

الجمهورية الجزائرية الديمقراطية الشعبية

République algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة عين تموشنت بلحاج بوشعيب

Université -Ain-Temouchent- Belhadj Bouchaib

Faculté des Sciences et de la Technologie

Département d'Electrotechnique



Projet de fin d'études de Master académique en :

Domaine : SCIENCES ET DE LA TECHNOLOGIE

Filière : Electromécanique

Spécialité : Electromécanique

:

**Thème : Etude et contrôle d'un bras Manipulateur à base de carte  
Arduino piloter par interface graphique en labview**

Présenté Par :

1) Mr Hammadi Abdel kader 15/10/1996 Ain temouchent

2) Mr Beddou Hachem 24/01/1996 Ain temouchent

Devant le jury composé de :

Dr ATTOU Amine

UAT.B.B (Ain Temouchent) Président

Dr ZEGAI Mohamed Amine

UAT.B.B (Ain Temouchent) Examineur

Dr LAKHDARI Lahcen

MAB UAT.B.B (Ain Temouchent) Encadrant

Année universitaire 2021/2022

---



## *Remerciements*

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

Nous tenons à exprimer sincère remerciement et notre reconnaissance à Mr Lakhdari Lahcen pour son encadrement et sa précieuse disponibilité qu'il a bien voulu nous accorder. Son expérience académique, sa rigueur intellectuelle et ses conseils dans le but de réaliser ce travail, nous ont permis de nous développer et de réussir ce projet.

Nous remercions très respectueusement Mr Attou Mohamed et Mr Zagai, de nous avoir fait l'honneur de juger notre modeste travail.

On ne saura clôturer ces remerciements sans exprimer toute notre affection pour les enseignants de l'Université de Ain Temouchent, qui ont bien voulu partager leurs savoir durant nos études.

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

## Résumé

Ce travail s'intéresse à l'étude et contrôle d'une commande d'un bras manipulateur à 4 degrés de liberté commandée par la carte Arduino, en créant une interface graphique utilisant le logiciel de LABVIEW, pour simuler les schémas et les circuits électriques on utilise :

Le logiciel ISIS, en fin en utilise le logiciel Matlab/Simulink pour tester le modèle et simuler les réponses.

**Mots clés :** Bras manipulateur, Arduino, Degré de liberté, Servomoteur, LABVIEW

## Summary

This work focuses on the study and control of a control of a manipulator arm with 4 degrees of freedom controlled by the Arduino card, by creating a graphic interface using the LABVIEW software, to simulate diagrams and electrical circuits. we use :

The ISIS software, in the end, uses the Matlab/Simulink software to test the model and simulate the responses.

**Keywords:** Manipulator arm, Arduino, Degree of freedom, Servomotor, LABVIEW

## ملخص

يركز هذا العمل على دراسة والتحكم في ذراع المتلاعب عند 4 درجات من الحرية التي يسيطر عليها الاردوينو , من خلال إنشاء واجهة رسومية باستخدام برنامج لابفيو لمحاكاة الرسوم البيانية والدوائر الكهربائية نستخدم:

برنامج ازييس في النهاية، نستخدم برنامج ماثلاب/سيمولينك لاختبار النموذج ومحاكاة الاستجابات

**الكلمات المفتاحية :** ذراع مناور ، اردوينو

، درجة الحرية ، محرك آلي LABVIEW



# Sommaire

<b>Sommaire</b>	I
<b>Liste des Figures</b>	II
<b>Liste des Tableaux</b>	III
<b>Notations Et Abréviations</b>	IV
<b>Introduction générale</b>	1
<b>Chapitre I : Etat de l'art</b>	
I.1 Introduction.....	2
I.2. Définitions.....	2
I.2.1. Définition d'un robot.....	2
I.2.2. Définition de la robotique.....	2
I.3. L'histoire de la robotique.....	2
I.4. Classification des robots.....	3
I.5. Types de robots	4
I.5.1. Robots mobiles.....	4
I.5.2. Robots manipulateurs.....	4
I.5.3 Caractéristiques d'un robot.....	5
I.5.4 Robot industriel.....	6
I.6. Bras manipulateur.....	6
I.7. Bras manipulateur.....	6
I.7.1. Le porteur.....	7
I.7.2. Le porteur.....	7
I.7.2.1. Segment.....	7
I.7.2.3. Effecteur.....	7
I.7.2.4. la base.....	8
I.7.2.5. L'organe terminal.....	8
I.8. Constituants d'un robot.....	8
I.8.1. Les actionneurs.....	9
I.8.2. Le système mécanique articulé (S.M.A.).....	9
I.8.3. Les articulations.....	9
I.8.3.1. Les articulations prismatiques.....	9
I.8.3.2. Les articulations rotoïdes.....	9
I.8.4. Organe terminal.....	9
I.8.5. Les capteurs.....	10
I.8.6. La commande.....	10
I.9. Classification des robots.....	10
I.9.1. Point de vue fonctionnel.....	10
I.9.1. Point de vue fonctionnel.....	10
I.9.1.1. Manipulateur à commande manuelle.....	10
I.9.1.2 Manipulateur automatique.....	11
I.9.1.3 Robots programmables.....	11
I.9.1.4. Robots intelligents.....	12
I.9.2 Classification géométrique.....	12
I.9.2.1 Structure cartésienne (PPP).....	13
I.9.2.2. La structure cylindrique (RPP) ou (PRP).....	12
I.9.2.3. La structure sphérique ou polaire à axe de rotation orthogonale (RRP).....	13
I.9.2.4. La structure dite SCARA.....	14
I.9.2.5. La structure anthropomorphe (RRR).....	14
I.10. Domaines d'application de la robotique.....	15
I.10.1. Les robots industriels.....	15
I.10.2. Robots domestiques ou ménagers.....	15
I.10.3. Robots en médecine et chirurgie.....	15
I.11. Les composants d'un système robotique.....	15
I.11.1. Unité informationnelle.....	15
I.11.2. Unité opérationnelle.....	16

