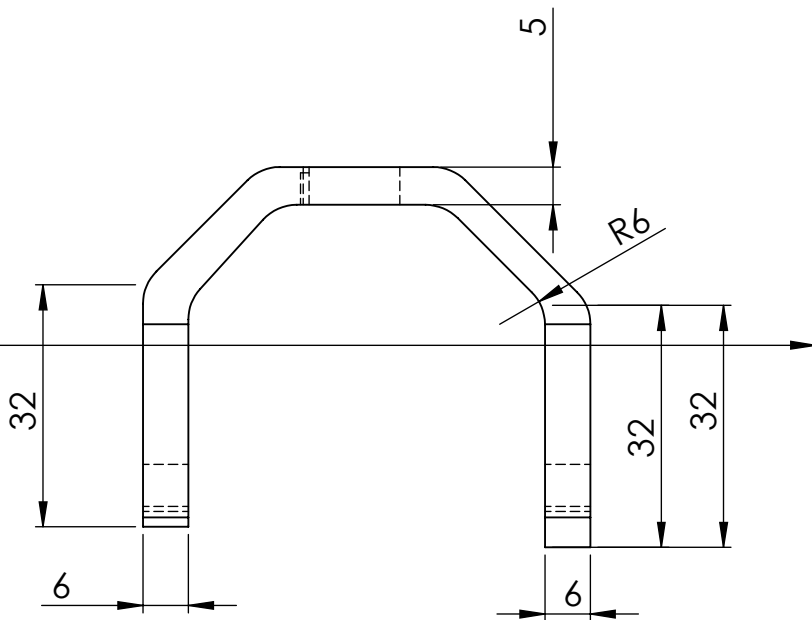
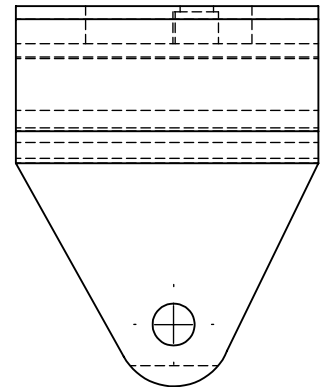
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
								Pièce 08	
								A4	
						MASSE:		ECHELLE:1:1	
								FEUILLE 1 SUR 1	



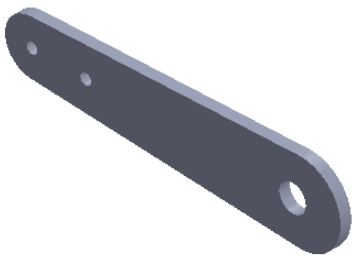
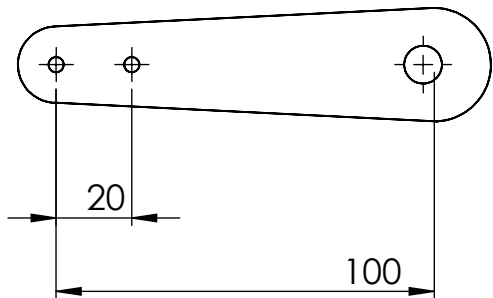
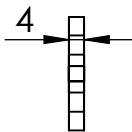
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
								Pièce 09	
								A4	
						MASSE:		ECHELLE:2:1	
								FEUILLE 1 SUR 1	



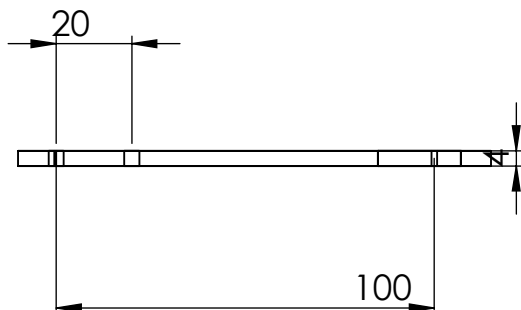
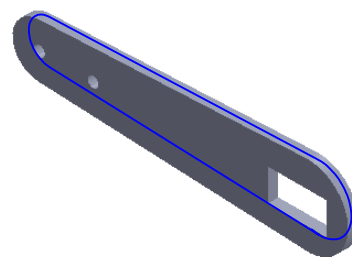
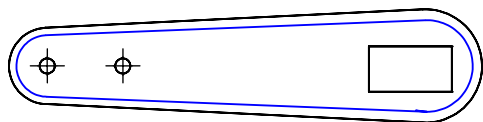
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:				CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
	NOM	SIGNATURE	DATE				TITRE:				
AUTEUR											
VERIF.											
APPR.											
FAB.											
QUAL.				MATERIAU:			No. DE PLAN		Pièce 10		A4
				MASSE:			ECHELLE:1:2		FEUILLE 1 SUR 1		



