

Remerciements

Nous remercions Allah qui nous a donné la patience et le courage durant ces années d'études.

Et nous tenons exprimer notre gratitude et nos remerciements pour toutes les personnes qui ont contribué à la réalisation de ce projet de fin d'étude.

Nous présentons nos sincères remerciements à tous nos enseignants.

Nous tenons à remercier notre encadreur Mr : BENSATTALAH Aissa.

Et enfin, nos sincères remerciements pour les membres de jury d'avoir accepté d'examiner et d'évaluer notre travail

Dédicace

Avant tout je remercie Allah qui nous aidés à élaborer ce modeste travail que je dédié :

A ma tendre mère

Spécialement à Ma chère mère qu'elle m'a donnés la patience et le courage durant toutes ces longues années.

A mon père

Particulièrement à mon père qui mérite le grand respect, qu'il trouve ici l'expression de mon affection et une récompense des sacrifices consentis pour moi.

A mes parents

A tous ma petite famille et ainsi que la grande de loin et au pré.

A mes chères amies

A tous mes chères amies et ceux que j'aime et qui me sont très chère.

Aux bougies qui brulent pour éclairer les autres.

A tous ceux qui m'ont enseigné des personnages.

A tous ceux qui ont contribué à la libre diffusion des connaissances, consacré ce modeste travail.

En fin je souhaite que dieu nous dirige vers la réussite dans toute la vie.

sommaire

Remerciements.....	1
Dédicace.....	2
Résumé.....	7
Abstract.....	9
Introduction générale.....	10
Chapitre I: Robot industriel	
I. Introduction.....	11
I.1. Le robot industriel.....	11
I.1.1. Historique.....	12
I.1.2. Domaine d'application des robots industriels.....	12
I.1.3. Constituants mécaniques du robot.....	16
I.1.3.1. La Structure Mécanique Articulée.....	16
I.1.3.2. Types de chaînes de la SMA.....	17
I.1.3.3. L'effecteur.....	18
I.1.3.3.1 L'effecteur dans les robots manipulateurs.....	18
I.1.3.3.1.1 Classification des effecteurs.....	18
I.1.3.3.1.2.Les Pinces.....	19
I.2. Les modèles mathématiques des robots en générale.....	22
I.2.1. Degré de libertés.....	22
I.2.2. Translation et rotation.....	23
I.2.2.1. Matrices de rotation.....	24
I.2.2.2. Matrice de transformation homogène.....	25
I.2.3. Modèle géométrique direct.....	26
I.2.3.1. Calcul du MGD.....	27
I.2.3.2. Le modèle géométrique direct et inverse.....	28
I.2.3.4. Calcul de la matrice finale.....	29
I.2.4. Inversion du modèle géométrique.....	30
I.2.4.1. Exemples (simples).....	30
I.2.3.3. La télé-opération et la télé-robotique.....	Erreur ! Signet non défini.

I.4.	Conclusion	38
<i>Chapitre II: Les Composant électronique</i>		
II.	Introduction.....	39
II.1.	La carte d'Arduino.....	39
II.1.1.	Présentation d'Arduino	39
II.1.2.	Les gammes de la carte Arduino	39
II.1.3.	Pourquoi Arduino UNO	44
II.1.4.	La constitution de la carte Arduino UNO	44
II.1.4.1.	Partie matérielle.....	45
II.1.4.1.1.	Le Microcontrôleur ATmega328	45
II.1.4.1.2.	Les sources de l'alimentation de la carte	47
II.1.4.1.3.	Les entrées et sorties.....	48
II.1.4.1.4.	Les ports de communications	49
II.2.	Les moteurs	51
II.2.1.	Les moteurs électriques.....	51
II.2.2.	Les types des moteurs électriques	52
II.2.2.1	Moteurs à courant continu DC	52
II.2.2.2.	Moteurs à courant alternatif AC	52
II.2.2.3.	Moteurs pas à pas	53
II.2.2.4.	Servomoteurs.....	54
II.3.	Conclusion	57
<i>Chapitre III: Modélisation et Réalisation</i>		
III.	Introduction.....	58
III.1.	La description du robot	58
III.2.	Les modèles mathématiques de robot	59
III.2.1.	Modèle géométrique direct de robot	59
III.2.2.	Modèle géométrique inverse de robot.....	61
III.3.	Les outils de développement.....	64
III.3.1.	L'environnement de la programmation.....	64
III.3.2.	Structure générale du programme (IDE Arduino)	64
III.3.2.1.	Injection du programme	65

III.3.2.2. Description du (IDE).....	66
III.3.2.3. Commentaires	66
III.3.2.4. Définition des variables	67
III.3.2.6. Programmation des interactions void loop.....	67
III.3.3. Les étapes de téléchargement du programme	68
III.4. La réalisation pratique d'un robot industriel contrôlé par ordinateur	69
III.4.1. Espace de travail	70
III.4.2. Mode de travail	72
III.4.3. Description du programme.....	74
III.5. La télé opération de robot	79
III.6. Conclusion	81
Conclusion générale.....	82



Résumé

L'Algérie a lancé plusieurs zones industrielles dans tout le pays comme la fabrication de véhicules (l'usine de fabrication de véhicules de marque Mercedes Benz d'Ain Bouchekif-Tiaret), le soudage pour l'industrie pétrolière et gazière ...etc. Cependant le grand problème des entreprises algériennes est basé sur les être humain que sur les robots industriels.

Un robot industriel est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains. Actuellement le monde industriel ne peut pas avancer sans l'utilisation de la robotique industrielle. Il existe plusieurs domaines d'applications des robots industriels par exemple : les robots de soudure et de peinture, principalement utilisés dans l'automobile ; les robots d'assemblage.

L'objectif de ce travail est la conception et la réalisation d'un robot industriel destiné principalement à la réalisation des tâches bien définies tel que le soudage puis nous essayons de le commander à distance pour faire la télé-opération.

Mots –clé : Modèle mathématique de robot, modèle géométrique directe, modèle géométrique indirecte, télé-opération.

المخلص

أطلقت الجزائر عدة مناطق صناعية في جميع الوطن كتركيب السيارات (مصنع تركيب السيارات بعين بوشقيف ولاية تيارت لعلامة مرسيدس بنز) كالتلحيم من أجل الصناعة البترولية و الغازية... الخ

في هذا الحال المشكل الاعظم للمؤسسات الجزائرية يركز على الانسان كما يركز على الروبوتات الصناعية.

الروبوت الصناعي هو وحدة ميكاترونية (مزيغ ميكانيكي إلكتروني ومعلوماتي) الذي يقوم بتنفيذ المهام التي عادة ماتكون خطرة متكررة أو مستحيلة بالنسبة للإنسان. حاليا وفي هذه الاونة العالم الصناعي لايمكن أن يحقق أي تقدم بدون إستعمال تكنولوجيا الروبوت الصناعي . كما يوجد عدة مجالات لاستخدام هذه التقنية مثال: روبوتات التلحيم و الطلاء، وتستخدم بشكل رئيسي في صناعة السيارات؛ روبوتات التجميع.

الهدف من هذا العمل هو تصميم وإنجاز روبوت صناعي مكلف أساسيا بإنجاز مهام معرفة بشكل دقيق مسبقا كالتلحيم ثم نقوم بالتحكم به من مسافة للقيام بالعملية عن بعد.

الكلمات المفتاحية:النموذج الرياضي للروبوت،النموذج الجيومتري المباشر،النموذج الجيومتري الغير المباشر،القيام بالعملية عن بعد.

Abstract

Algeria has launched several industrial zones throughout the country such as the installation of cars (the installation of cars in Ain Bouchif

The state of Tiaret for the Mercedes-Benz brand) such as welding for the petroleum and gas industry ... etc

In this case, the great problem of Algerian institutions focuses on the human and focuses on industrial robots.

An industrial robot is a micronatonic unit that performs tasks that are usually dangerous, repetitive or impossible for humans. Currently, the industrial world can not achieve any progress without the use of industrial robotics. There are several areas for using this technique, for example: welding and coating robots, mainly used in the automotive industry; assembly robots.

The goal of this work is to design and complete an industrial robot tasked with accomplishing tasks that are predefined, such as welding, and then control it from a distance to conduct the operation remotely.

Keywords: robot mathematical model, direct geometrical model, indirect geometric model, remote operation

Introduction générale

Un robot industriel est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains. Actuellement le monde industriel ne peut pas avancer sans l'utilisation de la robotique industrielle. Il existe plusieurs domaines d'applications des robots industriels par exemple : les robots de soudure et de peinture, principalement utilisés dans l'automobile ; les robots d'assemblage, très utilisés dans l'industrie.

La télé-opération est une activité qui consiste à commander un robot à distance via des applications web.

L'objectif de ce travail est la conception et la réalisation d'un robot industriel destiné principalement à la réalisation des tâches bien définies tel que le soudage puis nous essayons de le commander à distance pour faire le télé-opération. Dans la partie de téléopération notre objectif est de modéliser et de programmer en JavaScript et HTML afin de commander à distance via d'une application web notre robot réalisé dans la première phase.

Pour arriver à notre objectif ce mémoire est structuré en trois chapitres. **Le premier chapitre** présente les différents types de robot industriel et leurs matériels courants utilisés, ensuite nous présentons le modèle géométrique direct et inverse en utilisant la méthode Denavit-Hartenberg ; ensuite nous détaillons dans la troisième partie de ce chapitre le concept de la télé opération. **Le deuxième chapitre** est concentré aux composants électroniques en particulier la carte électronique Arduino et le microcontrôleur se sont les pièces nobles de la robotique et les systèmes automatisés. **Le dernier chapitre** est consacré à la conception et la réalisation pratique de notre robot, dans ce chapitre nous présentons la partie de modélisation mathématique de robot en utilisant la méthode Denavit-Hartenberg , puis nous présentons les technologies électroniques et les outils informatiques utilisées pour réaliser le robot industriel et enfin nous présentons les mécanismes et les outils informatiques pour réaliser le mécanisme de la télé opération.

I. Introduction

L'automatisation des tâches dans l'industrie a donné naissance aux robots manipulateurs. Ces derniers prennent chaque jour une place plus importante à la fois dans le milieu industriel et les opérations en milieux hostiles.

Dans ce chapitre nous allons présenter les robots industriels et leurs utilisations puis les modèles mathématiques utilisés pour les commander et à la fin de ce chapitre nous allons présenter le mécanisme de la télé opération.

I.1. Le robot industriel

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

En fait, l'image que chacun se fait d'un robot est généralement vague, souvent un robot est défini comme un manipulateur automatique à cycles programmables. Le robot est décrit avec ses deux espaces de travail. L'espace de travail articulaire définit les limites mécaniques de chaque articulation (moteur) prise individuellement. L'espace de travail de l'outil décrit l'ensemble des points accessible par l'outil. Le plus souvent, le volume de travail (workspace ou espace de travail) réfère au volume de travail de l'outil [DJOUGANE Fadhila, 2012].

Le terme robot a été introduit pour la première fois par l'auteur Tchèque Capek en 1920 dans sa pièce de théâtre R.U.R. (Rossum's Universal Robot) et il est dérivé du mot robota qui signifie travailleur (de force) [BOUZIANE Fatima Zohra, 2013].

L'Organisation Internationale de Normalisation définit le robot comme étant un manipulateur à plusieurs degrés de liberté, à commande automatique, reprogrammable, multi-applications, mobile ou non, destiné à être utilisé dans les applications d'automatisation industrielle [PRIEL MARC].

Pour « mériter » le nom de robot, un système doit posséder une certaine flexibilité, caractérisée par les propriétés suivantes :

- La versatilité : Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière ;
- L'auto-adaptativité : Un robot doit pouvoir s'adapter à un environnement changeant au cours de

l'exécution de ses tâches.

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel.

1.1.1. Historique

La robotique industrielle a connu un essor entre 1950-1970. Elle a vu le jour en 1954 lorsque Georges DEVOL a pu réaliser son brevet sur la robotique. Dans ce brevet **Devol** a conçu un robot qu'il a intitulé **Unimate**. En 1961, le premier Unimate fut utilisé dans les usines de GENERAL MOTORS. En 1966, l'entreprise Unimation continue de développer des robots et élaborent notamment des robots permettant de faire d'autres tâches, comme des robots de manipulation matérielle ou encore des robots conçus pour la soudure ou pour d'autres applications de ce genre[**DJOUGANE Fadhila, 2012**].

- 1947: Premier Manipulateur électrique téléopéré.
- 1954: Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société animation (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.

1.1.2. Domaine d'application des robots industriels

Les robots industriels utilisés dans Plusieurs domaines, dans cette partie nous citons quelques exemples :

[Belbachir Larbi 2013] a modélisé et a programmé en 3D le robot RX90 pour l'objectif de permettre à un groupe d'apprenants de l'université de technologie à travers une station informatique réalise un Travail Pratique de Robotique



Figure I- 1: Robot RX90

[BENABDELLAH YUCEF ,2013] a modélisé et a programmé en 3D le robot PUMA 560 pour l'objectif de permettre à un groupe d'apprenants de l'université de Tiaret ou de l'institut national de technologie Kasrchalala à travers une station informatique réalise un Travail Pratique de Robotique

Figure I.2.



Figure I- 2: Puma 560 Manipulateur

[Ait Dahmane Kahina, 2014] dans son projet de master il a fait la conception et la réalisation d'un bras manipulateur à 3 degrés de liberté. Ce bras peut faire des mouvements diverses et cela grâce à l'utilisation d'un Automate Programmable Industriel (API).



Figure I- 3: bras Manipulateur de Ait Dahmane Kahina

KR 210 KUKA robots à six axes et bras coudés assurent la manipulation des pièces forgées [4].



Figure I- 4: Manipulation de pièces incandescentes

La dextre du Canadarm2 (Température du milieu externe $+149^{\circ}\text{C}$ ou -126°C) Le Spécial Propose Dextrose Manipulatoire (SPDM), ou le Dextre, est un robot à deux bras, télémanipulateur mène des activités qui auparavant nécessitaient des sorties dans l'espace [5].

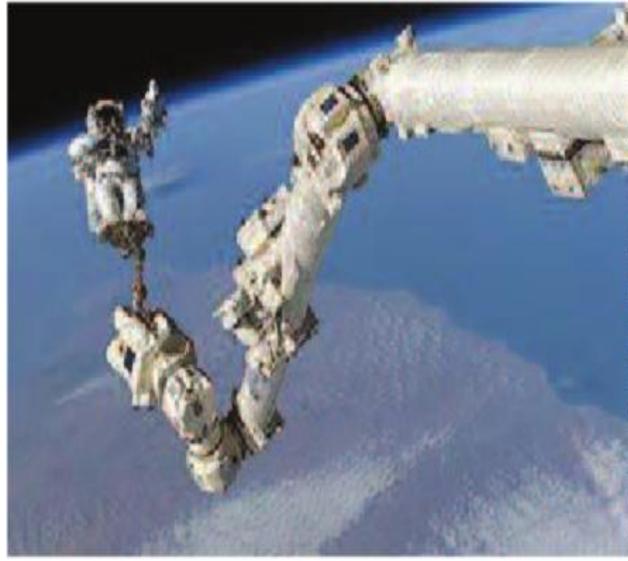


Figure I- 5: Le canadarm2 de la station spatiale Internationale

La Station spatiale internationale se trouve à 400 km au-dessus de la surface de la Terre. Elle est donc exposée à des températures allant jusqu'à 149°C, lorsqu'elle est alignée directement dans l'axe du Soleil. Quand elle se trouve dans l'ombre de la Terre, la température peut chuter jusqu'à -126°C. L'isolation multicouche (MLI) est utilisée pour contrôler le taux de transfert thermique. Une seule couche extérieure en tissu aluminé assure la protection des couches intermédiaires de la couverture du bras manipulateur et une grande partie de la station.

L'environnement à haute température comme les fours de traitement thermique est l'un des domaines exigeant le remplacement de l'homme par la machine dans les opérations de manipulation des pièces chaudes. La conception de ce type de machines ou robots est plus compliquée que celle des robots travaillant dans les conditions normales. Dans ce chapitre, on présente l'architecture et les caractéristiques des robots industriels, plus précisément les robots manipulateurs travaillant sous hautes températures. On s'intéresse à l'architecture de la structure mécanique articulée et l'organe terminal du robot industriel. L'étude bibliographique est constituée de deux parties. La première concerne les robots en milieux hostiles et l'architecture des robots manipulateurs. La deuxième se focalise sur les pinces en industries manufacturière.

I.1.3. Constituants mécaniques du robot

Deux sous-ensembles : une structure mécanique articulée et un effecteur

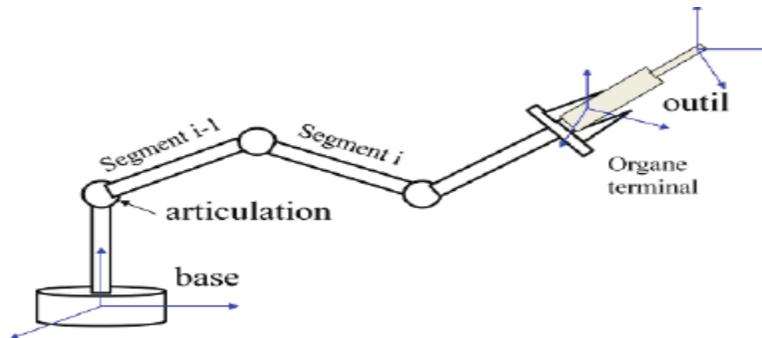


Figure I- 6: Constituants mécaniques du robot

I.1.3.1. La Structure Mécanique Articulée

Le système mécanique articulé (S.M.A.) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action. Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, généralement rigides, assemblés par des liaisons appelées articulations. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés. La structure mécanique d'un robot se distingue par trois ensembles : -Le véhicule Le véhicule assure le transport de la structure mécanique vers la zone d'action -Le porteur Le rôle du porteur consiste à mener un point du robot vers un lieu précis de l'espace. Il se compose des trois premiers degrés de liberté. Les différentes configurations sont résumées sur la (figure I.7).