I.11.3. La structure mécanique articulée.....	16
I.11.3.1. Structure mécanique articulée à chaîne cinématique simple.....	16
I.11.3.2. Structure mécanique articulée à chaîne cinématiques fermée....	16
I.12. Conclusion.....	17
<b>Chapitre II : modélisation des bras manipulateurs</b>	
II.1. Introduction.....	18
II.2. Modélisation.....	19
II.2.1. Repères et référentiels.....	19
II.2.1.1. Coordonnées cartésiennes.....	19
II.2.1.2. Coordonnées cylindriques.....	20
II.2.1.3. Coordonnées sphériques .....	21
II.2.2. Position et orientation d'un solide.....	22
II.2.3. Modèle géométrique direct.....	22
II.2.3.1. Paramètre de Denavit-Hartenberg (DH).....	23
II.2.3.2. Modélisation géométrique direct de notre simulation .....	25
II.2.3.3. Identification des paramètres de DH de notre bras manipulateur...	25
II.2.4. Le modèle géométrique inverse (MGI).....	27
II.2.4.1. Le modèle géométrique inverse de notre réalisation.....	27
II.2.5. Modélisation des actionneurs électriques.....	29
II.2.6. Modélisation cinématique direct.....	31
II.2.7. Modèle cinématique inverse.....	35
II.3. Conclusion.....	36
<b>Chapitre III Programmation et interface graphique</b>	
III.1. Introduction.....	37
III.2. Présentation de la carte Arduino.....	37
III.2.1. La carte Arduino.....	38
III.2.2. Les caractéristiques de la carte Arduino.....	39
III.2.3. L'environnement de programmation Arduino (IDE).....	42
III.3. Les actionneurs (servomoteurs).....	43
III.3.1. Les paramètres techniques des servomoteurs.....	45
III.4. Présentation du logiciel ISIS (Intelligent Schematic Input System).....	45
III.5. Définition de LabVIEW .....	46
III.6. Description du programme.....	46
III.6.1. Le diagramme.....	47
III.6.2. Les Outils.....	48
III.6.3. Les palettes .....	49
III.6.4. Les types de données utilisés dans LabVIEW.....	52
III.6.5. Les Bibliothèques.....	53
III.6.6. Logiciel LabVIEW en acquisition de données.....	53
III.8. Conclusion.....	54
<b>Chapitre IV Commande du Bras Manipulateur</b>	
IV.1 Introduction.....	55
IV.2 Présentation de la carte Arduino.....	55
IV.2.1. La carte Arduino UNO.....	55
IV.3 Présentation du logiciel ISIS.....	56
IV.4 Schéma de circuit sur ISIS.....	58
IV.5 Arduino IDE.....	58
IV.6 Logiciel « LabVIEW ».....	59
IV.6.1. Programmation et interface graphique.....	60
IV.6.2. Interface de commande sur LabVIEW.....	60
IV.6.2.1. Panneau avant (front panel).....	60
IV.6.2.2. Fonctionnement.....	61
IV.6.2.3. Block de diagramme.....	61
IV.6.2.4. Description du programme.....	61
IV.7 Logiciel Virtual Serial Port Driver.....	62
IV.7.1. Configuration VSPK .....	62
IV.8 Simulation du bras manipulateur avec la commande PD.....	63
IV.8.1. Le régulateur PD.....	63
IV.8.2. Commandes Dynamiques De Robot Manipulateur.....	64

IV.8.3 Simulation de bras manipulateur en temps réel.....	66
IV.9 Conclusion .....	69
<b>Conclusion Générale</b>	70
<b>Références Bibliographiques</b>	71



## Liste des Figures

Figure.I.1 : Robot mobile.....	1
Figure.I.2 : Bras Manipulateur.....	5
Figure.I.3: Architecture d'un robot.....	6
Figure I.4 : différentes structures sont montrées.....	8
Figure I.5 : Manipulateur à commande manuelle.....	10
Figure.I.6 : Manipulateur automatique.....	10
Figure.I.7: Robot programmable.....	11
Figure.I.8 : Robot intelligent.....	11
Figure.I.9: Structure PPP.....	12
Figure.I.10: Structure RPP.....	12
Figure.I.11: Structure RRP.....	13
Figure.I.12: Structure SCARA.....	13
Figure.I.13: Structure RRR.....	14
Figure II-1 : Le système de coordonnées cartésiennes.....	17
Figure II-2 : Le système de coordonnées cylindriques.....	18
Figure II-3 : Le système de coordonnées sphériques.....	19
Figure II-4 : Position d'un solide dans l'espace et son repère associé.....	19
Figure II-5 : Schématisation du lien entre le corps $i-1$ et $i$ .....	21
Figure II-6 : Placement des repères selon le modèle DH.....	22
Figure II-7 : Schéma fonctionnel d'un actionneur.....	25
Figure II-8 : Cas d'articulation prismatique.....	28
Figure II-9 : Cas d'articulation rotoïde.....	29
Figure III-1: Communication du logiciel IDE vers le matériel.....	33
Figure III-2: Composent de la carte arduino.....	34
Figure III-3: Interface Logiciel Arduino IDE.....	37
Figure III-4: Servomoteur.....	38
Figure III-5 : graphique des circuits.....	39
Figure III-6 : Fenêtre diagramme.....	41
Figure III-7 : Barre d'outils diagramme.....	41
Figure III-8: palette d'outils.....	42
Figure III-9: La palette de commande.....	43
Figure III-10 : palette de fonction.....	44
Figure III-11: fenêtre d'aide contextuelle.....	45
Figure III-12 : Les différents types de données utilisées dans l'environnement LabVIEW.....	45
Figure III-13 : Palette d'acquisition de données.....	46
Figure III-14 : DAQ assistant.....	46
Figure. IV. : La carte Arduino UNO.....	49
Figure.IV.2 : Ecran du chargement de Proteus.....	50
Figure. IV. :3 Interface ISIS.....	51
Figure.IV.4 : Environnement de développement Arduino.....	52
Figure.IV.5 : Interface de LabVIEW.....	53
Figure.IV.6 : Panneau avant.....	53
Figure.IV.7. : Block de diagramme sur LabVIEW.....	54
Figure.IV.8 : Description de block de diagramme.....	55
Figure.IV.9 : L'interface de logiciel Virtual Serial Port Driver.....	56
Figure.IV.10 : Schéma bloc de la commande P D.....	57
Figure.IV.11 : Positions et erreurs des différents segments par la loi de commande PD.....	58
Figure.IV.12 : Schéma bloc de la commande PD et le model 3D de bras manipulateur.....	59
Figure.IV.13 Positions et erreurs des différents segments par la loi de commande PD en temps réel.....	60
Figure.IV.14 Positions et erreurs des différents segments par la loi de commande PD en temps réel avec changement de repère.....	61

## Liste des Tableaux

Tableau II.1 : Types d'articulations et d'angles de rotations possibles du bras réalisé.....	22
Tableau II.2 : Tableau des paramètres de DENAVIT- HATENBERG de notre simulation....	23
Tableau III.1 : Tableau des composants de la carte arduino.....	35
Tableau III.2 : Paramètres techniques des servomoteurs utilisés.....	38

## Notations Et Abréviations

$a_i$	Distance entre $z_{i-1}$ et $z_i$ , le long de $x_i$
$\alpha_i$	Angle entre $z_{i-1}$ et $z_i$ , autour de $x_i$
$C(\theta, \dot{\theta})$	Matrice appelée courbe de Coriolis
$C(q)$	Vecteur des forces de charges gravitationnelles
$D(q)$	Matrice d'inertie liée à l'accélération
$D_c(t)$	Composante dérivée
$d_i$	Déplacement de l'articulation
$E_c(t)$	Erreur
$f_e$	Vecteur de l'effort extérieur
$G(\theta)$	Vecteur des forces de gravité
$H(q, \dot{q})$	Vecteur des forces centrifuges
$h_i$	Hauteur du centre de masse
$I_c(t)$	Composante intégrale
$I_j$	Courant de l'induit du moteur $j$
$J(q)$	Matrice jacobéenne
$J_j$	Inertie propre du moteur
$jk_d$	Constante dérivée du système
$k_i$	Constante intégrale du système
$k_p$	Constante proportionnelle du système
$K$	Energie cinétique
$Ka_j$	Constante proportionnelle
$Kc_j$	Coefficient du couple moteur $j$



# Introduction Générale

## Introduction générale

La commande d'un robot manipulateur consiste à lui permettre d'atteindre un état désiré pour réaliser l'étude et contrôle d'un bras de robot à base de carte Arduino piloté par interface graphique en Labview ; ou autrement dit, de lui permettre de générer les trajectoires nécessaires pour réaliser ces tâches.