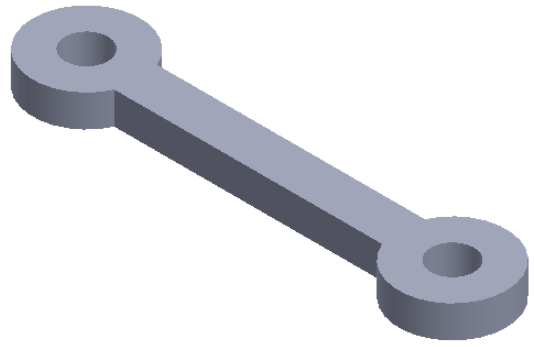
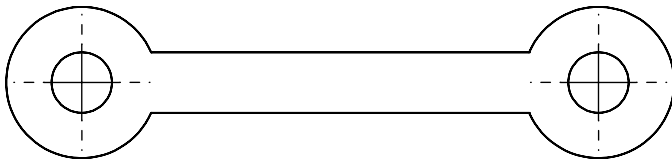
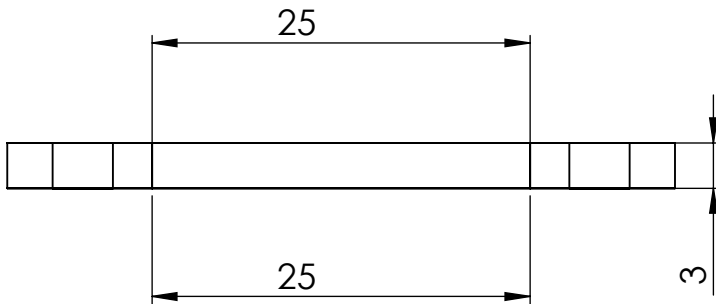
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAU:		No. DE PLAN		Pièce11	
								A4	
				MASSE:		ECHELLE:1:1		FEUILLE 1 SUR 1	



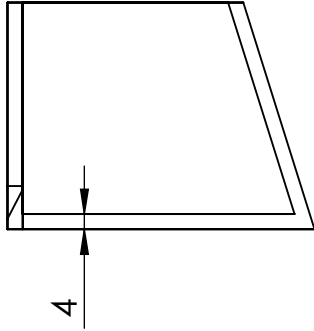
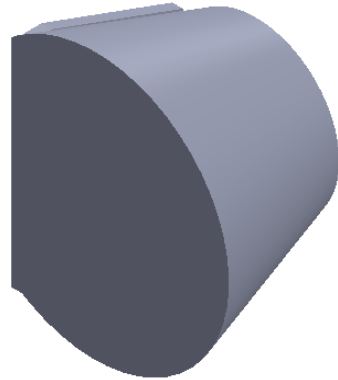
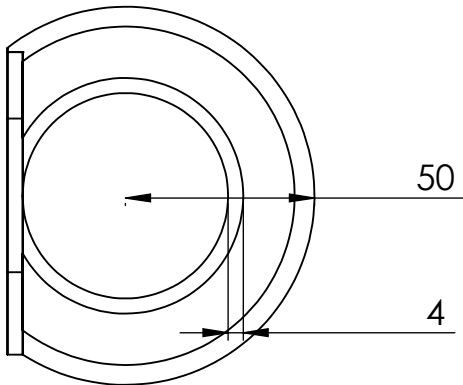
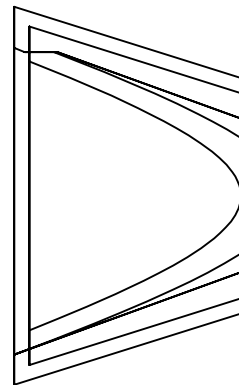
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
								Pièce 12	
								A4	
						MASSE:		ECHELLE:1:2	
								FEUILLE 1 SUR 1	



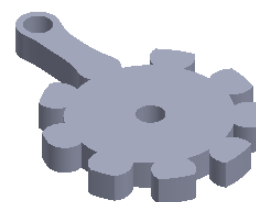
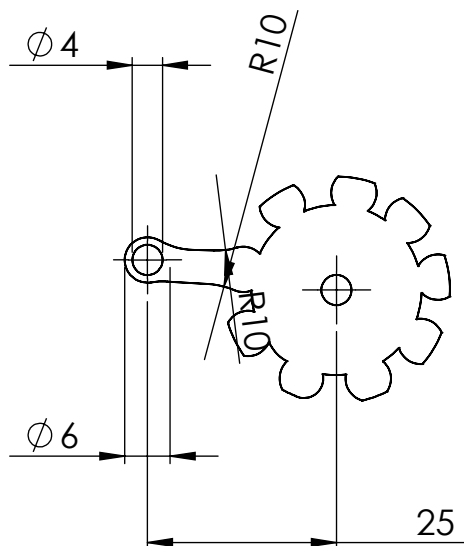
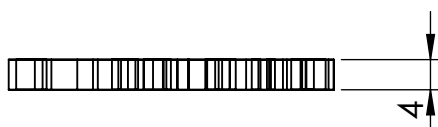
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:				CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE				TITRE:			
AUTEUR											
VERIF.											
APPR.											
FAB.											
QUAL.						MATERIAU:		No. DE PLAN		Pièce 13	
						MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1	



SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:		No. DE PLAN	
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						Pièce 14		A4	
				MASSE:		ECHELLE:2:1		FEUILLE 1 SUR 1	



SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAU:		No. DE PLAN		Pièce 15	
				MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1	



SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAU:		No. DE PLAN	
						MASSE:		ECHELLE:1:1	
								FEUILLE 1 SUR 1	

Pièce 16

A4