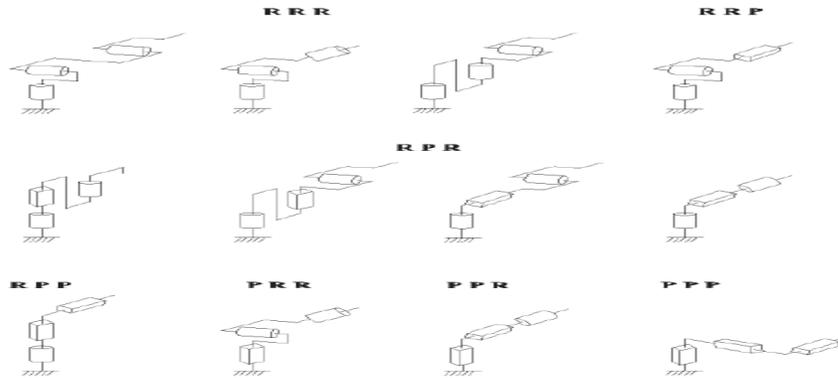


Figure I- 7: Principales architectures du porteur

Chaque architecture donne un espace de travail différent, convenable à la tâche à accomplir par le robot.

L'espace de travail est l'espace qui peut être balayé par le point de référence du poignet, augmenté de la plage de mouvement des articulations du poignet.

I.1.3.2. Types de chaînes de la SMA

On distingue trois types de chaînes:

- Chaînes ouvertes simples(A) (aucun retour mécanique d'un segment à un autre dans la chaîne).
- Chaînes arborescentes(B) (il existe plusieurs organes terminales qui agissent en parallèle)
- Chaînes fermées(C) (il existe un retour mécanique d'un ou plusieurs segments à un autre dans la chaîne)

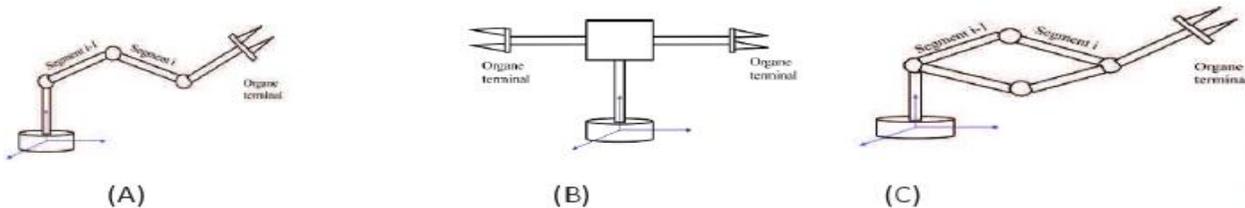


Figure I- 8: Types de chaînes

I.1.3.3. L'effecteur

L'effecteur est l'organe terminal du robot, fixé au poignet. Ce dernier se termine généralement par une plaque percée de trous filetés; cela permet la fixation de différents effecteurs à un robot universel et donc l'adaptation de celui-ci à des tâches spécifiques. Cet élément fera l'objet de la deuxième partie de l'étude bibliographique.

I.1.3.3.1. L'effecteur dans les robots manipulateurs

I.1.3.3.1.1. Classification des effecteurs

On peut distinguer deux grandes catégories d'effecteurs :

✓ Les outils

L'effecteur dans ce cas est dérivé des outils traditionnels: outil pour soudure, pistolet de peinture ou de collage, broches rotatives pour perçage, vissage, meulage, nettoyage, torche de chauffage, découpage, jet d'eau pour découpage...etc.



Figure I- 9: Outil de soudure et préhenseur

✓ Les organes de préhension

Ils sont utilisés pour saisir et maintenir les objets qui doivent être manipulés et déplacés par le robot. Parmi les applications, on peut citer le chargement et déchargement de machines, la saisie d'objets sur un convoyeur, le placement d'objets sur une palette de manutention... Un effecteur multiple comprend plusieurs effecteurs, fixés au même poignet. C'est souvent le cas des organes de préhension pour les applications de chargement-déchargement.

I.1.3.3.1.2. Les Pinces

Elles sont largement utilisées dans l'industrie manufacturière. Leur forme dépend de l'élément à saisir et l'environnement dans lequel elles agissent. Elles sont généralement composées (voir figure I.10) de :

1 : Corps de la pince

2 : Mâchoires

3 : Mors

4 : Rainure de capteur

5 : F = force de serrage sur un doigt seulement

6 : l'élément à saisir

7 : L = distance entre le centre de gravité de la charge et la surface de référence

8 : C = course d'un doigt

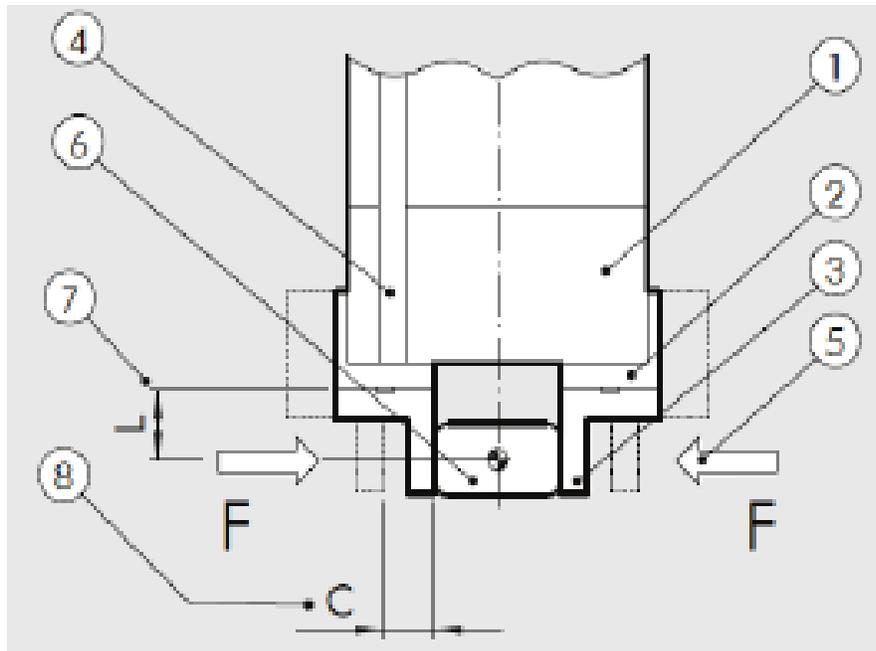


Figure I- 10: Outil de soudure et préhenseur

Les pinces peuvent classer en fonction du type de déplacement des doigts :

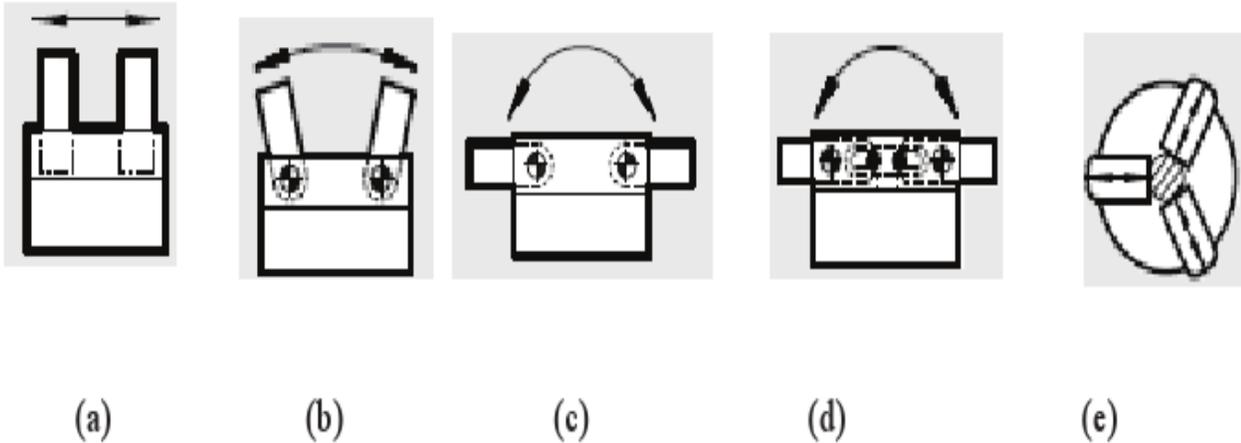


Figure I- 11: Type de pinces

✓ Pinces parallèles

Les doigts ont un déplacement linéaire. Ils peuvent être au nombre de 2, 3 ou parfois 4. Pinces

✓ Angulaires

Les doigts sont articulés et décrivent un mouvement en arc de cercle. Elles ont, en général un coût plus réduit que les pinces parallèles, mais avec quelques limitations (voir figure 1.9). Avec ce type de pinces : -si la pièce a des dimensions variables, la surface de contact varie. -si la pièce est cylindrique avec des dimensions variables, la position de l'axe de la pièce varie.

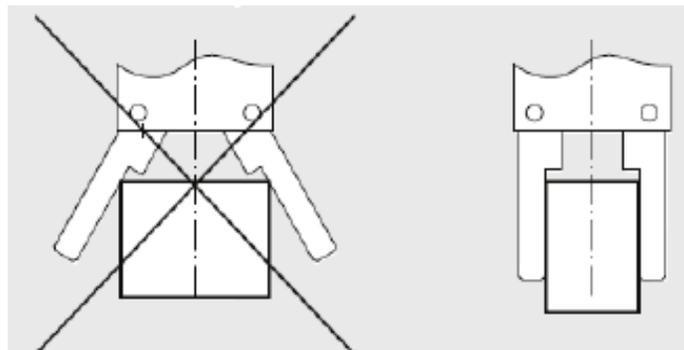


Figure I- 12: Limites d'utilisation des pinces à ouverture angulaire

✓ Pincés angulaires avec doigts rétractables

Les doigts ont un angle d'ouverture d'environ 90° . Les doigts de serrage peuvent s'effacer complètement sur la surface supérieure de la pince, permettant ainsi dans certains cas, d'éviter un mouvement linéaire de reculement

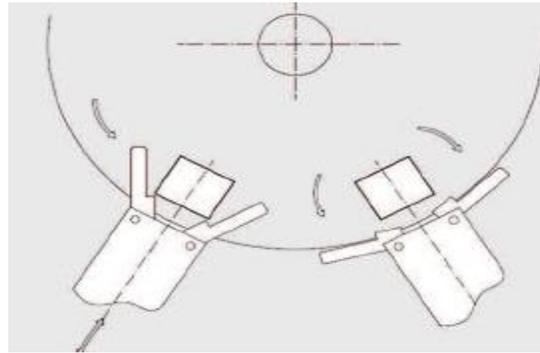


Figure I- 13: pincés à ouverture angulaire

✓ Pincés à genouillère

Pincés angulaires avec un mécanisme à genouillère, assurant une grande force de serrage. Le serrage est irréversible même lorsqu'il n'y a plus de pression, aussi la pièce ne peut être relâchée accidentellement. L'angle d'ouverture est de 90° , aussi elle agit comme une pince à doigts rétractables. La force de serrage n'est importante que sur une plage d'angle de rotation limitée.

✓ Pince à serrage concentrique(e) (intérieur ou extérieur)

Les pincés à deux doigts sont utilisées pour les pièces de forme prismatique, ou cylindrique à un seul diamètre. Les pincés à trois doigts peuvent être utilisées sur des pièces cylindriques à différents diamètres. Les pincés parallèles, concentrique et angulaire peuvent travailler dans les deux sens pour réaliser des serrages sur l'extérieur de la pièce ou en intérieur, dans un alésage par exemple (voir figure 1.12).



Figure I- 14: Serrage intérieur et extérieur

La pince angulaire à ouverture totale (180°), quant à elle, elle ne fonctionne qu'en serrage externe. En effet, elle a été conçue pour dégager totalement les doigts vers l'extérieur, afin de supprimer un mouvement de dégagement au manipulateur (figure 1.13)

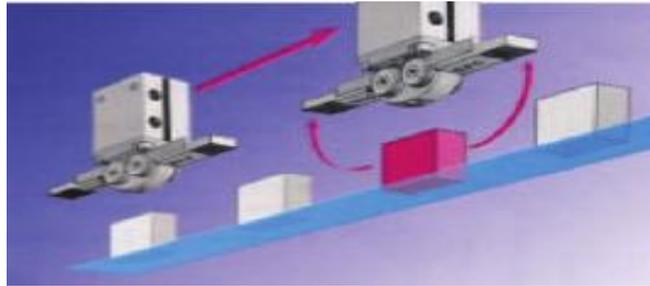


Figure I- 15: Dégagement complet des doigts de la pince à ouverture totale

1.2. Les modèles mathématiques des robots en générale

La modélisation de Système poly articulé a pour but de représenter au mieux le robot dans son environnement pour ensuite lui programmer des trajectoires avec la Planification de mouvement. La modélisation du robot consiste à représenter son comportement par des équations algébriques, soit du point de vue des positions, c'est le modèle géométrique, soit de point de vue des vitesses, c'est le modèle cinématique ou encore en considérant les efforts mis en jeu, c'est le modèle dynamique.

1.2.1. Degré de libertés

Positionnement d'un solide dans l'espace

La position d'un solide dans l'espace requiert 6 paramètres indépendants (Figure I.16) :

- 3 paramètres indépendants définissent la position d'un point, noté P, du solide (coordonnées cartésiennes, cylindriques, sphériques, dans la base du repère fixe),

- 3 paramètres indépendants déterminent l'orientation du solide autour du point P (angles d'Euler, paramètres d'Euler, ...).

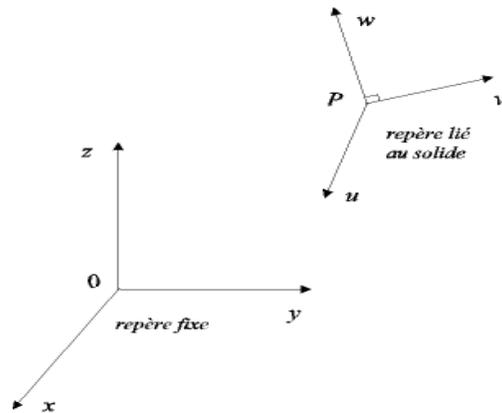


Figure I- 16: Changement de repère

On dit qu'un solide situé dans l'espace possède 6 degrés de liberté (**d.d.l.**). Réciproquement, il faut 6 variables de commande indépendantes pour placer de manière quelconque un solide dans l'espace. En pratique, les robots les plus courants sont dotés de 6 **d.d.l.**, d'au moins 6 actionneurs, ce qui permet de spécifier de manière quelconque la situation (position et orientation) de leurs organes terminaux [6].

1.2.2. Translation et rotation

On peut montrer que la position du repère de base R_1 se déduit de celle du repère de base R_0 par une translation et une rotation.

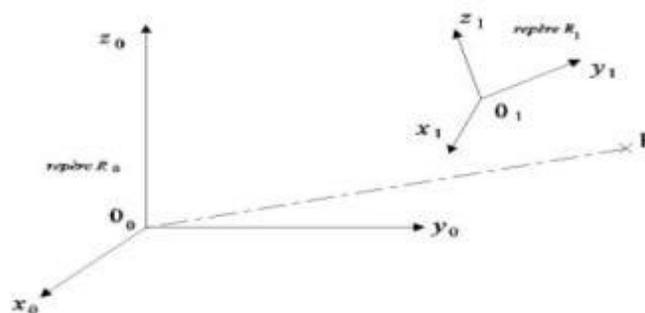


Figure I- 17: Positionnement d'un solide dans l'espace.

Le but est d'exprimer dans le repère de base R_0 les coordonnées du point P ayant pour coordonnées X_1, Y_1, Z_1 dans le repère de base R_1 (i.e. : $\vec{O_1P_{/1}} = (X_1, Y_1, Z_1)^t$), sachant que l'origine du repère R_1 , c'est-à-dire O_1 , a pour coordonnées a, b, c dans le repère R_0 .

On a donc:

$$\vec{O_1P_{/1}} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \text{ C'est-à-dire : } \vec{O_1P_{/1}} = X_1 \vec{x}_1 + Y_1 \vec{y}_1 + Z_1 \vec{z}_1 .$$

D' où :

$$\begin{aligned} \vec{O_0P_{/0}} &= \vec{O_0O_{1/0}} + \vec{O_1P_{/0}} \\ &= \vec{O_0O_{1/0}} + R_{01} \times \vec{O_1P_{/1}} \\ &= \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \end{aligned}$$

La matrice de rotation, notée $R_{01} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ contient les vecteurs de base de R_1 c'est-à-dire $\vec{x}_1, \vec{y}_1, \vec{z}_1$

exprimés dans la base R_0 , c'est-à-dire en fonction des vecteurs $\vec{x}_0, \vec{y}_0, \vec{z}_0$.

1.2.2.1. Matrices de rotation

Cas particulier de rotation autour d'un axe. Par convention, on considère que l'articulation rotoïde d'un robot se fait autour d'un des 3 axes d'un repère orthonormé. Soit, par exemple, une rotation d'angle θ_{x_0} autour de l'axe x_0 .

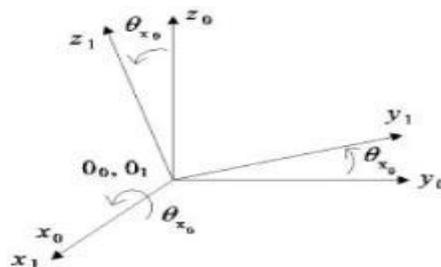


Figure I- 18: Orientation d'un repère.