De nos jours, les robots tendent d'investir de plus en plus notre vie quotidienne, leur avantage principal réside dans leur capacité à effectuer des tâches pénibles pour l'homme de manière précise et répétitif. Les bras manipulateurs sont souvent utilisés dans l'industrie pour effectuer des tâches variées de manipulation.

Ils sont les plus utilisés dans les systèmes industriels, car ils sont flexibles permettent d'augmenter la productivité, réduire les coûts de production et améliorer la qualité de la production. Nous retrouvons également ce type de robots dans les hôpitaux où ils assistent l'être humain dans des procédures chirurgicales difficiles ainsi que dans des milieux qui lui sont hostiles, tel que dans des pièces à températures très élevées ou radioactives. Ils ont également fait leurs preuves dans des domaines tels que l'agriculture, la construction, l'exploitation des fonds marins ou de l'espace.

Notre objectif d'étude consiste en l'étude théorique ainsi qu'en la simulation de la commande d'un bras manipulateur à quatre degrés de liberté.

Cet mémoire est structuré en quatre chapitres :

**Chapitre I - Etat de l'art :** contient les généralités, où l'on présente la définition de la robotique, son historique et les différents types de robots manipulateurs en précisant leurs domaines d'utilisation.

**Chapitre II - Modélisation des bras manipulateurs :** présente la modélisation du bras manipulateur étudié et traite la modélisation et de l'élaboration des modèles géométriques directs et inverse de bras étudié.

**Chapitre III - Programmation et interface graphique :** nous avons parlé sur la carte Arduino UNO que nous avons utilisé pour le commander ainsi l'interface graphique.

**Chapitre IV : Commande du bras manipulateur :** Dans ce chapitre, il a été montré les démarches que nous avons suivies pour la commande du bras manipulateurs par interface graphique et comment on a créé cette interface graphique. Et on présente les Tests et résultats obtenus. Finalement une conclusion générale clôturer cette mémoire.

# **Chapitre I : Etat de l'art**

## **Chapitre I : Etat de l'art**

### **I.1 Introduction**

La robotique est considérée comme l'un des principaux axes du progrès scientifique et technologique. Elle comprend l'automatisation de nombreux secteurs d'activités humaines, par

exemple, afin d'augmenter la productivité des usines, de stimuler la compétitivité entre elles et, surtout, de fournir de l'énergie. pour les travaux en zone à haut risque (spatial, démantèlement nucléaire, déminage, etc.).[1]

Dans ce chapitre, on donnera un certain nombre de définitions ainsi qu'un bref historique sur la robotique. On introduira les différents types de robots et leurs constituants. Enfin, on évoquera la classification des systèmes robotiques, tout en ouvrant une courte parenthèse sur les domaines d'applications possibles.

### **I.2. Définitions**

#### **1.2.1. Définition d'un robot**

Un robot est un mécanisme automatique capable d'exécuter certaines tâches selon un programme fixe ou modifiable [2].[3]

#### **I.2.2. Définition de la robotique**

Science et technique de la robotisation, de la conception et de la construction des robots [2].[3]

### **I.3. L'histoire de la robotique**

Depuis la nuit des temps, l'homme a cherché à se faire remplacer pour des tâches spécifiques pouvant être dangereuses, fastidieuses ou longues. Le concept du robot a été établi

par de nombreuses réalisations historiques créatives, tel que : l'horloge à l'eau de Clepsydre introduite par les Babyloniens (1400 avant JC), le théâtre de l'automate du Héron d'Alexandrie

(une centaine d'années après JC) et les automates de l'arabe Al-Jazari (1200 après JC) [3].

## Chapitre I : Etat de l'art

Un de ces spécimen très évolué fut présenté par Jacques de Vaucanson en 1738 : il représentait un homme jouant d'un instrument de musique à vent. Jacques de Vaucanson créa également un automate représentant un canard mangeant et refoulant sa nourriture après ingestion de cette dernière.

Cependant, l'apparition du robot physique a dû attendre l'arrivée de technologies avancées et plus adaptées au cours du XXe siècle.[3]

La robotique est passée par plusieurs générations comme suit :

- 1947 : Premier manipulateur électrique télé- opéré.
- 1954 : Premier robot programmable
- 1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une Chaine de montage de General Motors.
- 1961m : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.
- 1979 : Le robot HILARE. Les chercheurs du L.A.A.S. de Toulouse (France) étudièrent la planification des trajectoires d'un robot mobile ponctuel, dans un environnement totalement connu.
- 1984 : Le robot FLAKEY est équipé de deux roues motrices avec encodeurs, mais sa vitesse maximale est de 66 cm/ms au lieu de quelques centimètres par seconde. Ce robot est capable de naviguer dans des environnements réels.
- 1993 : Le robot ERRATIC a été conçu par Kurt comme un robot mobile de faible coût pour ses cours de robotique.

Les robots mobiles actuels : A présent la plupart des travaux de recherche portent sur les problèmes de perception. La planification de trajectoires, l'analyse et la modélisation de l'environnement de robot, appliqué sur des robots mobiles commerciaux. Également la recherche actuelle sur la conception mécanique des robots mobiles pour des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robots.[4][5]

### I.4. Classification des robots

On retiendra pour notre part 3 types de robot :

- **Les manipulateurs :**
  - Les trajectoires sont non quelconques dans l'espace,

## Chapitre I : Etat de l'art

- Les positions sont discrètes avec 2 ou 3 valeurs par axe,
- La commande est séquentielle. [7]

### ➤ Les télémanipulateurs

Appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :

- Les trajectoires peuvent être quelconques dans l'espace,
- Les trajectoires sont définies de manière instantanée par l'opérateur, généralement à partir d'un pupitre de commande (joystick).[7]

### ➤ Les robots

- Les trajectoires peuvent être quelconques dans l'espace,
- L'exécution est automatique,
- Les informations extéroceptives peuvent modifier le comportement du robot.[7]

## I.5. Types de robots

Il existe deux types de robots : les robots mobiles et les robots manipulateurs

### I.5.1. Robots mobiles

En général, on appelle robots mobiles l'ensemble des robots à base mobile. L'usage veut cependant que l'on montre le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.[8] [9]



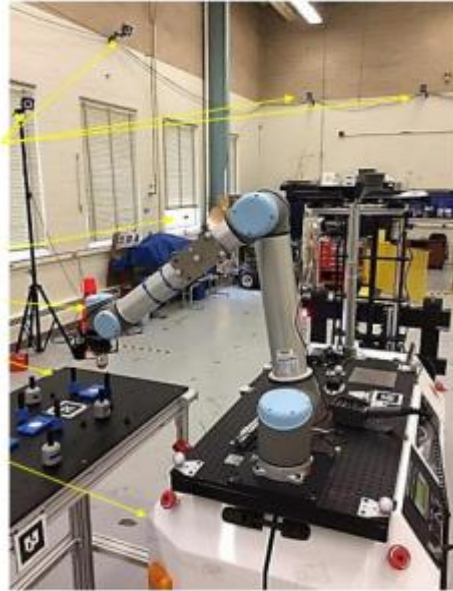
Figure.I.1. Robot mobile [9]

### I.5.2. Robots manipulateurs



## Chapitre I : Etat de l'art

Les robots manipulateurs sont des robots intégrés dans un espace propre destiné au transfert de produits semi-finis entre les lignes de production. Le robot est généralement suspendu pour des questions d'encombrement et par contraintes d'implantation. [8][10]



**Figure.I.2.** Bras Manipulateur [I-10]

### I.5.3 Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte : [7]

- La charge maximum transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- L'architecture du S.M.A., le choix est guidé par la tâche à réaliser (quelle est la rigidité de la structure ?).
- Le volume de travail, défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. L'espace de travail (reachable workspace), également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre via au moins une orientation. L'espace de travail dextre (dextrous workspace) est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur

## Chapitre I : Etat de l'art

(organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.

- Le positionnement absolu, correspondant à l'erreur entre un point souhaité (réel) – défini par une position et une orientation dans l'espace cartésien – et le point atteint et calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 mm.
- La répétabilité, ce paramètre caractérise la capacité que le robot a à retourner vers un point (position, orientation) donné. La répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives. En général, la répétabilité est de l'ordre de 0,1 mm.
- La vitesse de déplacement (vitesse maximum en élongation maximum), accélération. - La masse du robot. - Le coût du robot. - La maintenance, ...

### I.5.4 Robot industriel

La définition diffère selon les pays, on se contentera de la version normalisée par l'organisation internationale de normalisation (ISO) :

« Une machine formée par un mécanisme incluant plusieurs degrés de libertés, ayant souvent l'apparence d'un ou de plusieurs bras se terminant par un poignet capable de tenir des outils, des pièces ou un dispositif d'inspection » [11].

### I.6. Bras manipulateur

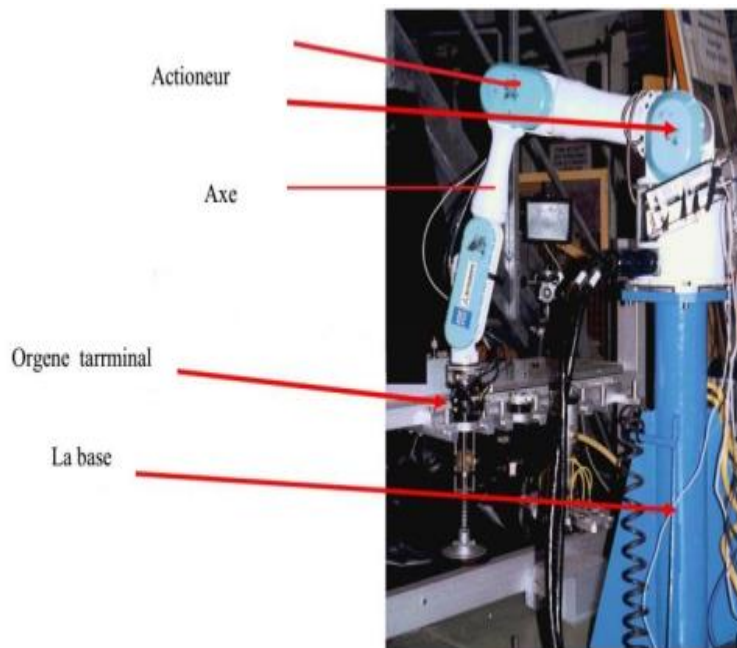
Un bras manipulateur est un bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Il peut être autonome ou contrôlé manuellement et peut

être utilisé pour effectuer une variété de tâches avec une grande précision. [12]

### I.7. Bras manipulateur

La figure ci-dessous représente un exemple d'architecture d'un robot.

## Chapitre I : Etat de l'art



**Figure.I.3.** Architecture d'un robot [13]

Le robot manipulateur est constitué par les éléments suivant :

### **I.7.1. Le porteur**

Le porteur représente l'essentiel du système mécanique articulé (actionneur, segment, articulation, organe terminal), il a pour rôle d'amener l'organe terminal dans une situation imposée [6]. Il est constitué de :

### **I.7.2. Le porteur**

Les actionneurs sont les «muscles» de manipulateurs. Le contrôleur envoie des signaux aux actionneurs, qui, à son tour, déplacent les articulations du robot et des jonctions, les types communs des actionneurs sont les servomoteurs, les moteurs pas à pas, les actionneurs pneumatiques et les vérins hydrauliques. Les actionneurs sont sous le contrôle du contrôleur. [13]

#### **I.7.2.1. Segment :**

C'est le corps principal du robot qui comprend les jonctions, les articulations, et d'autres éléments de structure du robot. Il convient de noter ici que le manipulateur seul n'est pas un robot [13]

#### **I.7.2.3. Effecteur:**

## Chapitre I : Etat de l'art

Cette partie est reliée à la dernière jonction (main) d'un manipulateur qui gère généralement les objets, établit des connexions à d'autres machines ou effectue les tâches requises. [13]

### I.7.2.4. la base:

La base du manipulateur est fixée sur le lieu du travail.[13]

### I.7.2.5. L'organe terminal :

Il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. [14]

## I.8. Constituants d'un robot

Un robot manipulateur est constitué par deux sous-ensembles distincts, un organe terminal (dispositif destiné à manipuler des objets) et une structure mécanique articulée (SMA) (architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations). Les chaînes peuvent être soit ouvertes ou en série (tous les corps ont au plus deux liaisons), arborescentes (au moins l'un des corps a plus de deux liaisons) ou fermées (l'organe terminal est relié à la base du mécanisme par plusieurs chaînes). Ces différentes structures sont montrées dans la Figure I.4.[15]