On a :

$$R_{01}(x_0, \theta_{x_0}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{x_0}) & -\sin(\theta_{x_0}) \\ 0 & \sin(\theta_{x_0}) & \cos(\theta_{x_0}) \end{pmatrix}$$

Exprimons dans le repère R_0 les coordonnées (X_1, Y_1, Z_1) du point P dans le repère R_1 , sachant que l'origine O_1 du repère R_1 a pour coordonnées a, b, c dans le repère R_0 .

$$\text{On a : } \vec{O_0P_{/0}} = \vec{O_0O_{1/0}} + R_{01} \times \vec{O_1P_{/1}}$$

$$= \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{x_0}) & -\sin(\theta_{x_0}) \\ 0 & \sin(\theta_{x_0}) & \cos(\theta_{x_0}) \end{pmatrix} \times \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$$

1.2.2.2. Matrice de transformation homogène

La présence conjointe de produits et de sommes dans l'équation vectorielle $\vec{O_0P_{/0}} = \vec{O_0O_{1/0}} + R_{01} \times \vec{O_1P_{/1}}$ est peu commode pour effectuer des calculs systématiques, dus par exemple à des changements successifs de repères. On lui préfère une représentation matricielle de dimension 4, basée sur les coordonnées homogènes.

Soit un point M de l'espace, rapporté à trois axes rectangulaires, donné par la relation :

$$\vec{OM} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Alors la représentation du point M à l'aide de coordonnées homogènes est faite avec un quaternion :

$$\vec{OM} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{Avec : } a = \frac{x}{w}, b = \frac{y}{w}, c = \frac{z}{w} \text{ et } w = 1.$$

Soit la matrice de transformation homogène T_{01} correspondant à la matrice partitionnée suivante :

$$T_{01} = \begin{pmatrix} R_{01(3,3)} & t_{01(3,1)} \\ 000 & 1 \end{pmatrix}$$

Avec : $R_{01(3,3)}$ représente la rotation et $t_{01(3,1)}$ la translation.

Remarquons que le facteur d'échelle est unitaire ($w = 1$).

La matrice T_{01} représente la transformation permettant de passer du repère R_1 au repère R_0 .

En effet, elle permet d'exprimer dans le repère R_0 les coordonnées d'un vecteur exprimées dans le repère R_1 . Autrementdit, on a:

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = T_{01} \times \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

1.2.3. Modèle géométrique direct

Le modèle géométrique direct (MGD) d'un bras manipulateur exprime la situation de son Organe Terminal (OT) en fonction de sa configuration. Donc il exprime la position instantanée de l'OT par rapport au repère fixe lié au bâti.

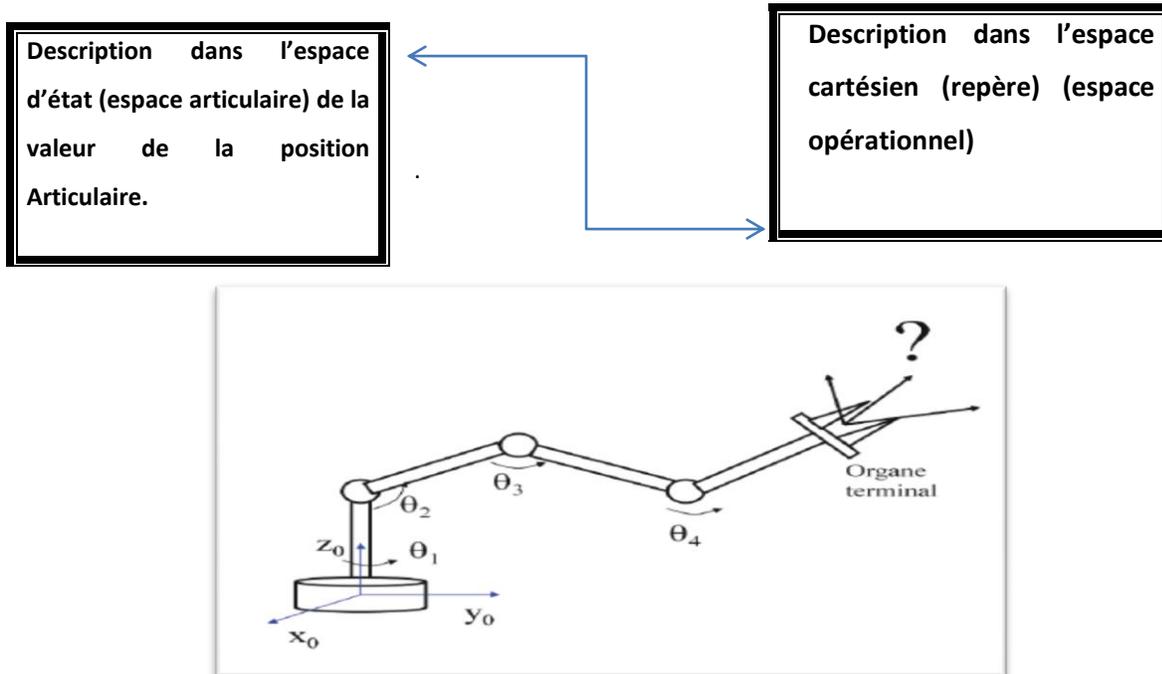


Figure I- 19: SMA Paramétrée

I.2.3.1. Calcul du MGD

La méthode de **Denavit-Hartenberg** est la plus répandue. Dans nos calculs, on utilise DH non modifiée. Pour définir ces transformations, il est tout d'abord nécessaire de définir les axes des liaisons :

-Les axes \vec{Z}_n sont suivant les axes des liaisons.

-Les axes \vec{X}_n sont parallèles à la normale commune à \vec{Z}_{n-1} et \vec{Z}_n soit :

$$\vec{X}_n = \vec{Z}_{n-1} \wedge \vec{Z}_n$$

-Les axes \vec{Y}_n sont choisi de manière à former un trièdre direct avec les axes \vec{X}_n et \vec{Z}_n

Chaque transformation entre deux corps successifs est donc décrite par quatre paramètres :

1. d , la distance selon l'axe \vec{Z}_n entre les axes \vec{X}_n et \vec{X}_{n+1} .
2. θ l'angle entre autour de l'axe \vec{Z}_n entre les axes \vec{X}_n et \vec{X}_{n+1} .
3. R , la distance selon l'axe \vec{X}_n entre les axes \vec{Z}_n et \vec{Z}_{n-1} . C'est donc également la longueur de la normale commune.
4. α , l'angle entre autour de l'axe \vec{X}_n entre les axes \vec{Z}_n et \vec{Z}_{n-1} .

En multipliant les matrices de rotation et de translation élémentaire, on peut obtenir la transformation globale entre deux liaisons successives qui s'écrit :

$$\mathbf{T} = \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & d_j \\ \cos(\alpha_j)\sin(\theta_j) & \cos(\alpha_j)\cos(\theta_j) & -\sin(\alpha_j) & -r_j.\sin(\alpha_j) \\ \sin(\alpha_j)\sin(\theta_j) & \sin(\alpha_j)\cos(\theta_j) & \cos(\alpha_j) & r_j.\cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

I.2.3.2. Le modèle géométrique direct et inverse

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

Dans le cas d'une chaîne simple ouverte, il peut être représenté par la matrice $T_{0,n}$ qui se calcule par :

$$T_{0,n} = T_{0,1}(\theta_1) \times T_{1,2}(\theta_2) \times \dots \times T_{n-1,n}(\theta_n)$$

I.2.3.3. Tableau de Denavit-Hartenberg

Dan cette section nous prenons l'exemple de Bans dampe de la figure I.19. voila tableau de **Denavit-Hartenberg** :

Articulation	1	2	3	4	5
α_j	0	90°	0	0	90°
d_j	0	0	d_1	d_2	0
θ_j	q_1	q_2	q_3	q_4	q_5
R	r_1	0	0	0	0

On a :

$$T_{j-1,j} = \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & a_j \\ \cos(\alpha_j)\sin(\theta_j) & \cos(\alpha_j)\cos(\theta_j) & -\sin(\alpha_j) & -r_j.\sin(\alpha_j) \\ \sin(\alpha_j)\sin(\theta_j) & \sin(\alpha_j)\cos(\theta_j) & \cos(\alpha_j) & r_j.\cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Sachant que **n=5**

Pour calculer $T_{0,5}$ on doit calculer : $T_{0,1}(\theta_1)$, $T_{1,2}(\theta_2)$ jusqu'à $T_{4,5}(\theta_5)$.

$$T_{0,1}(\theta_1) = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Et} \quad T_{1,2}(\theta_2) = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{2,3}(\theta_3) = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & d_1 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Et} \quad T_{3,4}(\theta_4) = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & d_2 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{T}_{4,5}(\theta_5) = \begin{pmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

I.2.3.4. Calcul de la matrice finale

On multiplie la matrice $\mathbf{T}_{4,5}$ par $\mathbf{T}_{3,4}$ pour qu'on obtient la matrice $\mathbf{T}_{3,5}$, on a donc :

$$\mathbf{T}_{3,5} = \mathbf{T}_{3,4} \times \mathbf{T}_{4,5} = \mathbf{T}_{3,5}(\theta_5) = \begin{pmatrix} \cos(\theta_4)\cos(\theta_5) & -\cos(\theta_4)\sin(\theta_5) & \sin(\theta_4) & d_2 \\ \sin(\theta_4)\cos(\theta_5) & -\sin(\theta_4)\sin(\theta_5) & -\cos(\theta_4) & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

On a: $\mathbf{T}_{1,3} = \mathbf{T}_{1,2} \times \mathbf{T}_{2,3}$

$$\mathbf{T}_{13} = \begin{pmatrix} \cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3) & -\cos(\theta_2)\sin(\theta_3) - \sin(\theta_2)\cos(\theta_3) & 0 & \cos(\theta_2)d_1 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2)\cos(\theta_3) + \cos(\theta_2)\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3) & 0 & \sin(\theta_2)d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Soit encore:

$$\mathbf{T}_{0,3} = \mathbf{T}_{0,1} \times \mathbf{T}_{1,3}$$

$$\mathbf{T}_{03} = \begin{pmatrix} \cos(\theta_1)(\cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3)) & \cos(\theta_1)(-\cos(\theta_2)\sin(\theta_3) - \sin(\theta_2)\cos(\theta_3)) & \sin(\theta_1) & \cos(\theta_1)\cos(\theta_2)d_1 \\ \sin(\theta_1)(\cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3)) & \sin(\theta_1)(-\cos(\theta_2)\sin(\theta_3) - \sin(\theta_2)\cos(\theta_3)) & -\cos(\theta_1) & \sin(\theta_1)\cos(\theta_2)d_1 \\ \sin(\theta_2)\cos(\theta_3) + \cos(\theta_2)\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3) & 0 & \sin(\theta_2)d_1 + R_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Pour obtenir la matrice finale on multiplie $\mathbf{T}_{0,3}$ par $\mathbf{T}_{3,5}$ et on aura :

$$\mathbf{T}_{0,5} = \mathbf{T}_{0,3} \times \mathbf{T}_{3,5} =$$

$$\begin{pmatrix} C_1S_{11}C_4C_5 + C_1S_{12}S_4C_5 + S_1S_4 & -C_1S_{11}C_4S_5 - C_1S_{12}S_4S_5 + S_1C_4 & C_1(C_2C_3 - S_2S_3)S_4 - C_1(-C_2C_3 - S_2C_3)C_4C_1S_{11}d_2 & C_1S_{11}d_2 + C_1C_2d_1 \\ S_1S_{11}C_4C_5 + S_1S_{12}S_4C_5 + & -S_1S_{11}(C_4S_5 - S_1)S_{12}S_4S_5 - C_1C_4 & ((S_1C_2C_3 - S_2S_3)S_3)S_4 - S_1(-C_2S_3 - S_2C_3)C_4 & S_1C_2C_3 - S_2S_3d_2 + S_1C_2d_1 \\ -S_{12}C_4C_5 + S_{11}S_4C_5 & -S_{12}C_3S_4 - S_{11}C_4 & -(C_2C_3 - (S_2S_3)S_4 - (C_2C_3 - S_2S_3))C_4 & S_2C_3 + (C_2S_3)d_2 + S_2(d_1) + R_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned}C_1 &= \cos(\theta_1), \\C_2 &= \cos(\theta_2), \\C_3 &= \cos(\theta_3), \\C_4 &= \cos(\theta_4), \\C_5 &= \cos(\theta_5), \\S_1 &= \sin(\theta_1), \\S_2 &= \sin(\theta_2), \\S_3 &= \sin(\theta_3), \\S_4 &= \sin(\theta_4), \\S_5 &= \sin(\theta_5), \\S_{11} &= C_2 C_3 - S_2 S_3, \\S_{12} &= -C_2 S_3 - S_2 C_3.\end{aligned}$$

1.2.4. Inversion du modèle géométrique

Dans ce modèle on a déterminé les coordonnées articulaires permettant d'obtenir une situation désirée pour l'organe terminal et spécifiée par les coordonnées opérationnelles

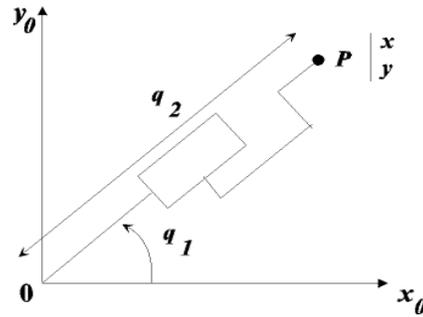
Il n'existe pas de méthode systématique d'inversion du modèle géométrique. Lorsqu'elle existe, la forme explicite, issue d'une inversion mathématique, qui donne toutes les solutions possibles au problème inverse (il y a rarement unicité de la solution) constitue le modèle géométrique inverse. Il existe un certain nombre de méthodes pour calculer le modèle géométrique inverse, notamment la méthode de Paul13 qui traite séparément chaque cas particulier et qui convient pour la plupart des robots industriels. Lorsque le modèle géométrique inverse n'existe pas, c'est-à-dire qu'il n'existe pas une forme explicite, on peut calculer une solution particulière du problème inverse par des procédures numériques, qui est une solution locale au sens où elle dépend des conditions initiales .

Notons que de telles méthodes peuvent être pénalisantes du point de vue du temps de calcul.

1.2.4.1.Exemples (simples)

1er exemple:

Soit le manipulateur évoluant dans un plan et décrit dans la figure suivante.



On a le modèle géométrique direct suivant $\begin{cases} x = q_2 \cos(q_1) \\ y = q_2 \sin(q_1) \end{cases}$

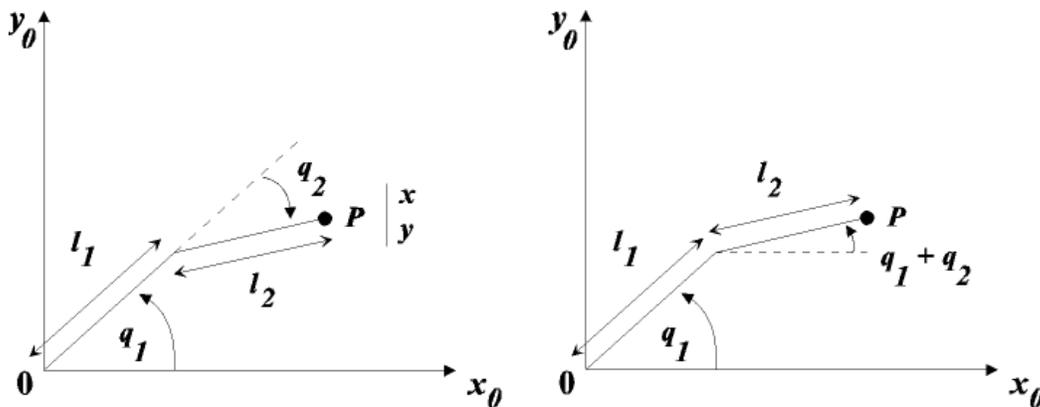
Une démarche analytique simple permet de déterminer le modèle géométrique inverse. On a :

$$\tan(q_1) = \frac{y}{x} \Rightarrow q_1 = \tan^{-1}\left(\frac{y}{x}\right)$$

Et $x^2 + y^2 = q_2^2 \Rightarrow q_2 = \sqrt{x^2 + y^2}$

2^{ème} exemple :

Soit le manipulateur RR plan décrit ci-dessous.



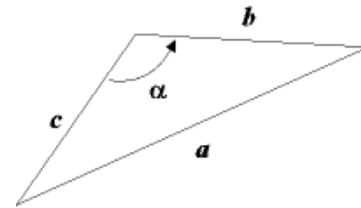
On a le modèle géométrique direct suivant : $\begin{cases} x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{cases}$

Soit un système de 2 équations à 2 inconnues.

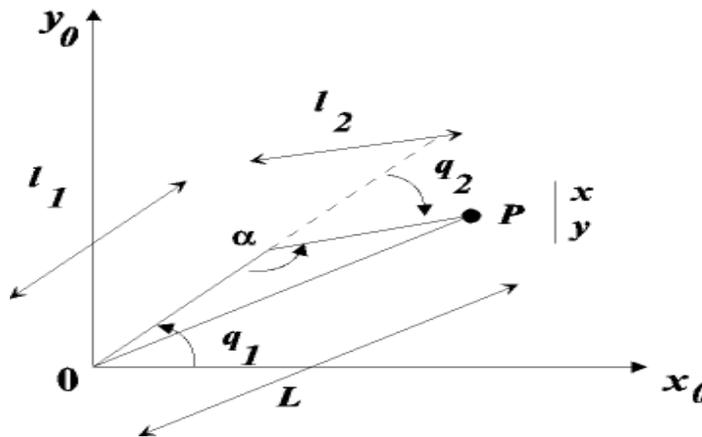
Une démarche analytique, procédant par substitution, permet de déterminer le modèle géométrique inverse.