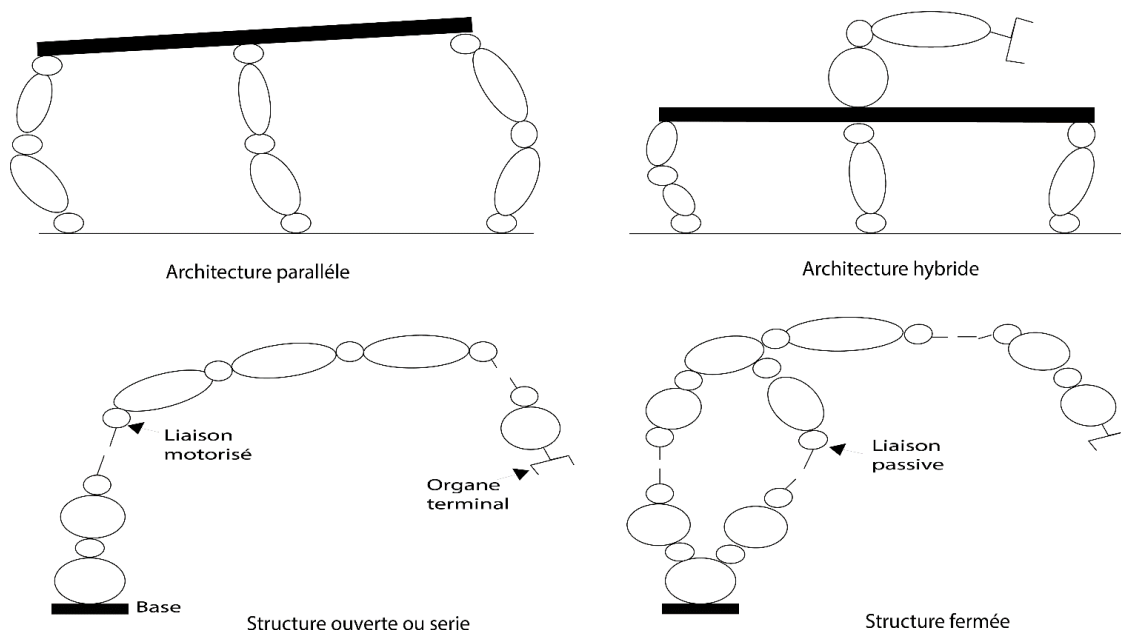


Figure I.4 différentes structures sont montrées [16]

## Chapitre I : Etat de l'art

On distingue classiquement 4 parties principales dans un robot manipulateur :

### **I.8.1. Les actionneurs**

Les actionneurs sont l'ensemble des moteurs associés à des transmissions (courroies crantées) pour animer le S.M.A. Ils utilisent souvent des moteurs électriques à aimant permanent, à courant continu et à commande par l'induit (la tension n'est continue qu'en moyenne, car en général l'alimentation est un hacheur de tension à fréquence élevée). Les actionneurs sont le plus souvent hydrauliques pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique). Ils doivent agir en translation (vérin hydraulique) ou en rotation (moteur hydraulique). Il existe également des actionneurs pneumatiques ayant un usage général pour les manipulateurs à cycles. [8]

### **I.8.2. Le système mécanique articulé (S.M.A.)**

Il s'agit d'un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action. Le rôle du S.M.A. est de guider l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques

de vitesse et d'accélération données. Il est composé d'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. [8]

### **I.8.3. Les articulations**

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Une articulation complexe peut se ramener à une combinaison d'articulations prismatiques ou rotoïdes. [8]

#### **I.8.3.1. Les articulations prismatiques**

Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe. [8]

#### **I.8.3.2. Les articulations rotoïdes**

Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe. [8]

### **I.8.4. Organe terminal**

## **Chapitre I : Etat de l'art**

On regroupe tous les dispositifs destinés à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique. [8]

### **I.8.5. Les capteurs**

Les capteurs sont les organes de perception. Ils sont dits proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle). [8]

### **I.8.6. La commande**

La partie commande synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur. [8]

## **I.9. Classification des robots**

La classification des systèmes robotiques est difficile, car il existe de nombreux critères pour leurs descriptions. Cependant nous allons les classer selon deux structures qui sont les suivantes :

- Point de vue fonctionnel
- Point de vue géométrique

### **I.9.1. Point de vue fonctionnel**

Le nombre de classes et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous : [17]

### **I.9.1. Point de vue fonctionnel**

Le nombre de classes et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous : [17]

#### **I.9.1.1. Manipulateur à commande manuelle**

**Figure.I.5.** représente les manipulateurs à commande manuelle.

## Chapitre I : Etat de l'art



**Figure I.5** Manipulateur à commande manuelle [18]

### I.9.1.2 Manipulateur automatique

**Figure.I.6.** Montre un bras manipulateur qui exerce des mouvements de soudure sans l'intervention de l'homme. [19]



**Figure.I.6.** Manipulateur automatique [19]

### I.9.1.3 Robots programmables

Ils répètent les mouvements qu'on leur a appris ou programmés, sans informations sur l'environnement ou la tâche effectuée.

On peut aussi faire la distinction entre robots « play-back » qui reproduisent la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne.[20]

## Chapitre I : Etat de l'art

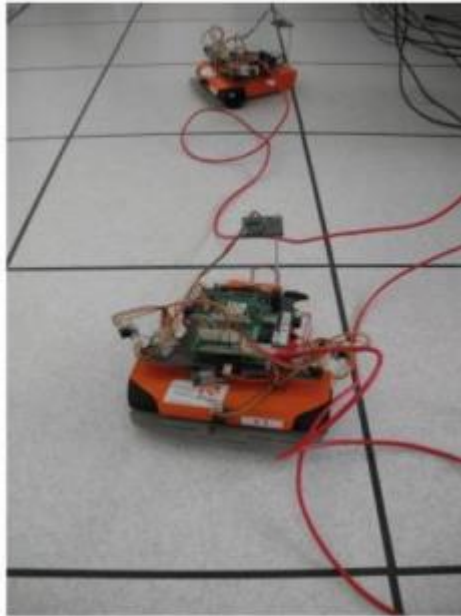


Figure.I.7. Robot programmable [20]

### I.9.1.4. Robots intelligents

On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts, ...). On étudie des robots de troisième génération, capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe, grâce à l'utilisation de l'intelligence artificielle [21]



Figure.I.8 Robot intelligent [22]

### I.9.2 Classification géométrique

Il existe différentes architectures du porteur : la structure cartésienne (PPP), la structure cylindrique (RPP ou PRP), la structure sphérique ou polaire (RRP), la structure dite SCARA (RRP) et enfin la structure anthropomorphe (RRR)



## Chapitre I : Etat de l'art

### I.9.2.1 Structure cartésienne (PPP)

Les caractéristiques d'une structure cartésienne (PPP) sont : trois axes deux à deux en série avec trois degrés de liberté, une très bonne précision et une grande lenteur. Le volume de travail est un parallélépipède dont les dimensions sont les translations permises par les trois liaisons prismatiques. [23]

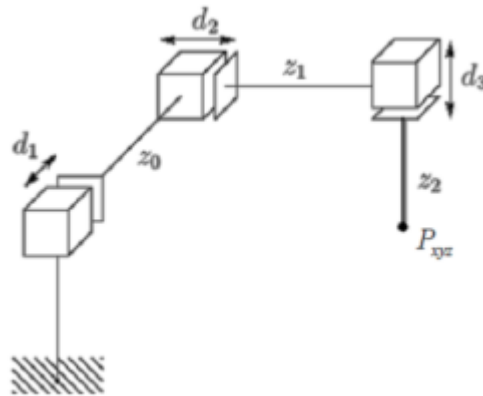


Figure.I.9 Structure PPP [24]

### I.9.2.2. La structure cylindrique (RPP) ou (PRP)

Les caractéristiques d'une structure cylindrique (RPP ou PRP) sont : trois axes non perpendiculaires en série avec trois degrés de liberté et une grande rapidité. Le volume de travail est un cylindre plein ou creux, autrement dit un tore à section rectangulaire, dont la hauteur L est la translation permise par une liaison prismatique.[24]

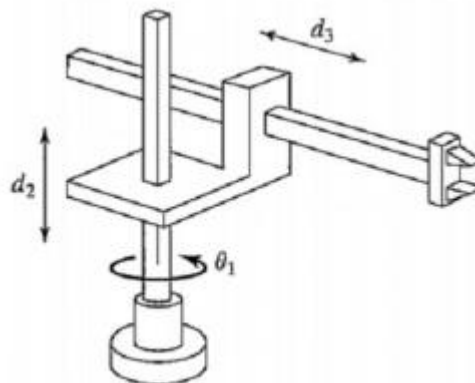


Figure.I.10 Structure RPP [I-25]

### I.9.2.3. La structure sphérique ou polaire à axe de rotation orthogonale (RRP)

Le volume de travail est une sphère creuse, dont les rayons intérieur et extérieur sont

## Chapitre I : Etat de l'art

fixés soit par la disposition de la liaison prismatique et la translation qu'elle permet, soit par les longueurs des deux parties du bras [26].

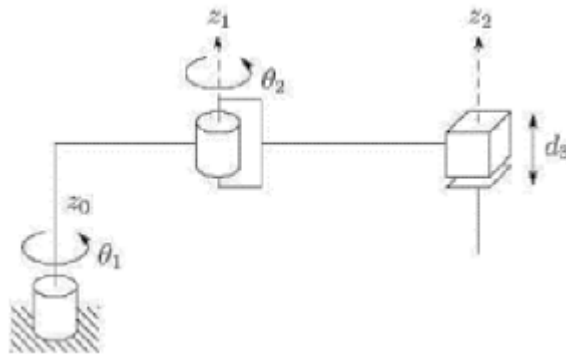


Figure.I.11 Structure RRP [26]

### I.9.2.4. La structure dite SCARA

Les caractéristiques d'une structure SCARA (Selective Compliance Articulated Robot for Assemblage) sont : 3 axes en série cylindrique (RRP) ayant trois degrés de liberté. Les particularités de cette structure sont qu'elles sont précises et très rapides [26].

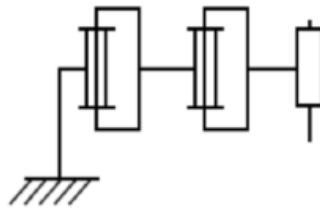


Figure.I.12 Structure SCARA [27]

### I.9.2.5. La structure anthropomorphe (RRR)

Cette structure à une architecture plus généraliste reproduisant le bras humain. Son enveloppe de travail à une cinématique et dynamique complexes, mais sa configuration est plus flexible.[28]

## Chapitre I : Etat de l'art

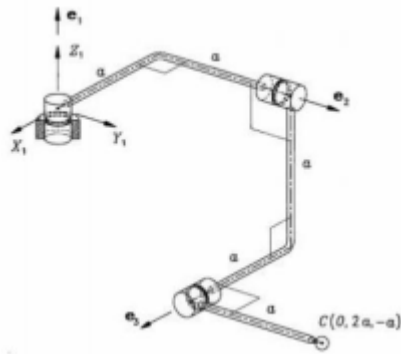


Figure.I.13 Structure RRR [28]

### I.10. Domaines d'application de la robotique

Les domaines d'application sont très variables et ils évoluent les années. On peut citer :

#### I.10.1. Les robots industriels :

Robots industriels sont des robots utilisés dans un environnement de fabrication industrielle. Ils sont utilisés dans la fabrication des automobiles, des composants et des pièces électroniques, des médicaments et de nombreux produits. [29]

#### I.10.2. Robots domestiques ou ménagers :

Robots utilisés à la maison. Ce type de robots comprend de nombreux appareils très différents, tels que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches. En outre, certains robots de surveillance et de télé présence pouvaient être considérées comme des robots ménagers se il est utilisé dans cet environnement. [29]

#### I.10.3. Robots en médecine et chirurgie:

Les robots semblent avoir de l'avenir à l'hôpital. Rob.0odoc aide à réaliser certaines opérations de chirurgie. Le robot infirmier est encore en projet. Le cyber squelette HAL aide les personnes à se déplacer. Et le robot patient permet aux futurs chirurgiens-dentistes d'apprendre à soigner sans faire de dégâts... [29]

### I.11. Les composants d'un système robotique

#### I.11.1. Unité informationnelle

Reçoit les instructions décrivant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de ces tâches. Les systèmes actuels fonctionnent en interaction permanente selon le cycle information-décision-action [30].

## **Chapitre I : Etat de l'art**

### **I.11.2. Unité opérationnelle**

Exerce les actions commandées en empruntant la puissance nécessaire à la source d'énergie. Cette partie, qui constitue le robot physique, intègre la structure mécanique (segments, articulations, architecture...), les modules d'énergie (amplificateurs, variateurs, servovalves...), les convertisseurs d'énergie (moteurs, vérins...), les chaînes cinématiques de transmission mécanique (réducteurs, vis à billes, courroies crantées ....), les capteurs de proprioceptifs placés sur chaque axe pour mesurer en permanence leur position et leur vitesse, et enfin l'effecteur, ou organe terminal, qui est en interaction avec l'environnement [30].

### **I.11.3. La structure mécanique articulée**

Un robot manipulateur est constitué généralement de deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns à la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté (ddl) qui est par définition le nombre de mouvements indépendants possibles d'un solide par rapport au solide qui lui est directement relié. Une structure mécanique articulée peut être représentée par une architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être dites soit ouvertes ou en série dans lesquelles tous les corps ont au plus deux liaisons, ou bien arborescentes où au moins l'un des corps a plus de deux liaisons. Les chaînes peuvent aussi être fermées dans lesquelles l'organe terminal est relié à la base du mécanisme par l'intermédiaire de plusieurs chaînes [31].

#### **I.11.3.1. Structure mécanique articulée à chaîne cinématique simple**

C'est une chaîne cinématique dont chaque membre possède un degré de connexion (nombre de liaisons mécaniques) inférieur ou égal à deux. Un robot sériel est formé d'une chaîne cinématique simple dont la base et l'organe effecteur possèdent un degré de connexion de un (c'est-à-dire qu'il n'est relié qu'à un seul corps) et les autres éléments un degré de connexion de deux [31].

#### **I.11.3.2. Structure mécanique articulée à chaîne cinématiques fermée**

C'est une chaîne cinématique dont l'un des membres, différent de la base, possède un degré de connexion supérieur ou égal à trois [31].

# **Chapitre I : Etat de l'art**

## **I.12. Conclusion**

Dans ce chapitre, nous avons donné un aperçu général de la robotique : l'histoire de ces robots, leur structure, leurs usages, les différents types et leurs classifications et les domaines d'application.