Rappel (Théorème de Pythagore généralisé) :

On a: $a^2 = b^2 + c^2 - 2bc \cos (\alpha)$



Complétons le schéma du modèle géométrique direct :



On a les relations suivantes :

$l^2 = x^2 + y^2$ Et $l^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos (\alpha)$ avec : $\alpha = \pi + q_2$

avec (d'après le théorème de Pythagore généralisé),

D'où $x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2 \cos (q_2)$ (On a $\cos(\pi + \alpha) = -\cos(\alpha) =$)

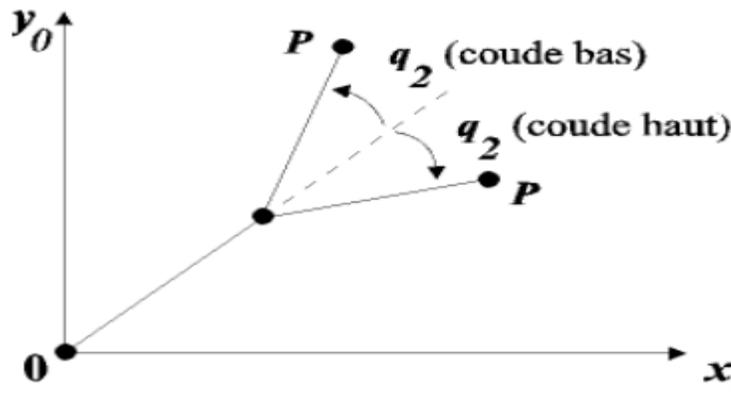
Soit $\cos (q_2) = \frac{x^2+y^2-(l_1^2+l_2^2)}{2l_1l_2}$

sachant que $\cos (q) = \alpha$ avec $\alpha \in [-1,1] \Rightarrow q = \mp \text{Arcos} (\alpha)$, il en résulte que

$q_2 = \pm \text{Arcos} \left(\frac{x^2+y^2-(l_1^2+l_2^2)}{2l_1l_2} \right)$, sous réserve que $-1 \leq \frac{x^2+y^2-(l_1^2+l_2^2)}{2l_1l_2} \leq 1$

Remarque :

- La condition $-1 \leq \frac{x^2+y^2-(l_1^2+l_2^2)}{2l_1l_2} \leq 1$ indique que la position du point P doit être atteignable.
- Lorsque q_2 est positif (resp., négatif), le robot a une posture coude bas (resp., coude haut), voir figure ci-dessous.



D'autre part, en développant les expressions $\cos(q_1 + q_2)$ et $\sin(q_1 + q_2)$ dans le système correspondant au modèle géométrique direct, on obtient :

$$\begin{cases} x = (l_1 + l_2 \cos q_2) \cos q_1 - l_2 \sin q_2 \sin q_1 \\ y = (l_1 + l_2 \cos q_2) \sin q_1 + l_2 \sin q_2 \cos q_1 \end{cases}$$

Calculons le déterminant de ce système d'équations linéaires par rapport aux 2 inconnues $\cos q_1$ Et $\sin q_1$, on a :

$$\begin{vmatrix} l_1 + l_2 \cos q_2 & -l_2 \sin q_2 \\ l_2 \sin q_2 & l_1 + l_2 \cos q_2 \end{vmatrix} = l_1^2 + l_2^2 + 2l_1l_2 \cos (q_2) = x^2 + y^2$$

(cf. Eq. (1)), soient, via la méthode de Cramer,

$$\cos q_1 = \frac{\begin{vmatrix} x & -l_2 \sin q_2 \\ y & l_1 + l_2 \cos q_2 \end{vmatrix}}{x^2 + y^2} \quad \text{Et} \quad \sin q_1 = \frac{\begin{vmatrix} l_1 + l_2 \cos q_2 & x \\ l_2 \sin q_2 & y \end{vmatrix}}{x^2 + y^2}$$

Soient :

$$\cos q_1 = \frac{1}{x^2+y^2} (x (l_1 + l_2 \cos q_2) + y l_2 \sin q_2)$$

$$\sin q_1 = \frac{1}{x^2+y^2} (y (l_1 + l_2 \cos q_2) - x l_2 \sin q_2)$$

Il en résulte :

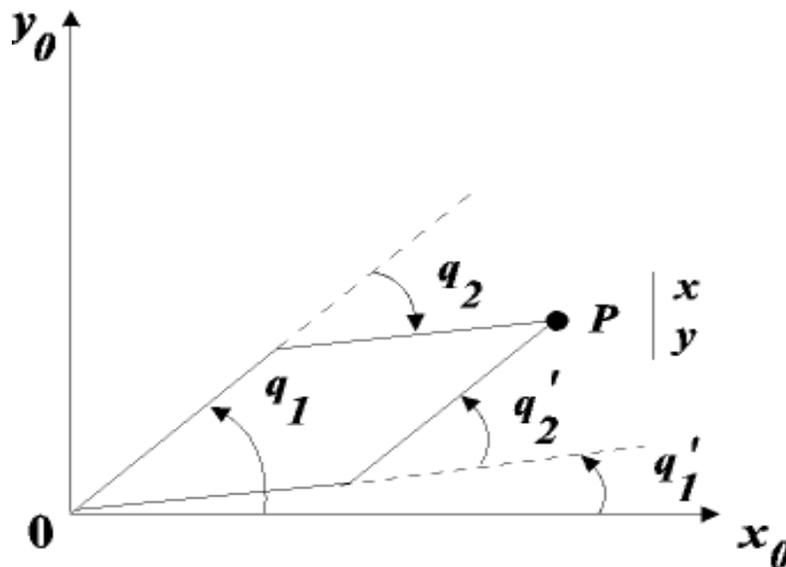
$$q_1 = \tan^{-1} \left(\frac{y (l_1 + l_2 \cos q_2) - x l_2 \sin q_2}{x (l_1 + l_2 \cos q_2) + y l_2 \sin q_2} \right)$$

Ainsi, on aboutit au modèle géométrique inverse suivant :

$$q_1 = \tan^{-1} \left(\frac{y (l_1 + l_2 \cos q_2) - x l_2 \sin q_2}{x (l_1 + l_2 \cos q_2) + y l_2 \sin q_2} \right)$$

$$q_2 = \pm \cos^{-1} \left(\frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \right)$$

On remarque qu'il y a 2 solutions, correspondant à 2 postures différentes du bras (sous réserve qu'il n'y ait pas de butées sur les articulations) : L'une est dite « coude haut », l'autre « coude bas » (cf. figure suivante).



I.2.3.3. La télé-opération et la télé-robotique

Ces dernières années, les technologies de l'information et de la communication ont permis d'un côté, pour les activités du tertiaire de ne plus concentrer la production du service dans un lieu unique. D'un autre côté, ces technologies ont permis de dispenser la présence de l'homme dans des milieux dangereux (milieu nucléaire, haute température, etc.) ou tout simplement pour des missions d'exploration (milieu sous-marin, spatial, etc.).

Dans cette partie, nous présentons une forme de télétravail (travail à distance), celle liée au contrôle à distance de systèmes robotisés, nous présenterons alors la télé robotique et la télé opération et nous verrons comment les techniques de programmation web telle que javascript, HTML et http permet de réaliser cette opération.

Selon [Samir Otmane, 2018] La Télé opération désigne les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention (dispositif esclave), commandé à partir d'une station de contrôle (dispositif maître), par l'intermédiaire d'un canal de télécommunication (figure I.20).

Elle a pris ses origines dans le besoin de prolonger le geste de l'homme au-delà de la main, et se poursuit par l'ambition de se trouver là où on ne se trouve pas (Télé présence) comme illustré dans la figure I.21 [Samir Otmane, 2018].

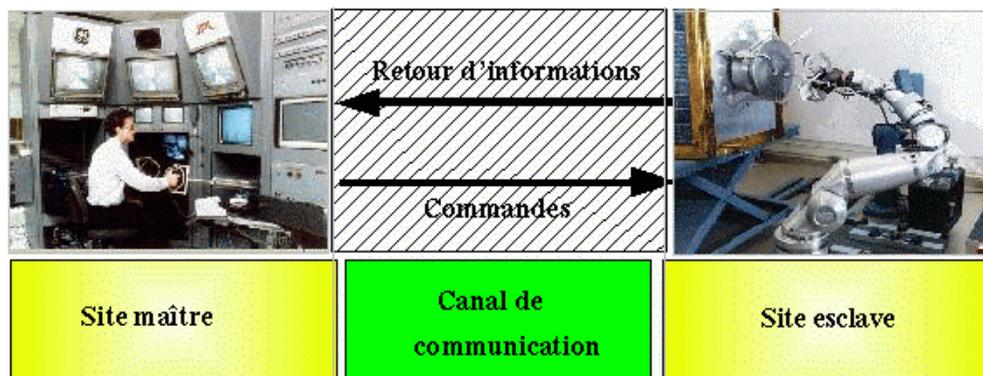


Figure I.20. Illustration de l'architecture générale d'un système de Télé opération



Figure I.21. Illustration d'un système de télé présence avec un robot mobile (Image du groupe de recherche en télé présence : California Corporation Telepresence Research)

Parmi les problèmes de la télé opération, nous présentons brièvement ceux évoqués et détaillés dans les deux ouvrages « télé opération et télé robotique » [COI 02] et « télé opération et réalité virtuelle » [KHE 02] à savoir :

- les délais (retards) de transmission ;
- la réalisation d'une transparence idéale ;
- la prise en compte du facteur humain et sécurité ;
- l'autonomie et partage de commande.

[Samir Otmane, 2018] a résumé les problèmes liés aux retards de transmission.

Dans les systèmes de télé opération originels, l'opérateur doit impérativement subir un apprentissage qui peut être long avant d'obtenir une bonne adaptation et une bonne maîtrise du système de télé opération. En plus l'adaptation obtenue peut ne pas être adéquate pour une autre architecture. La sophistication et le nombre important d'applications télé opérées engendre une augmentation de la charge de travail de l'opérateur, sa fatigue et ce qui entraîne une diminution de ses performances et augmente ainsi les risques d'erreurs de télé opération.

L'évolution de l'informatique et des supports de télécommunication numérique est à l'origine de l'évolution de la télé opération. Grâce à eux, les robots esclaves ont pu être déportés à des distances considérables [VER 84]. En effet, initialement l'opérateur commandait le robot esclave dans un niveau assez bas, les trajectoires issues des actions désirées sont envoyées directement via le dispositif maître. Le retour d'information (Images, force de contact, etc.) se fait directement vers l'opérateur. Mais

malheureusement, cette approche n'a pas duré très longtemps, car les chercheurs se sont vite rendu compte de la nécessité d'exploiter les possibilités que peuvent offrir des ordinateurs pour apporter une assistance à l'opérateur. Depuis, plusieurs approches et concepts ont été proposés pour améliorer et faire évoluer les systèmes de télé opération :

- la Télé opération Assistée par Ordinateur – TAO : Elle est vue comme une voie intermédiaire entre la télé opération originelle bas niveau et la supervision. L'objectif de la TAO est de réaliser à chaque instant et pour toute étape de la tâche, une exploitation optimale des ressources d'un ordinateur. La TAO bénéficie actuellement de l'utilisation de la réalité virtuelle et augmentée ;

- la Télé opération à Désignation d'Objectif (supervisée) – TDO : Dans ce mode, l'opérateur est plutôt vu comme un superviseur. L'intervention de l'opérateur se limite dans ce cas à la désignation d'objectifs qui seront réalisés par le robot ;

- la Télé opération Semi Autonome – TSA : La plupart des systèmes de télé opération actuels, s'orientent vers l'utilisation des deux concepts précédents (TAO et TDO). En effet ils tentent de moderniser la télé opération par une meilleure exploitation simultanée de l'autonomie du robot et des capacités de l'opérateur.

La Télé opération utilisée dans plusieurs domaines telles que : Le domaine spatial, le domaine nucléaire, le domaine médical et l'aide aux personnes handicapées, le domaine sous-marin, le domaine militaire ...etc.

1.4. Conclusion

Dans ce chapitre, nous avons présenté l'architecture mécanique générale et le modèle mathématique général sur les robots d'un bras manipulateur. Cette dernière est composée du porteur et du poignet. Le porteur, formé des trois premiers ddl, assure le positionnement de l'organe terminal. Son architecture, donne l'espace de travail du bras. Le poignet, est composé en maximum de trois liaisons rotoïdes et on peut traiter les problèmes mathématiquement, il détermine l'orientation de l'organe terminal. Dans la deuxième partie, on a présenté un ensemble d'outils utilisés en manutention d'objet et dans la dernière partie, nous avons parlé de la partie mathématique de robot. Nous sommes arrivés à retirer les critères de choix d'un organe de préhension et qui sont :

- Les caractéristiques du porteur dépendant surtout de la masse maximale transportable.
- Les données concernant la géométrie de l'objet à transférer
- La technologie de préhension, en particulier le mode de saisie et le principe du maintien en position de l'objet en cours de transfert.

A la fin de ce chapitre nous avons présenté la télé opération.

II. Introduction

N'importe quel système industriel constitué d'un ensemble des composants électroniques. Par exemple Si nous le considérons le coté mécanique de robot les pieds et mains dans l'homme donc la carte électronique (**microcontrôleur**) est la cervelle qui commandé et son contrôlé le cote mécanique. Pour notre bras de robot manipulateur nous chosions la carte électronique **Arduino** sur lesquelles se trouve un microcontrôleur. Dans ce chapitre nous allons présenter les composants électroniques d'un système industriel et en particulier les composants électroniques utilisés dans la fabrication de la robotique.

II.1. La carte d'Arduino

II.1.1. Présentation d'Arduino

Arduino est une carte électronique en matériel libre (**open hardware**) pour la création artistique interactive. Il est un projet créé par une équipe de développeurs, composée de six personnes : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino. [**Simply Arduino**]

II.1.2. Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino sur le marcher, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- **Arduino Leonardo**

L'arduino Leonardo est le digne successeur de l'arduino UNO mais elle a du mal à prendre la relève de l'Arduino Uno, plus mature et exempté de tout bug possible.

Chapitre II:

L'image de ce modelé est le suit :

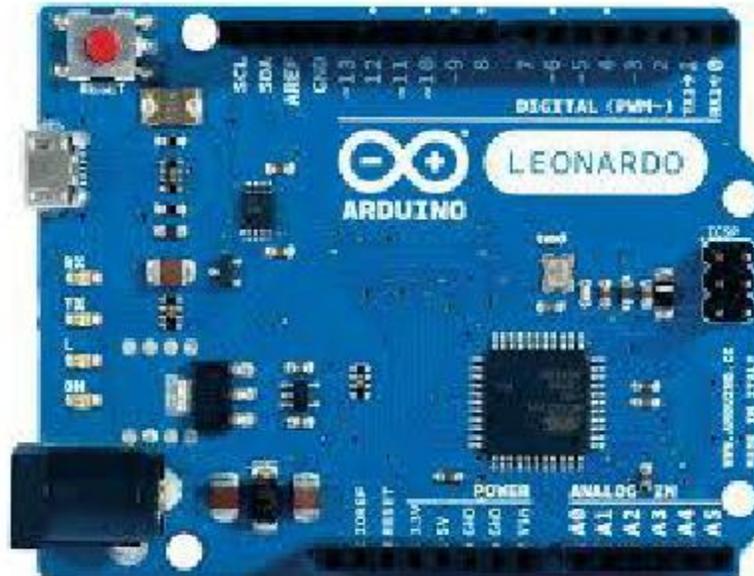


Figure II- 1: Arduino Leonardo

- **Arduino Mega**

L'arduino méga est une arduino UNO mais avec plus d'entrées et sorties (54 contre 14), plus de mémoire également ce qui peut être utile pour les programmes un peu complexe tout comme son processeur, plus puissant.



Figure II- 2: Arduino Méga

Chapitre II:

- **Arduino DUE**

L'Arduino DUE est la fameuse Arduino Méga. Celle-ci est une Arduino Méga plus puissante pour des calculs plus rapides.



Figure II- 3: Arduino DUE

- **Arduino Mini Pro**

Cette carte Arduino est simplifiée au maximum, elle n'intègre même pas de port USB. C'est la plus petite carte Arduino officielle, utile lors de projets professionnels.



Figure II- 4: Arduino Mini Pro

Chapitre II:

- **Arduino Yun**

La carte Arduino Yun est la première carte Arduino avec wifi intégré qui combine à la fois la puissance de linux avec la facilité d'utilisation d'Arduino.