## **Chapitre II : modélisation des bras manipulateurs**

## **Chapitre II : modélisation des bras manipulateurs**

### **II.1.Introduction**

La maîtrise de la conception et du fonctionnement de mécanismes complexes motorisés, ou « machines », a toujours été un facteur important de progrès technologiques et parfois aussi sociaux et économiques dans divers domaines : transports, production industrielle, travaux publics, exploration et travail en milieux hostiles, imagerie médicale, etc. La conception, la fabrication et le pilotage de ces machines ont été rendus possibles grâce aux connaissances scientifiques et techniques en mécanique, thermodynamique, électrotechnique et hydraulique. Ces machines sont destinées à augmenter les possibilités de l'homme pour la rapidité de ses déplacements et de ses actions, et pour l'amplification de sa force et de son champ d'action, notamment dans l'accomplissement de tâches pénibles, dangereuses et/ou répétitives. Les fonctions généralement concernées sont :

Le déplacement à plus ou moins longue distance sur terre, sur mer, dans l'air et sous l'eau ;

La « manipulation » au sens large du terme :

Déplacer un outil pour saisir des objets ou de la matière, les transporter et les déposer,

Exercer des efforts sur l'environnement ; la combinaison des deux fonctions précédentes.

Un robot manipulateur peut donc être considéré d'une manière générale, vu par son environnement, comme un générateur de mouvements et d'efforts dans les diverses directions de l'espace. .[32]

Sur le plan des applications les plus courantes, on peut distinguer :

les robots industriels, travaillant généralement à poste fixe, de manière totalement autonome, et dont les « tâches » sont programmées sur le site par apprentissage, ou bien hors ligne en utilisant un langage spécialisé ou des moyens de conception assistée par ordinateur ;

Les robots pour l'intervention et l'exploration en milieu hostile et mal connu (nucléaire, planétaire, sous-marin) qui sont le plus souvent téléopérés mais peuvent être dotés d'une certaine autonomie locale compte tenu des difficultés relatives aux

## Chapitre II : modélisation des bras manipulateurs

délais des transmissions et à leur faible bande passante. Les outils de la « réalité virtuelle » qui émergent dans de nombreux laboratoires et industries sont susceptibles d'aider les opérateurs dans la commande des manipulateurs dont sont dotés les véhicules robotisés.

Dans tous ces cas où les robots manipulateurs ne sont pas directement télé- opérés et doivent posséder une certaine autonomie d'action, leurs systèmes de commande automatique doivent connaître et compenser les imprécisions éventuelles puisque l'opérateur humain n'est pas directement dans la boucle d'asservissement, pour s'adapter aux caractéristiques des machines et de leur environnement. Pour cela, il est nécessaire de passer par une modélisation mathématique précise de la géométrie et de la dynamique des bras manipulateurs.

Cet article a pour but de familiariser le lecteur avec les principales notions nécessaires à l'analyse de la morphologie d'un robot et de ses capacités de travail, afin de choisir et d'utiliser au mieux un robot industriel du commerce ou même de concevoir une machine particulière mieux adaptée à ses besoins. [32]

### II.2. Modélisation

#### II.2.1. Repères et référentiels

À un instant donné, on repère la position du point M par le vecteur  $r=OM^{\rightarrow}$  appelé vecteur position. Ce dernier est représenté par les valeurs algébriques des projections sur une base orthonormée. [33]

##### II.2.1.1. Coordonnées cartésiennes

On considère un repère constitué de trois axes rattachés à un point origine Caractéristique du solide de référence (R). À ce repère on associe une base orthonormée directe  $(\vec{x}, \vec{y}, \vec{u}, \vec{y}, \vec{u}, \vec{z})$  Les vecteurs  $(\vec{x}, \vec{y}, \vec{u}, \vec{y}, \vec{u}, \vec{z})$  sont alors les vecteurs unitaires des axes OX, OY et OZ respectivement . Les coordonnées cartésiennes sont les plus utilisées en robotique, car elles sont les plus simples pour la mesure des distances. [35]

## Chapitre II : modélisation des bras manipulateurs

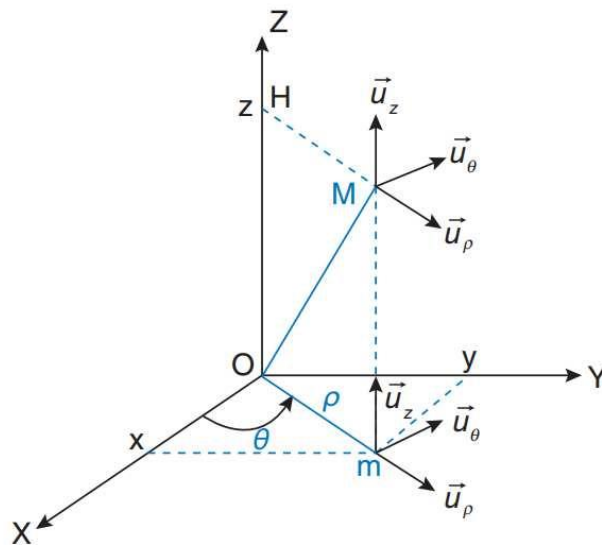


Figure II-1 Le système de coordonnées cartésiennes [35]

### II.2.1.2. Coordonnées cylindriques

La position du point M est définie dans un repère  $(O, \vec{u}_x, \vec{u}_y, \vec{u}_z)$ . On introduit la base  $(\vec{u}_\rho, \vec{u}_\theta, \vec{u}_z)$  orthonormée directe, associée aux coordonnées cylindriques  $(\rho, \theta, z)$ . Les relations entre les coordonnées cylindriques et cartésiennes sont les suivantes [28]

$$\begin{cases} x = \rho \cos(\theta) \\ y = \rho \sin(\theta) \\ z = z \end{cases} \quad (2.1)$$



## Chapitre II : modélisation des bras manipulateurs

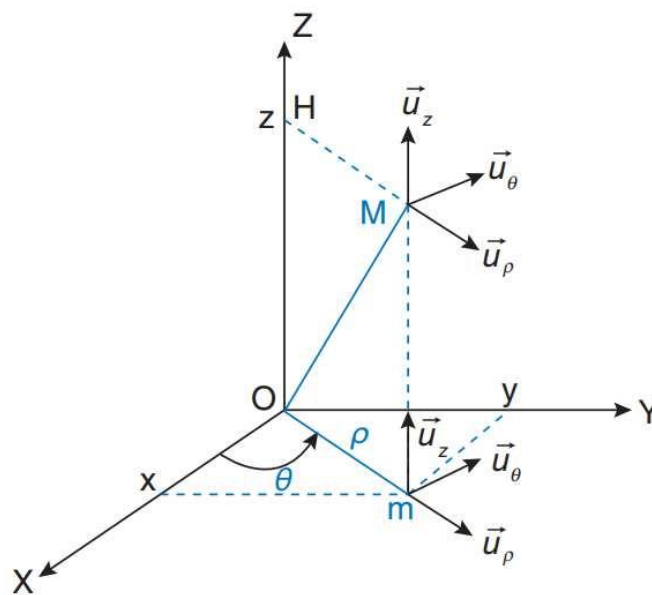


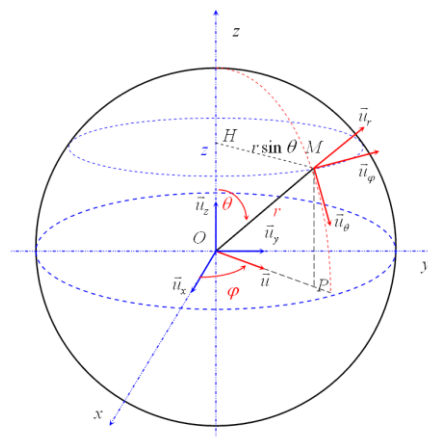
Figure II-2 Le système de coordonnées cylindriques [28]