Figure II- 5: Arduino Yun

- **Arduino Nano**

L'arduino Nano est une arduino miniaturisé qui permet la miniaturisation, Soudures nécessaires

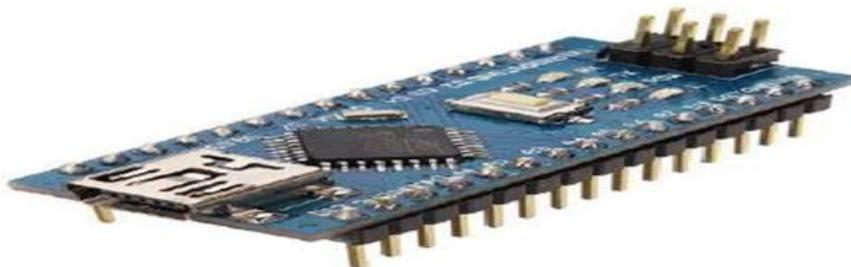


Figure II- 6: Arduino Nano

Chapitre II:

- **Arduino UNO**

Cette carte convient pour n'importe quel débutant et elle convient pour tous les petits projets. L'image de cette carte est ci-dessous. (La carte que nous avons utilisée dans notre réalisation)

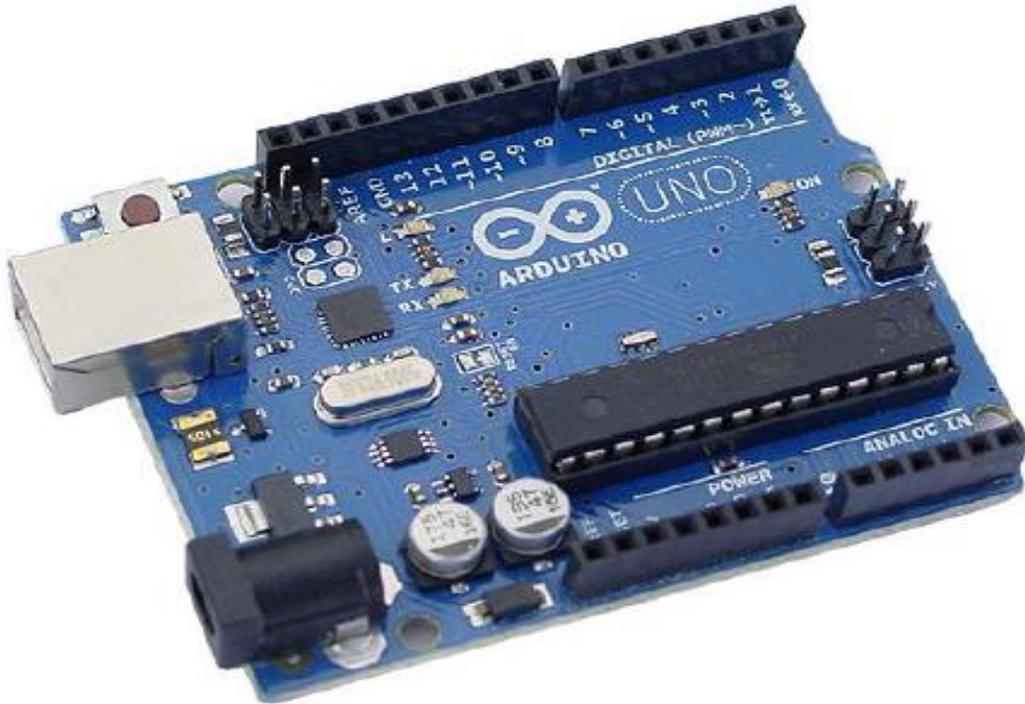


Figure II- 7: Arduino Uno

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal a 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure si dessous. [**Arduino applications avancées**]

II.1.3. *Pourquoi Arduino UNO*

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- **Le prix (réduits) :** les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (le prix est 2500 Dinars).
- **Un environnement de programmation clair et simple :** l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Multi plateforme :** le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. Par contre la plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Matériel Open source et extensible :** les cartes Arduino sont basées sur les Microcontrôleurs ATmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino.

II.1.4. *La constitution de la carte Arduino UNO*

Un module Arduino uno est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

II.1.4.1. *Partie matérielle*

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

II.1.4.1.1. *Le Microcontrôleur ATmega328*

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure I.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino. [Simply Arduino]



Le composant CMS



Le composant classique

Figure II- 8: Microcontrôleur ATmega328

ATmega328 contient les portes d'entrée et sortie qui ont des propriétés (interruption, PWM, CAN, TX/RX...)

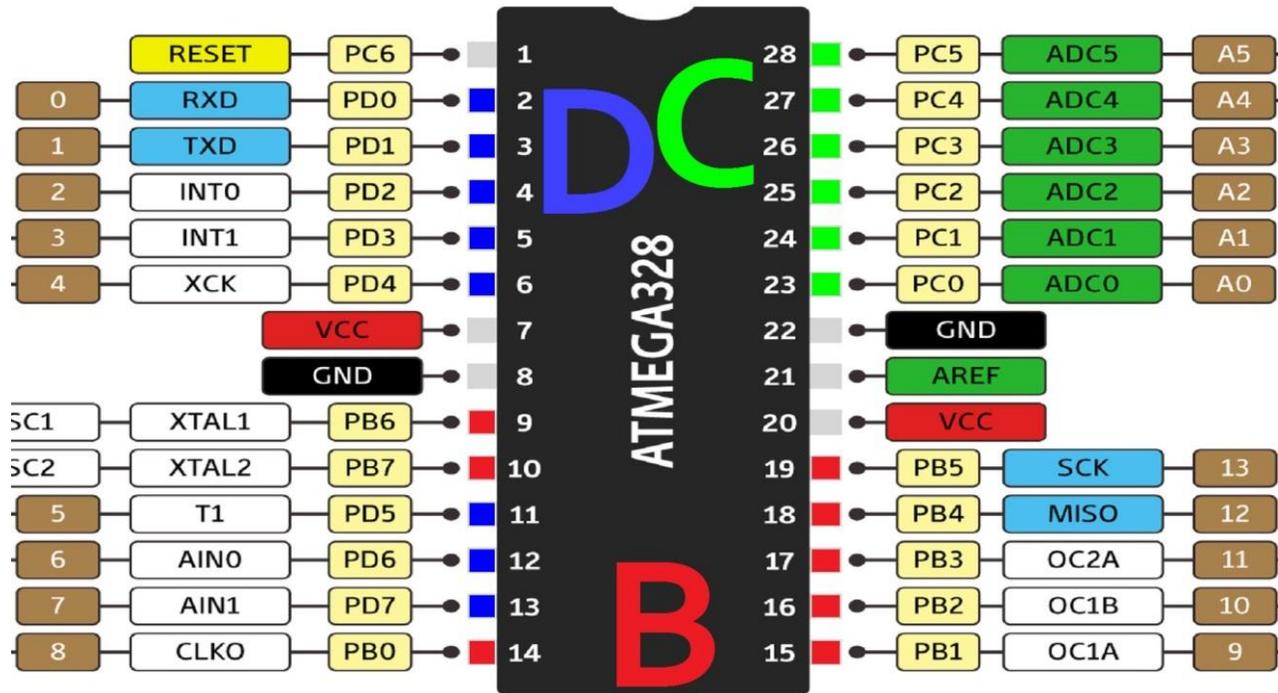


Figure II- 9:les portes d'entrée et sortie de ATmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM :** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [Bartmann E, 2014]

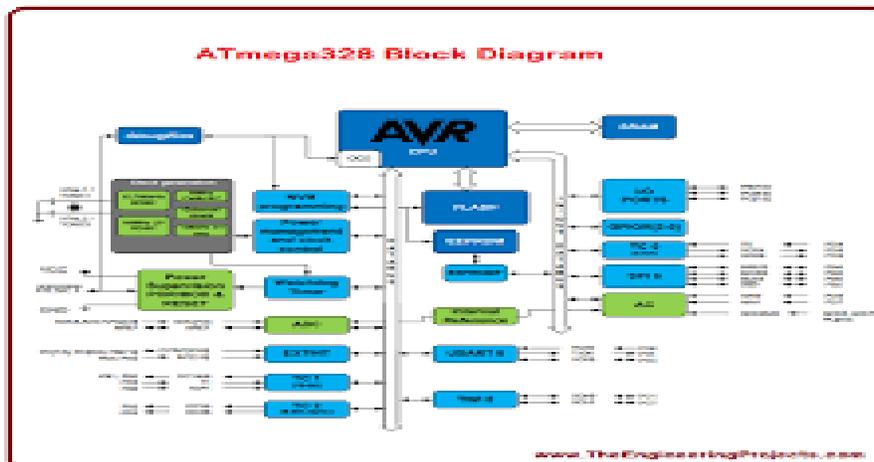


Figure II- 10: Atmiga328 structure

II.1.4.1.2. Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [**Three-phase inverter using Arduino**]

II.1.4.1.3. Les entrées et sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode ()`, `digitalWrite ()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digital Write (broche, HIGH)`. En plus, certaines broches ont des fonctions spécialisées:

- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée):

Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite ()`.

- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Méga.
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`).
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte. La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead ()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analog Reference ()` du langage Arduino.

Chapitre II:

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [arduino.cc]

II.1.4.1.4. Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil:

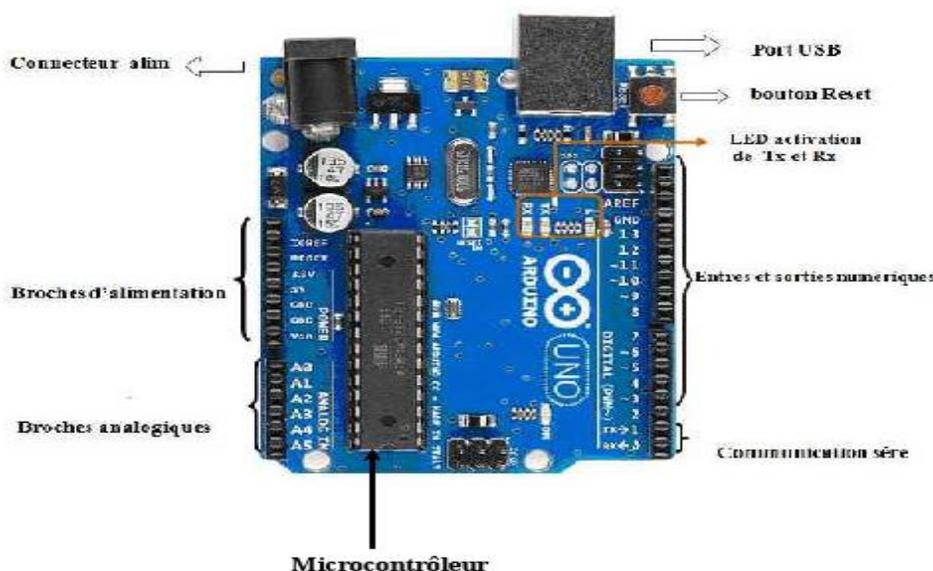


Figure II- 11: Constitution de la carte Arduino UNO

- **Les Accessoires de la carte Arduino**

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

- **Communication**

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

- **Le module Arduino Bluetooth**

Le module microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant. [Bartmann E., 2014]



Figure II- 12: Type de modules Bluetooth

- **Le module shield Arduino Wifi**

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



Figure II- 13:Type de modules Bluetooth

- **Les capteurs**

Un capteur est une interface entre un processus physique et une information manipulable. et fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique.



Figure II- 14: Capteurs Arduino

II.2. Les moteurs

II.2.1. Les moteurs électriques

Un moteur électrique est une machine électromécanique capable de transformer l'énergie électrique en énergie mécanique. Les moteurs électriques sont tous réversibles : ils sont capables de produire du courant électrique si on les fait tourner par un moyen mécanique.

II.2.2. Les types des moteurs électriques

II.2.2.1. Moteurs à courant continu DC

Un moteur à courant continu à aimant permanent se compose de trois éléments essentiels :

- **Une partie fixe** : l'inducteur (stator) constitué d'un aimant permanent.
- **Une partie mobile** : l'induit (rotor) comportant une ou plusieurs bobines tournant dans le champ magnétique créée par l'inducteur.
- Le collecteur qui associé aux balais, permet de relier les enroulements du rotor à l'extérieur du moteur.

[Theodore W, Gilbert S ,2000]

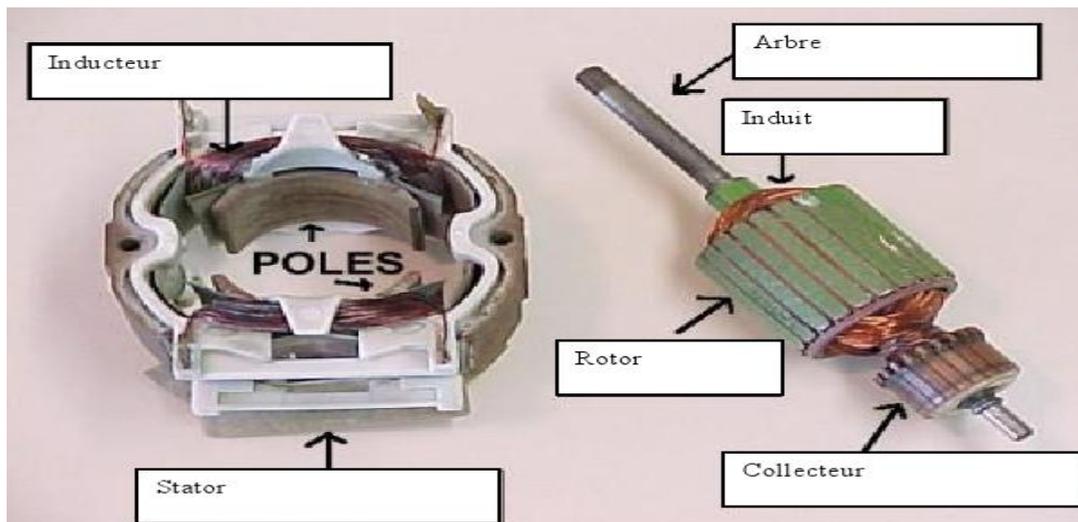


Figure II- 15: Représentation d'un Moteurs à courant continu DC

II.2.2.2. Moteurs à courant alternatif AC

Pour les applications de faible et moyenne puissance (jusqu'à quelques kilowatts), le réseau monophasé standard suffit. Pour des applications de forte puissance, les moteurs à courant alternatif sont généralement alimentés par une source de courants polyphasés. Le système le plus fréquemment utilisé est alors le triphasé utilisé par les distributeurs d'électricité, Ces moteurs alternatifs se déclinent en trois types :

- **Les moteurs universels :** Le moteur universel est un moteur asynchrone, c'est à dire qu'il n'est pas synchronisé à la fréquence du réseau. Ce dernier se compose d'un rotor et d'un stator bobiné en série, le moteur tourne toujours dans le même sens, car quel que soit la polarité, le champ magnétique du rotor et du stator sont toujours dans le même sens (les deux changent en même temps).
- **Les moteurs synchrones :** Comme génératrice. On l'appelle alors « alternateur ». Mis à part pour la réalisation de groupe électrogène de faible puissance, cette machine est généralement triphasée. Pour la production d'électricité.
- **Les moteurs asynchrones:** Connue également sous le terme d'origine anglo-saxonne de « machine à induction», est une machine à courant alternatif sans alimentation électrique du rotor. Le terme anglais provient des courants induits dans le rotor par induction magnétique. Le terme « asynchrone » provient du fait que la vitesse de ces machines n'est pas forcément proportionnelle à la fréquence des courants qui les traversent.

II.2.2.3. **Moteurs pas à pas**

Les moteurs pas à pas sont utilisés pour les positionnements angulaires précis (imprimantes, scanners, disques durs ...). Contrairement aux moteurs à courant continu, il ne nécessite pas de boucle d'asservissement et sont plus simples à commander. Il existe trois types de moteur : à aimant permanent, à reluctance variable et hybrides. Dans tous les types de moteur, on positionne le rotor en modifiant la direction d'un champ magnétique créé par les bobinages du stator. Ils nécessitent un circuit de commande qui comporte une partie logique et une commande de puissance. La partie logique détermine pour chaque pas quelles sont les bobines alimentées et le sens de rotation. La fréquence de l'horloge du circuit logique détermine la vitesse de rotation.

Ces moteurs pas à pas se déclinent en trois types :

- **Moteurs à aimant permanent :** C'est le modèle dont le fonctionnement est le plus simple. Le rotor est constitué d'un aimant permanent, et le stator comporte deux paires de bobines. En agissant sur les bobines alimentées, et le sens des courants, on fait varier le champ créé par le stator, à chaque pas, la direction du champ induit par le stator tourne de 90°. L'aimant permanent suit le déplacement du champ magnétique créé par les bobines et s'oriente selon une de ses quatre positions stables. Comme le rotor est aimanté, lorsque le moteur n'est pas

alimenté le flux magnétique dû à l'aimant permanent crée un couple résiduel en se plaçant dans l'axe de l'une des bobines.

- **Moteurs à réluctance variable** : Le rotor est en fer doux et comporte un nombre de pôles différent du stator. Le rotor se déplace pour que le flux le traversant soit maximum. Ces moteurs n'ont de couple de maintien si aucune bobine n'est alimentée.
- **Moteurs hybrides** : Le rotor est constitué par deux pièces en fer doux ayant chacune n pôles séparées par un aimant permanent magnétisé dans le sens de l'axe du rotor. Le nombre m de pôles du stator est différent de celui du rotor. Le rotor se déplace pour que le flux qui le traverse soit maximum. En mode pas entier, les bobines sont alimentées paire par paire alternativement avec inversion à chaque pas. Il est nécessaire d'avoir un rotor polarisé pour imposer le sens de rotation à chaque commutation.



Figure II- 16:un moteur pas à pas

II.2.2.4. Servomoteurs

Le servomoteur : un servomoteur est, comme son nom l'indique, un moteur mais avec quelques spécificités en plus. Contrairement à un moteur classique qui est utilisé pour tourner avec une vitesse proportionnelle à un courant ou à une tension, pour obtenir une position. Il effectue une rotation suivant un angle déterminé. Le plus souvent l'angle est compris entre 0 et 180° puis il garde cette position. Les servomoteurs sont des actionneurs. Très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un

Chapitre II:

bras qui porte une charge. Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.



Figure II- 17:Représentation d'un servomoteur

• Servomoteur et arduino (la commande)

Les servomoteurs sont pilotés par un fil de commande et alimentés par deux autres fils. Habituellement, ces 3 fils sont rassemblés dans une prise au format standard.



Figure II- 18: Fiche de commande d'un servomoteur

- Un fil rouge est relié à l'alimentation positive (+5 ou +6 V selon le servo)
- le fil marron est relié à la masse (GND)
- le fil jaune est utilisé pour la commande.

Chapitre II:

Il y aurait beaucoup à dire sur le fonctionnement d'un servomoteur, ses composants, son moteur et le petit potentiomètre qui permet de connaître sa position mais cette fiche va droit au but et se limitera à son utilisation avec l'Arduino. Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré. Avec la programmation de l'Arduino, Une bibliothèque dédiée, la bibliothèque « servo », permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

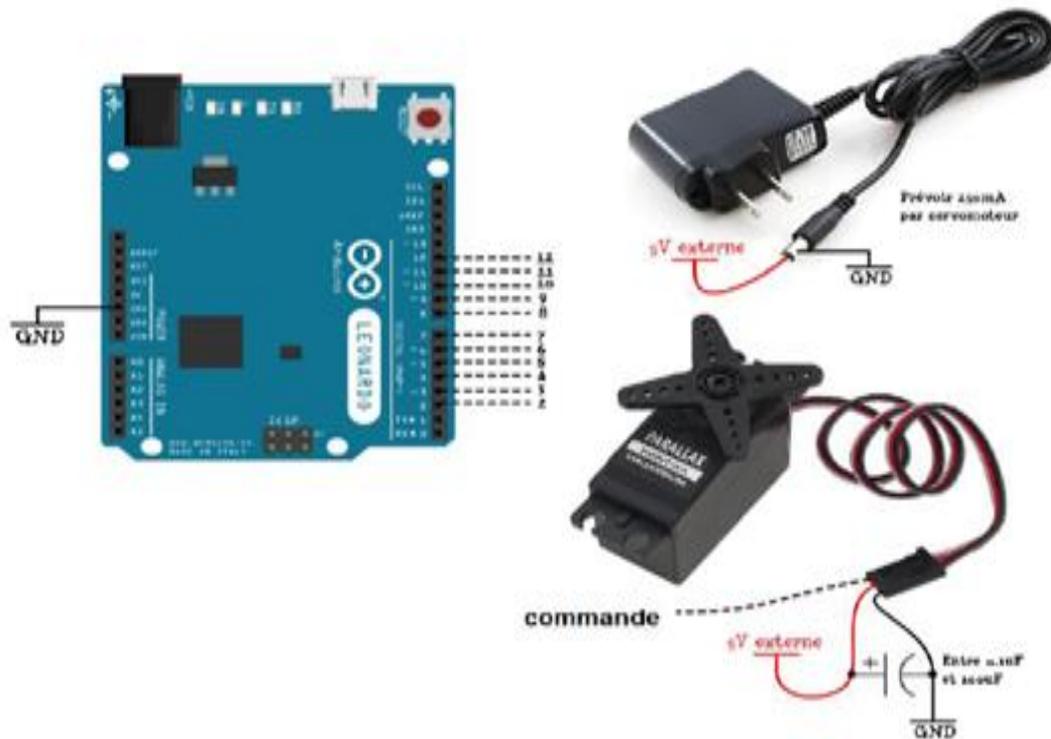


Figure II- 19: Servomoteur et arduino.

II.3. Conclusion

Dans ce chapitre, on a étudié le fonctionnement de la carte arduino et quelques types de cette carte, on a aussi on a identifié les types de commande possibles sur un Brait de robot, et à la fin on a citiez les types des moteurs, sachant que nous avons choisi la carte arduino UNO et le SERVO MOTEUR pour notre réalisation, ce que nous allons voir sur les chapitres suivants.

III. Introduction

Les chapitres précédents ont porté sur l'étude des principaux constituants d'un robot qui représentent les piliers de la réalisation de notre bras.

Dans ce chapitre, nous allons présenter la partie mécanique de notre bras manipulateur ainsi que la structure et les segments du bras. Dans la partie commande, nous allons expliquer comment commander le bras manipulateur et ses mouvements par l'utilisation de modèle géométrique direct et inverse en utilisant la méthode de **Denavit-Hartenberg**. Puis nous allons présenter la réalisation de la télé opération sur le robot que nous avons réalisé.

III.1. La description du robot

Notre robot est composé de quatre éléments essentiels, qui se représentent dans la suite :

La base de robot : la base de robot contient un moteur au centre qui permet à la base de tourner autour de lui avec un angle de π .

Segment 1 : le segment 1 est un bras fixé dans un moteur qui représente une articulation, ce dernier est placé dans la base du robot, qui permet à ce segment de tourner avec un angle de π , la longueur de segment est : $L_1 = 17$ cm

Segment 2 : Le segment 2 est un bras fixé dans un moteur, ce deuxième moteur est placé dans le premier segment, le deuxième segment tourne avec un angle de π , et sa longueur est : $L_2 = 11.5$ cm.

Organe terminale :

c'est un fer à soudeuse de longueur de 4cm.



Figure III- 1: Robot industriel

III.2. Les modèles mathématiques de robot

Concernant la modélisation nous avons utilisé la méthode de Denavit-Hartenberg qui à travers laquelle on peut faciliter et proximité ces dernières dans la réalité.

III.2.1. Modèle géométrique direct de robot

Dans le **M.G.D** nous recherchons le vecteur de position de l'organe terminal avec la connaissance de l'angle. Le tableau suivant représente les paramètres géométriques du robot

Tableaux 1 : Les paramètres géométriques du robot

<i>Articulation</i>	1	2	3	4
α_j	0	90°	0	0
d_j	0	0	$D3$	$D4$
θ_j	θ_1	θ_2	θ_3	0
R_j	0	0	0	0

Paramètres de liaison :

\mathbf{d}_n la longueur de la liaison.

α_n : l'angle de torsion entre les axes d'articulation.

Paramètres communs:

θ_n : l'angle entre les liaisons.

\mathbf{R}_n : la distance entre les liaisons.

Le symbolique modèle géométrique direct pour le robot (Figure III.1).

D'après le tableau 1 et en utilisant les paramètres de **Denavit-Hartenberg** modifiées, nous obtenons les matrices de transformation $T_{j-1,j}$ comme suite :

$$T_{0,1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{1,2} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2,3} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & D3 \\ 0 & 0 & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{3,4} = \begin{bmatrix} 1 & 0 & 0 & D4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,4} = T_{0,1} \times T_{1,2} \times T_{2,3} \times T_{3,4}$$

$$T_{0,4} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & D3 \\ 0 & 0 & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & D4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On pose: $C23 = \cos(\theta_2 + \theta_3)$

ET $S23 = \sin(\theta_2 + \theta_3)$

$$T_{0.4} = \begin{bmatrix} c1c2c3 & -c1s2s3 & -c1c2s3 & D4c1c23 + D3c1c2 \\ s1c2c3 - s1s2s3 & -s1c2s3 - s1s2c3 & -c1 & D4s1c23 + D3s1c2 \\ s2c3 + c2s3 & -s2s3 + c2c3 & 0 & D4s23 + D3s2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0.4} = \begin{bmatrix} c1c23 & -c1s23 & -c1c2s3 & D4c1c23 + D3c1c2 \\ s1c23 & -s1s23 & -c1 & D4s1c23 + D3s1c2 \\ s23 & -s2s3 + c2c3 & 0 & D4s23 + D3s2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pour les coordonnées de position P_x , P_y et P_z nous obtenons la dernière colonne et le premier trois lignes :

$$P_x = D4c1c23 + D3c1c2$$

$$P_y = D4s1c23 + D3s1c2$$

$$P_z = D4s23 + D3s2$$

III.2.2. Modèle géométrique inverse de robot

Entre les méthodes qui permet de trouver les solutions

Type 1: $Xr_i = Y$

Type 2: $XS\theta_i + XC\theta_i = Z$

Type 3: $\begin{cases} X_1 S\theta_i = Y_1 \\ X_2 C\theta_i = Y_2 \end{cases}$

$$\text{Type 4: } \begin{cases} X_1 r_j S \theta_i = Y_1 \\ X_2 r_j C \theta_i = Y_2 \end{cases}$$

$$\text{Type 5: } \begin{cases} X_1 S \theta_i = Y_1 + Z_1 r_j \\ X_2 C \theta_i = Y_2 + Z_2 r_j \end{cases}$$

$$\text{Type 6: } \begin{cases} W S \theta_i = X C \theta_i + Y S \theta_i + Z_1 \\ W C \theta_i = X S \theta_i - Y C \theta_i + Z_2 \end{cases}$$

$$\text{Type 7: } \begin{cases} W_1 C \theta_j + W_2 S \theta_j = X C \theta_i + Y S \theta_i + Z_1 \\ W_1 S \theta_j - W_2 C \theta_j = X S \theta_i - Y C \theta_i + Z_2 \end{cases}$$

$$\text{Type 8: } \begin{cases} X C \theta_i + Y C (\theta_i + \theta_j) = Z_1 \\ X S \theta_i + Y S (\theta_i + \theta_j) = Z_2 \end{cases}$$

On a:

$$\begin{cases} P_x = D_4 c_1 c_2^3 + D_3 c_1 c_2 \\ P_y = D_4 s_1 c_2^3 + D_3 s_1 c_2 \\ P_z = D_4 s_2^3 + D_3 s_2 \end{cases}$$

Donc :

$$\begin{cases} P_x^2 = c^2 \cdot 1 [d^2 4 c^2 2^3 + d^3 3 c^2 2 + 2 d 4 c 2^3 c 2] \\ P_y^2 = s^2 \cdot 1 [d^2 4 c^2 2^3 + d^3 3 c^2 2 + 2 d 4 c 2^3 c 2] \end{cases}$$

Et:

$$\sqrt{P_x^2 + P_y^2} = d 4 c 2^3 + d 3 c 2$$

- **Remarque:** les deux équations de type 8

$$Pz = d_4 s_2^3 + d_3 s_2$$

On pose :

$$\begin{cases} X=d_3 \text{ et } \sqrt{Px^2 + Py^2} = Z_1 \\ Y=d_4 \text{ et } Pz = Z_2 \end{cases}$$

Donc θ_3 :

$$C\theta_3 = (Z_1^2 + Z_2^2 - X^2 - Y^2) / 2XY$$

$$\theta_3 = \arctan(\sqrt{1 - C^2 \theta_3}, C\theta_3) \quad \text{avec: } \sqrt{1 - C^2 \theta_3} = S\theta_3 .$$

$$S\theta_2 = \frac{Z_2 B_1 - Z_1 B_2}{B_1^2 + B_2^2} \quad \text{et} \quad C\theta_2 = \frac{Z_2 B_1 + Z_1 B_2}{B_1^2 + B_2^2}$$

Avec :

$$\begin{cases} B_1 = X + Y C\theta_3 \\ B_2 = Y S\theta_3 \end{cases}$$

Donc :

$$\theta_2 = \arctan(S\theta_2, C\theta_2)$$

III.3. Les outils de développement

III.3.1. L'environnement de la programmation

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information. Ce programme appelé IDE Arduino.

III.3.2. Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation. La programmation de l'Arduino est basée sur langage C.

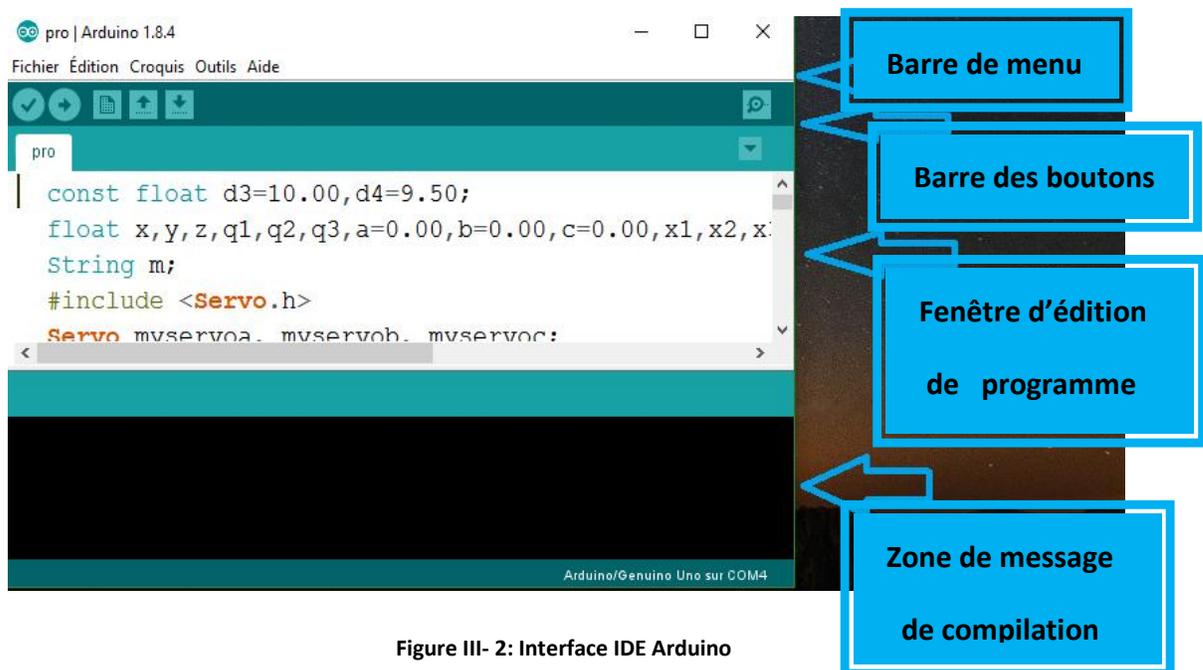


Figure III- 2: Interface IDE Arduino

III.3.2.1. Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 4) comme montre les deux figures suivantes.

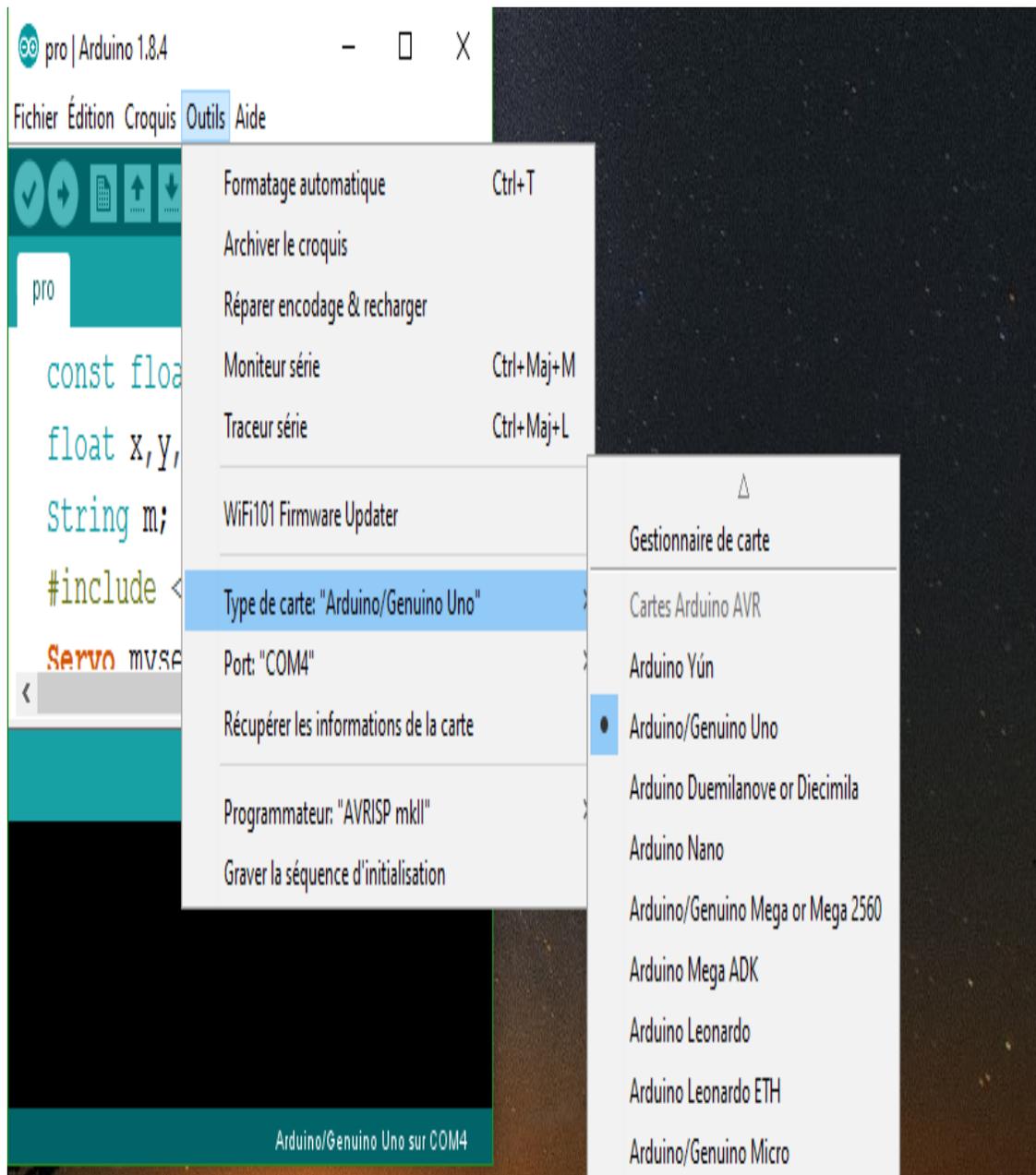


Figure III- 3: Le choix du type de la carte Arduino uno

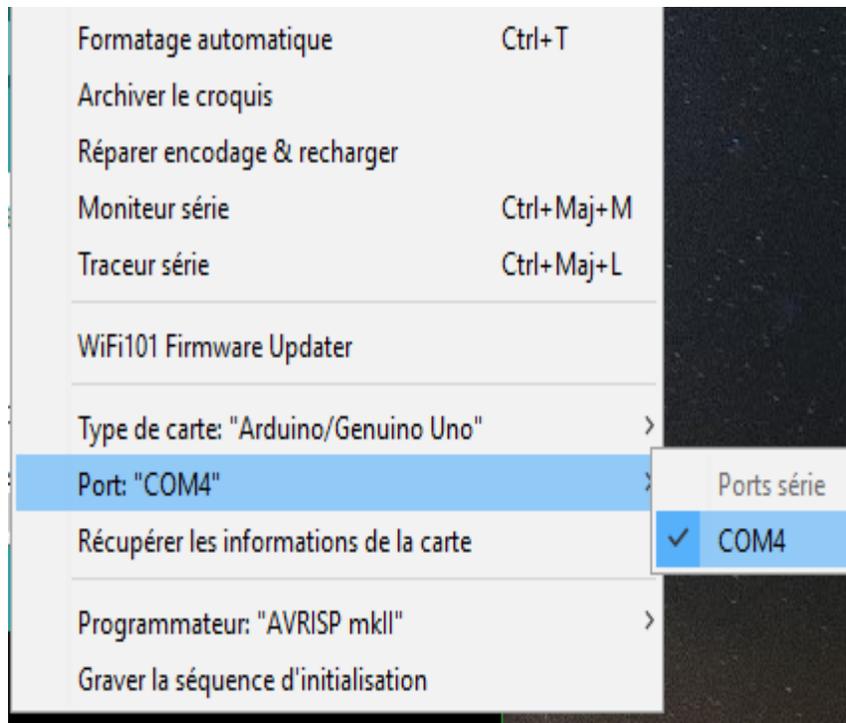


Figure III- 4: la sélection du numéro de port USB (COM 4)

III.3.2.2. Description du (IDE)

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

III.3.2.3. Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

1 /* programme de command DC moteur avec Smartphone via Bluetooth-----

2 /*et fait également clignoter la diode de test de la carte-----

3/ *-----

III.3.2.4. Définition des variables

Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la 3^{ème} sortie numérique ; cette variable doit être définie et nommée ici moteur pin 3 ; la syntaxe est pour désigner un nombre entier est int.

```
4 int moteur 1 = 3; // mettre le moteur au pin 3-----
```

III.3.2.5. Configuration des entres et des sorties void setup ()

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on va configurer moteur pin en sortie ; pin mode (nom, état) est une des quatre fonctions relatives aux entrées – sorties numériques.

```
5 /*void setup () {-----
```

```
6 /* mettre le moteur 1 comme sortie:-----
```

```
7 /*pinMode (motor 1, OUTPUT) ; // lorsque le pin 3 est activé le moteur tourne-----
```

```
8} -----
```

III.3.2.6. Programmation des interactions void loop

Dans cette boucle, on définit les opérations à effectuer dans l'ordre digital write (nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- Delay (temps en mili-seconde) est la commande d'attente entre deux instructions.
- Chaque ligne d'instruction est terminée par un point-virgule.
- Ne pas oublier les accolades qu'encadre la boucle.

```
9/* void loop () {-----
```

```
10/* digital write (motor 1, HIGH); -----
```

```
11 delay (3000) -----
```

```
12 digital Write(motor 1, LOW); -----  
13 delay (1000) -----  
14} -----
```

III.3.3. Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d’injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L’exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d’alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

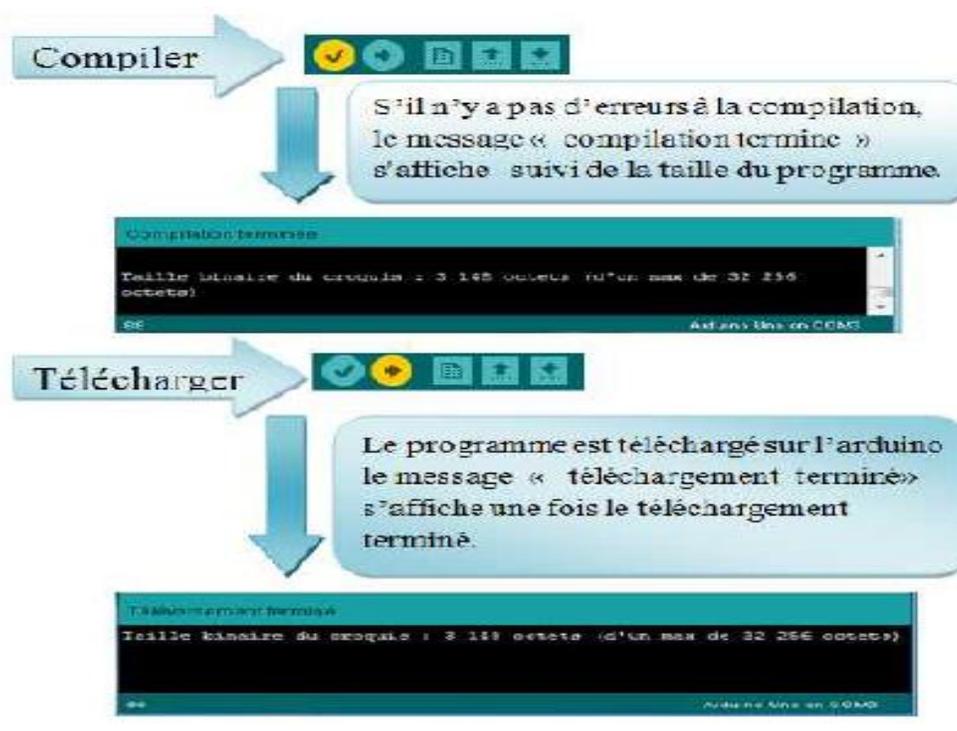


Figure III- 5: Les étapes de téléchargement du code

III.4. La réalisation pratique d'un robot industriel contrôlé par ordinateur

Notre robot est de genre industriel et son rôle se présente par le soudage des objets.

Dans cette section, la commande est appliquée par la déclaration des positions à l'aide des coordonnées cartésiennes et on les injecte dans le moniteur série et ce dernier nous permet de convertir ces coordonnées en angles et par conséquent, l'arduino reçoit toutes ces informations et lui les transmet au cerveau des moteurs comme des ordres qui nous apparaissent à la fin par le mouvement requis.

III.4.1. Espace de travail



Figure III- 6: Les Positions 1 et 2 du robot

- **Dans la position 1** : la base du Robot est en état initial avec un angle 0.
- **Dans la position 2** : la base du robot dans l'état maximal, il fait un demi-cercle, il tourne avec un angle π .

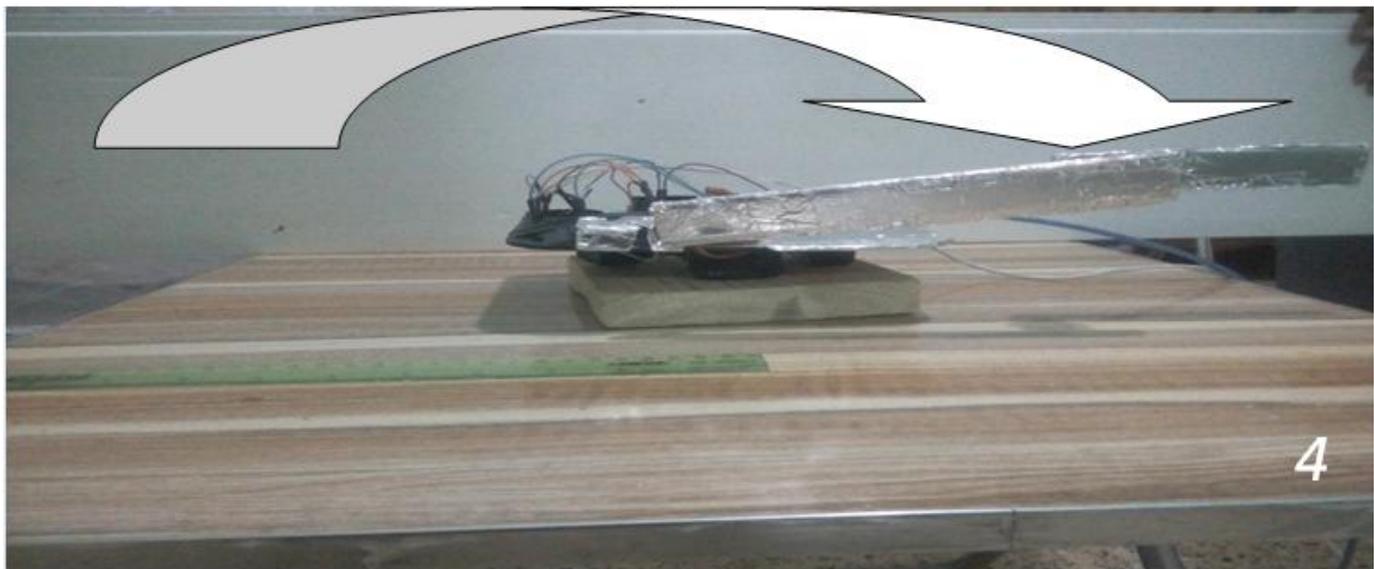
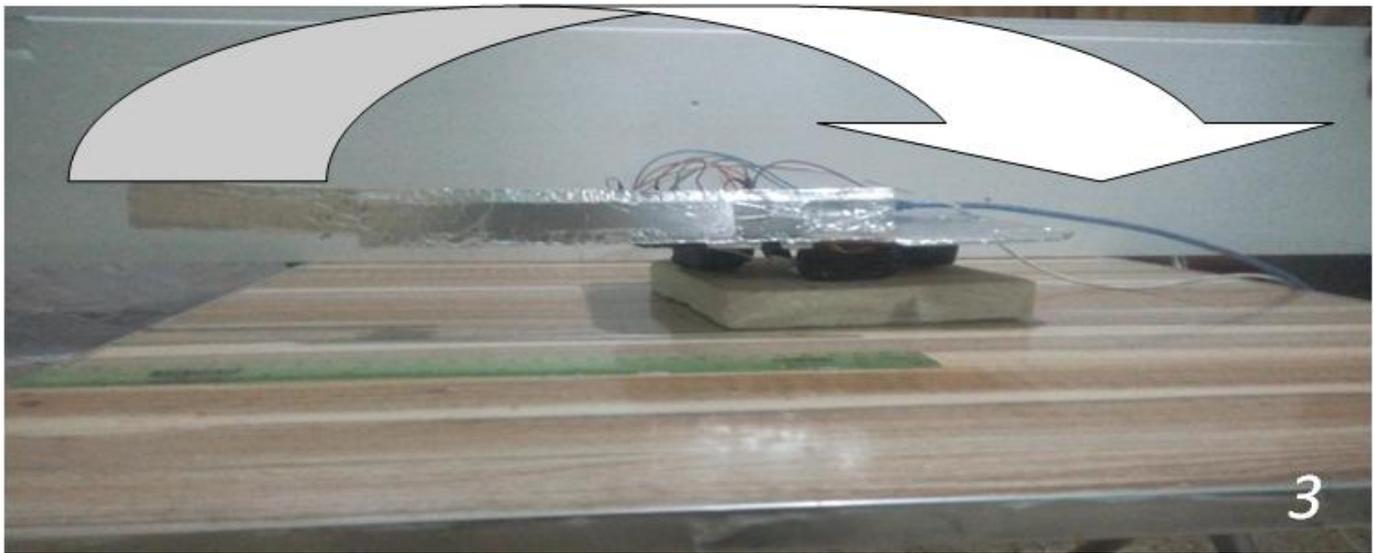


Figure III- 7: les positions 3 et 4 du robot

- **Dans la position 3** : le bras de robot est aussi en l'état initial avec un angle 0
- **Dans la position 4** : le bras de Robot fait un demi-cercle avec un angle de π

Finalement on a conclu que la rotation du robot (la rotation de la base et la rotation du bras) fait une demi-sphère pour tourner à son état final, le rayon de la sphère est la longueur du bras.

III.4.2. Mode de travail

Pour tester le fonctionnement de notre robot on a déjà choisi des coordonnées qui on veut les atteindre avec ce dernier, donc on a fait des captures de différents positions par les scénarios suivantes :

- **Scénario 01 :**

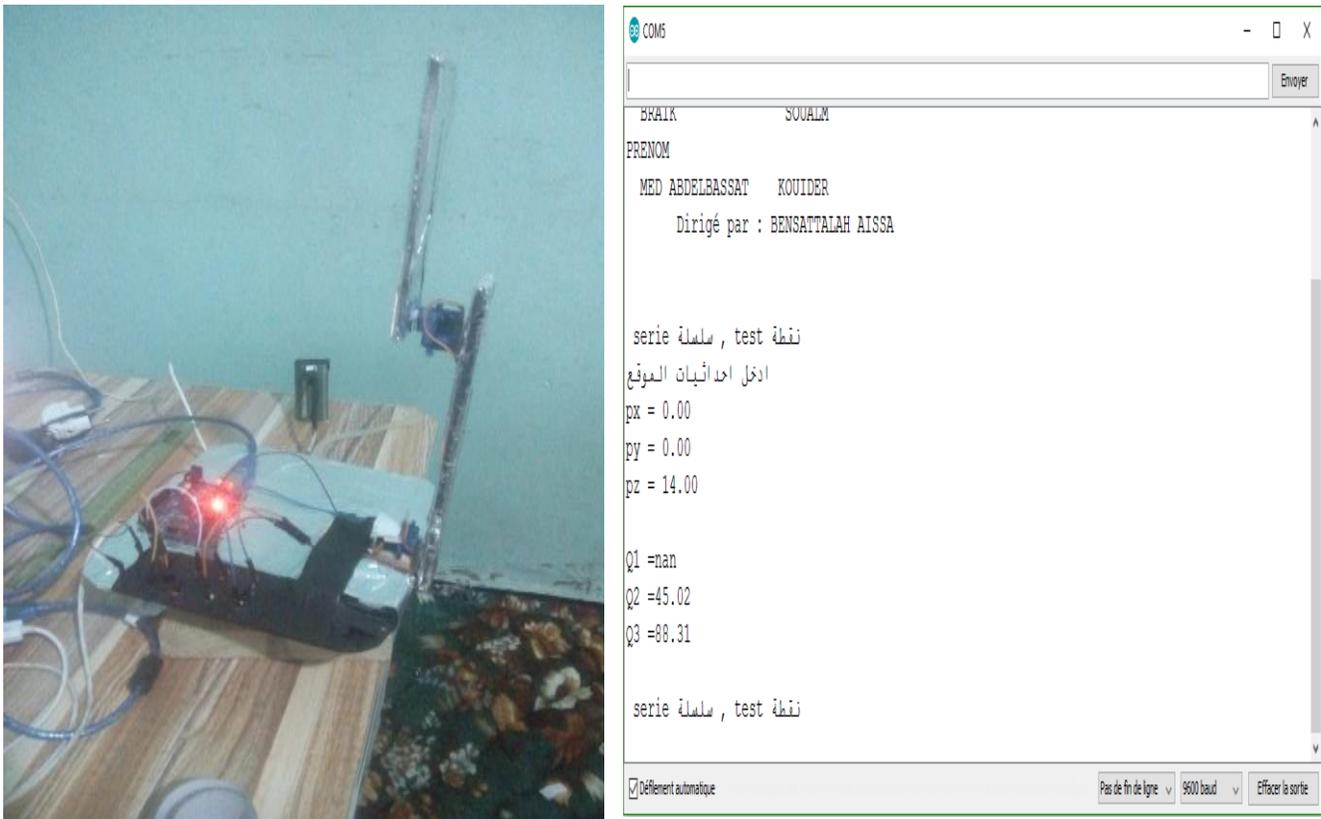


Figure III- 8: Première position

- Scénario 02 :

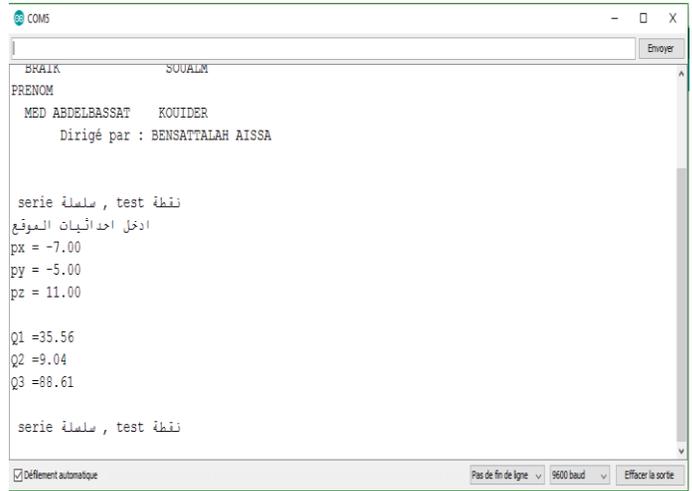
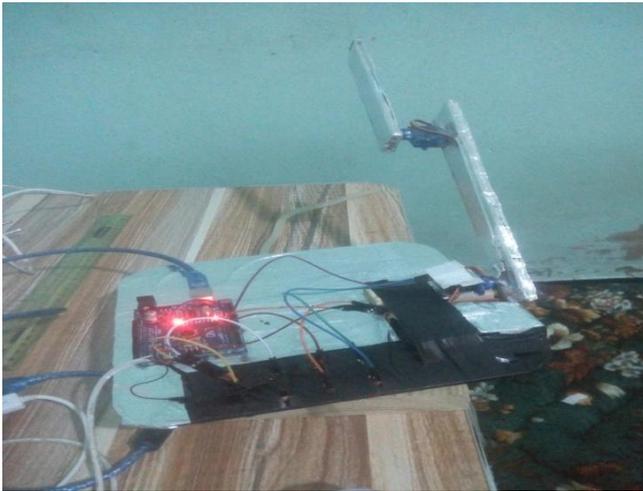


Figure III- 9: Deuxième position

- Scénario 03 :

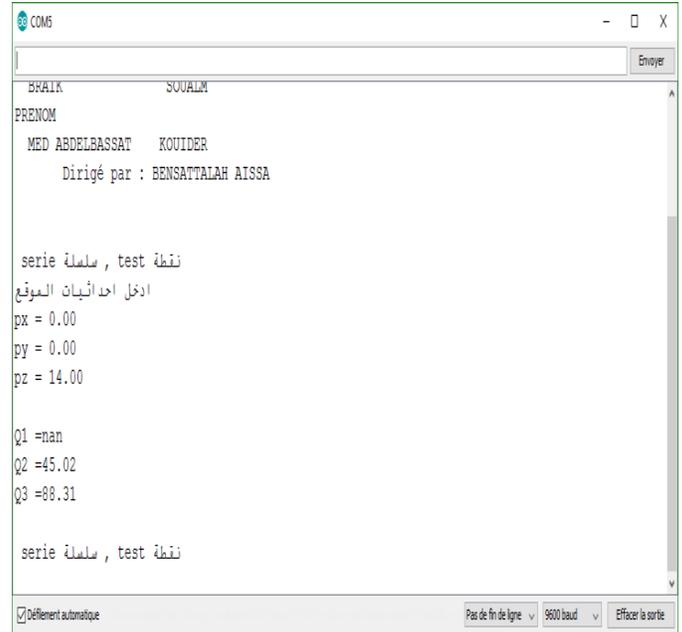
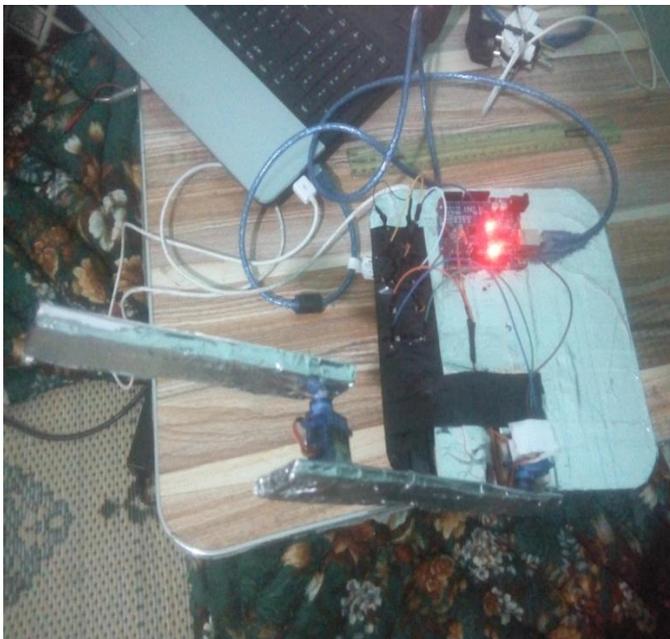


Figure III- 10: troisième position

- **Scénario 04 :**

On a donné une série des points pour faire les opérations voulues, la figure ci-dessus montre cette opération.

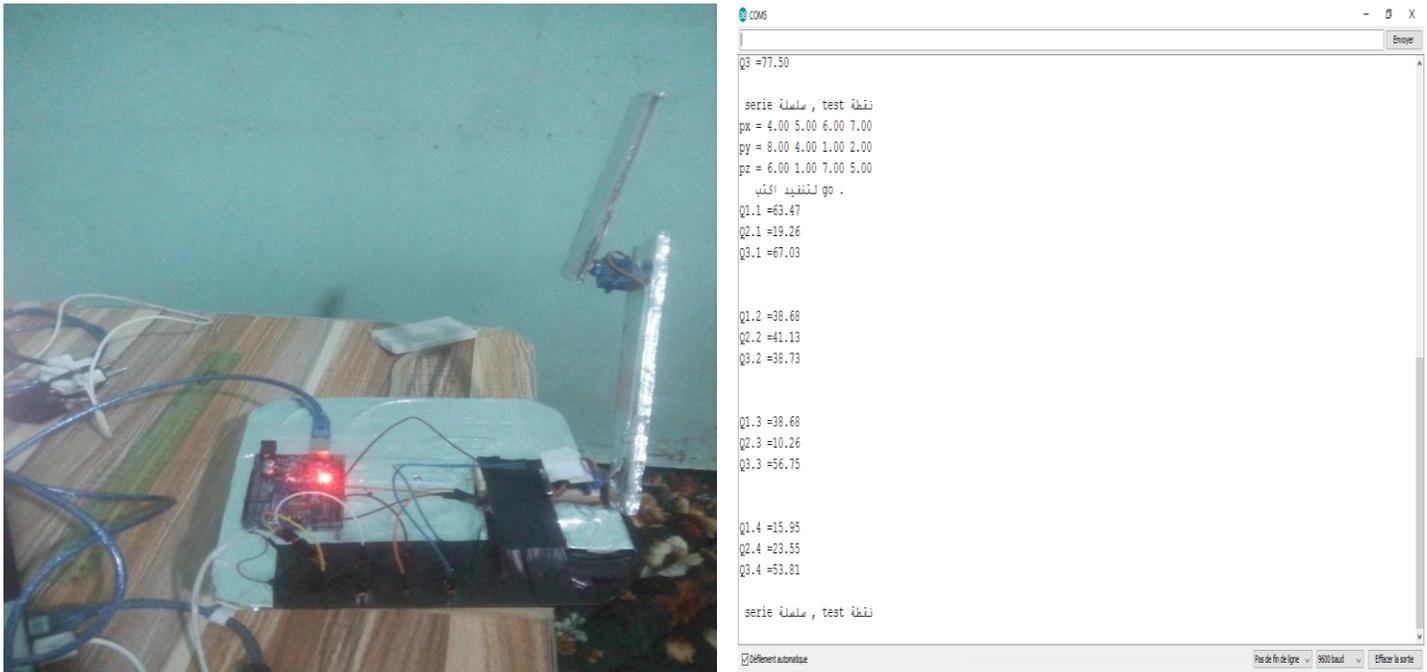


Figure III- 11: quatrième Position

III.4.3. Description du programme

Notre programme représente le système d'exploitation de robot. Dans cette section nous détaillons notre programme.

Au début nous avons déclaré les variables que nous voulons les utiliser dans la fonction void loop (),

```
const float d3=10.00,d4=9.50;
float x, y, z, q1, q2, q3, a=0.00, b=0.00, c=0.00, x1, x2, x3, x4, y1, y2, y3, y4, z1, z2, z3, z4;
String m;
#include <Servo.h>
Servo myservoa, myservob, myservoc;
...

```

#include <Servo.h> : représente la bibliothèque de cerveau moteur.

Dans cette étape nous avons arrivé à la fonction void setup ().

```
void setup() {
  Serial.begin(9600);

  myservoa.attach(3);
  myservob.attach(5);
  myservoc.attach(6);

  myservoa.write(a);
  myservob.write(b);
  myservoc.write(c);
}
```

La fonction **Serial.begin(9600)** permet d'échanger les données entre le micro –ordinateur et la carte Arduino et par conséquent on peut ouvrir l'interface de monétaire serial. La valeur 9600 représente la vitesse de transmission des trames.

My servo.attach () : attach () c'est une fonction qu'elle responsable de broncher le servo avec les pines d'Arduino.

My servoa.write () : c'est une fonction pour donner l'ordre pour exécuter la rotation du moteur.

```
void loop() {
  Serial.println( " serie سلسلة , test نقطة " );
  while(Serial.available()==0){}
  m=Serial.readString();
  Serial.flush();
  if(m=="test")
  ,
}
```

Dans cette étape nous avons expliqué le contenu de la fonction `void loop ()` :

Serial.println () : c'est une fonction d'affichage nous avons utilisé pour afficher les coordonnées tapées ou un ensemble des coordonnées tapées.

While (Serial.available () ==0) {} : **while** c'est une boucle conditionné et la condition utilisé dans le **While Serial.available () ==0** signifie que si ne recevrait pas dans le serial alors on va exécuter le programme dans le boucle {}.

m=Sreial.readString () : m est une variable de type string recevait **Sreial.readString ()** ;

La fonction **Sreial.readString ()** prend tout ce qui écrit sur le clavier et de stocker dans la variable m.

Serial.flush () : cette fonction permet de décharger la trame de transmission pour utiliser la fonction **Sreial.readString ()** encore dans une autre variable.

If (m== 'test ') : si la condition **m== 'test '** est vérifiée donc on va exécuter le programme a l'intérieur.

```
Serial.println("ادخل احداثيات الموقع");
Serial.print("px = ");
while (Serial.available() == 0) {}
x=Serial.parseFloat();
Serial.println(x);
Serial.flush();
delay(1000);
Serial.print("py = ");
while (Serial.available() == 0) {}
delay(1000);
y=Serial.parseFloat();
Serial.println(y);
Serial.flush();
delay(1000);
Serial.print("pz = ");
while (Serial.available() == 0) {}
delay(1000);
z=Serial.parseFloat();
Serial.flush();
Serial.println(z);
Serial.println("مغز أكتب");
while (Serial.available() == 0) {}
m=Serial.readString();
```

X= **Serial.parseFloat** () : m est une variable de type float (réel) reçoit **Serial.parseFloat**

La fonction **Serial.parseFloat** () prend seulement des nombres réels c'est-à-dire ce qui écrit sur le clavier et stocké dans x.

Remarque :

Si nous écrivons seulement les lettres stockées 0 dans x. La même chose pour charger Y et Z ; nous vérifions actuellement la condition suivante `if (m== 'mgi')`.

```
float q=atan(y/x);
q1=(q*180)/3.14 ;
Serial.print("Q1 =");
Serial.println(q1);
delay(1000);

float h1=sqrt(sq(x)+sq(y));
float c3=(sq(h1)+sq(z)-sq(d3)-sq(d4))/(2*d3*d4);

float q6=atan((sqrt(1-sq(c3)))/c3);
q3=abs((q6*180)/3.14) ;
float b1=d3+d4*c3;
float b2=d4*sqrt(1-sq(c3)) ;
float s2=(b1*z-b2*h1)/(sq(b1)+sq(b2));
float c2=(b1*z+b2*h1)/(sq(b1)+sq(b2));

float q4 =atan(s2/c2);
q2=abs((q4*180)/3.14) ;
Serial.print("Q2 =");
Serial.println(q2);
delay(1000);
Serial.print("Q3 =");
Serial.println(q3);
```

Dans ces lignes de programmation, nous avons introduit les mathématiques obtenues par la méthode **D-H**.

```

if (q1>a)

    {for (int i=a; i<q1; i++)
      { myservoa.write(i);delay(30);}

      a=q1;
      delay(2500);}

else

    {for (int i=a; i>q1; i--)
      {myservoa.write(i); delay(30);}

      a=q1;
      delay(1500);}

```

Dans cette étape nous avons montré comment faire tourner les servo moteurs :

If (q1>a) : si l'angle de $q1 > a$, le traitement va exécuter à l'intérieur de la boucle for (**pour**), le moteur est démarré pour tourner vers la gauche (**direction positive**) avec une vitesse 1 degré à chaque 30 ms.

Else : si la condition n'est pas vérifiée, le moteur est démarré pour tourner mais vers la droite (**direction négative**) avec une vitesse 1 degré à chaque 30 ms.

Enfin, le même programme va être répété pour $q2$ et $q3$.

III.5. La télé opération de robot

Une fois nous avons terminé la réalisation du robot avec les tests de vérification des positions nous passons à la deuxième étape c'est la réalisation de la télé opération, la figure suivante présente les mécanismes de la programmation de la télé opération. Nous avons utilisé les technologies de JavaScript

et HTML5 pour la programmation de l'interface de commande de robot à distance donc cette fenêtre nous a permet de commander à distance le robot.

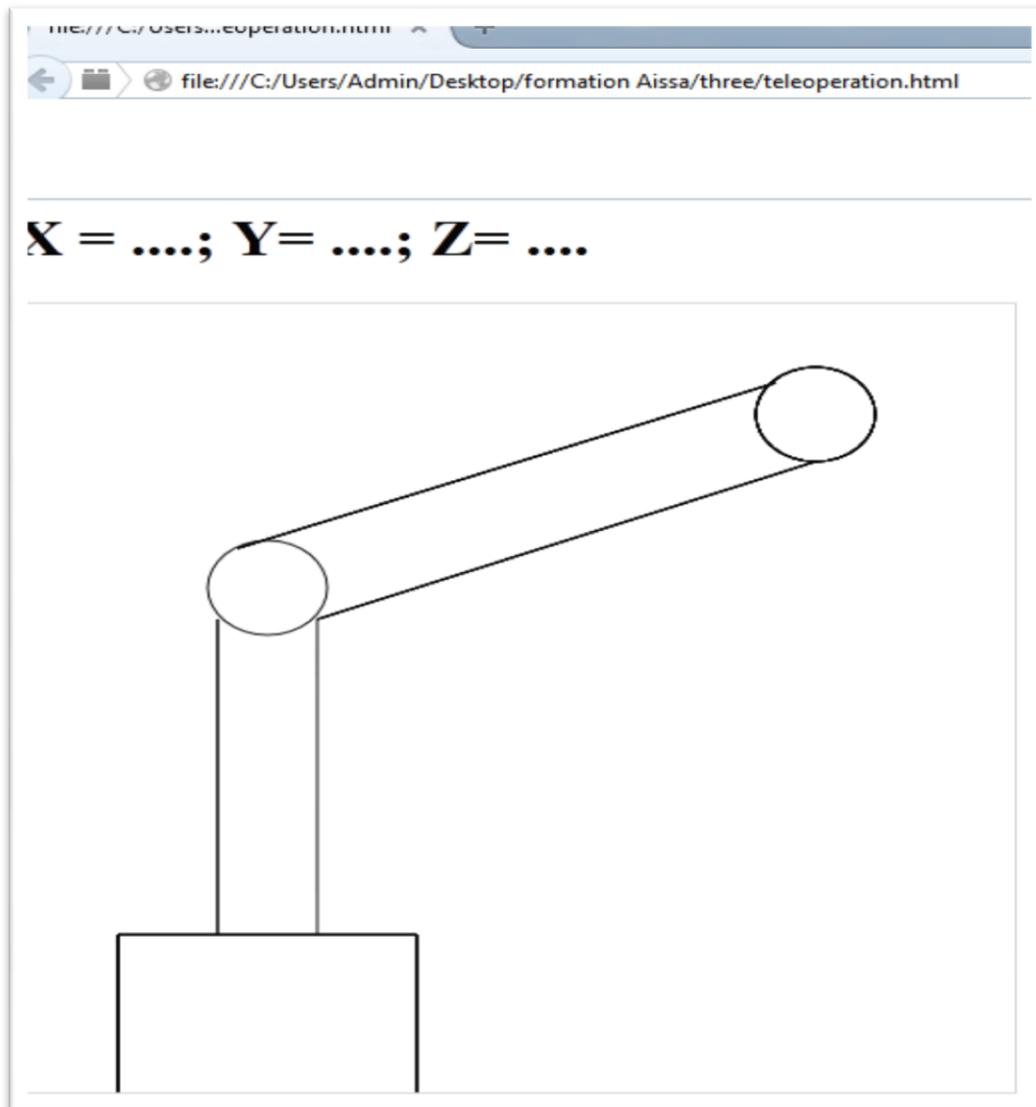


Figure I- 20:Télé opération de robot par la technologie Web (Javascript et HTML)

III.6. Conclusion

Dans ce chapitre nous avons d'abord extrait l'équation mathématique du mouvement du robot en utilisant la méthode « H.D » ; ce qui nous a aidé dans la programmation par « Arduino » en préparant mécaniquement le robot et en installant ses parties principales. Nous avons apporté des résultats satisfaisants et une relation étroite entre la modélisation mathématique et la programmation.

Enfin nous avons appliqué le mécanisme de la télé opération sur le robot que nous avons réalisé par l'utilisation des technologies de programmation Web.

Conclusion générale

Le but de notre projet est l'utilisation de logiciel et la carte électronique Arduino pour manipuler un robot industriel à trois articulations puis d'appliquer sur ce robot la télé opération.

Dans ce travail nous avons étudié, d'abord nous avons présenté des robots industriels de leurs différentes types et usages, et nous avons ensuite étudié la manière de la programmation en parlant de la carte Arduino, nous avons donc étudié les types des cartes, leurs composantes internes et leurs utilisations notamment l'ARDUINO UNO c'est la fabrication de robot de soudage avec programmation.

Ce travail trouve quelques contraintes telles que la recherche des composants électroniques.

Cette étude nous a permis d'enrichir nos connaissances pratiques dans le domaine de la robotique industrielle, et nous a donné l'occasion de travailler par la programmation C, la modélisation et l'utilisation des modèles mathématique dans un cas réel, les techniques nécessaires pour l'utilisation de l'Arduino et le microcontrôleur dans les montages électroniques.

Dans ce travail nous avons conclu que la partie de programmation occupe 80% de nos efforts tant que la partie électronique ne demande pas un grande d'effort. Donc l'informatique industrielle c'est la programmation.

Ce projet pour avoir servir de base pour commander d'autres systèmes ou d'autres robots ayant différentes tâches et application selon le besoin.

Et aussi comme une perspective importante est d'utiliser intelligence artificielle (IA) dans la robotique industrielle.