### II.2.1.3. Coordonnées sphériques

La position du point M est définie dans un repère  $(O, \vec{u}_p, \vec{u}_\theta, \vec{u}_\varphi)$ . On introduit la base

$(\vec{u}_p, \vec{u}_\theta, \vec{u}_\varphi)$  orthonormée directe, associée aux coordonnées sphériques  $(\rho, \theta, \varphi)$ . Les relations entre les coordonnées sphériques et cartésiennes sont les suivantes : [30]

$$\begin{cases} x = \rho \sin(\theta) \cos(\varphi) \\ y = \rho \sin(\theta) \sin(\varphi) \\ z = \rho \cos(\theta) \end{cases} \quad (2.2)$$



## Chapitre II : modélisation des bras manipulateurs

Figure II-3 Le système de coordonnées sphériques [30]

### II.2.2. Position et orientation d'un solide

Si pour décrire la position d'un point, trois dimensions sont nécessaires et suffisantes, Ce n'est plus le cas pour décrire la position d'un solide et son orientation. Il faut trois Coordonnées pour positionner le centre d'inertie des solides et trois autres coordonnées pour Son orientation dans l'espace. Six coordonnées sont donc nécessaires et suffisantes pour placer un objet dans l'espace. [I6]

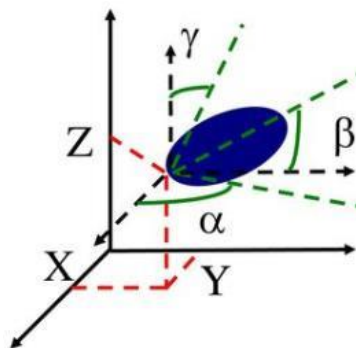


Figure II-4 Position d'un solide dans l'espace et son repère associé [16]

### II.2.3. Modèle géométrique direct

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent

D'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du

Robot, en fonction de ses coordonnées articulaires. Dans le cas d'une chaîne ouverte simple, il Peut être représenté par la matrice de transformation  $T_n^0$  :

$$T_n^0 = T_1^0(q_1)T_2^1(q_2) \dots T_n^{n-1}(q_n) \quad (2.3)$$

Le MGD du robot peut aussi être représenté par la relation :

$$X = f(q) \quad (2.4)$$

## Chapitre II : modélisation des bras manipulateurs

Avec,  $q$  : Vecteur des variables articulaires tel que :

$$q = [q_1 \ q_2 \ \dots \ q_n]^T \quad (2.5)$$

Les coordonnées opérationnelles sont définies par :

$$X = [x_1 \ x_2 \ \dots \ x_m]^T \quad (2.6)$$

Plusieurs possibilités existent pour définir le vecteur  $X$ . Par exemple, avec les éléments de la matrice  $T_n^0$  : [25]

$$X = [P_x \ P_y \ P_z \ s_x \ s_y \ s_z \ n_x \ n_y \ n_z \ a_x \ a_y \ a_z]^T \quad (2.7)$$

Ou bien, sachant que  $s = n_x a$  :

$$X = [P_x \ P_y \ P_z \ n_x \ n_y \ n_z \ a_x \ a_y \ a_z]^T \quad (2.8)$$

### II.2.3.1. Paramètre de Denavit-Hartenberg (DH)

Les paramètres DH sont utilisés pour systématiser la modélisation de n'importe quel type de robot série. Ils permettent de simplifier le modèle géométrique, mais également d'établir une norme reconnue par tous.

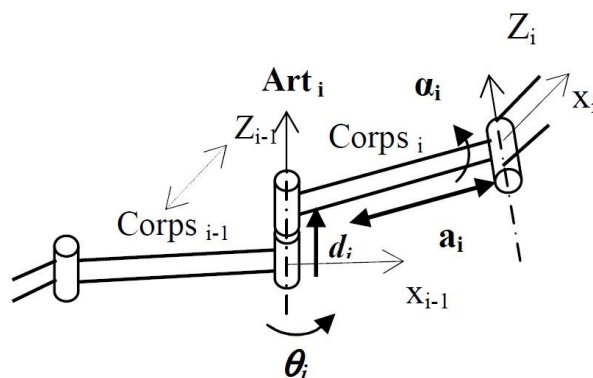
Afin d'établir les paramètres de DH, chacun des repères est établi en se basant sur trois règles :

- L'axe  $Z_{i-1}$  est selon l'axe de mouvement de l'art- $i$ .
- L'axe  $x_{i-1}$  est aligné entre les articulations  $i$  et  $(i+1)$ . Si  $a_i$  est nul, alors  $x_{i-1}$  est perpendiculaire à  $z_{i-1}$  et  $z_i$ .
- L'axe  $Y_i$  complète le repère  $i$  pour former un repère « main droite » .

## Chapitre II : modélisation des bras manipulateurs

Pour passer de  $R_{i-1}$  à  $R_i$ , on peut suivre le chemin suivant :

- |  |  |
|--|--|
| 1- Rotation ( $z_{i-1}$ , $\theta_i$ ) | Avec, $a_i$ : Distance entre $z_{i-1}$ et $z_i$ , le long de $x_i$ .     |
| 2- Translation ( $z_{i-1}$ , $d_i$ )   | $\alpha_i$ : Angle entre $z_{i-1}$ et $z_i$ , autour de $x_i$ .          |
| 3- Translation ( $x_i$ , $a_i$ )       | $\theta_i$ : Angle entre $x_{i-1}$ et $x_i$ , autour de $z_{i-1}$ .      |
| 4- Rotation ( $x_i$ , $\alpha_i$ )     | $d_i$ : Distance de $O_{i-1}$ à l'intersection de $z_{i-1}$ avec $x_i$ . |



**Figure II-5** Schématisation du lien entre le corps  $i-1$  et  $i$ . [25]

La matrice de passage d'une articulation à la suivante est donc :

$$T = T_{i-1}^i = T_{z,\theta_i} \cdot T_{z,d_i} \cdot T_{x,a_i} \cdot T_{x,\alpha_i} \quad \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \cdot s\theta_i & \sin\alpha_i \cdot s\theta_i & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cdot c\theta_i & -\sin\alpha_i \cdot c\theta_i & a_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

## Chapitre II : modélisation des bras manipulateurs

Le MGD d'un bras de robot est obtenu en utilisant le produit : [15]

$$TMG = T_0^n = T_0^1.T_1^2....T_{n-1}^n \quad (2.10)$$

### II.2.3.2. Modélisation géométrique direct de notre simulation

Notre simulation est un bras manipulateur de trois ddl qui se compose d'un organe terminal et d'une structure mécanique articulée. L'organe terminal est constitué d'un servomoteur qui contrôle l'ouverture et la fermeture de la pince. La structure mécanique contient trois servomoteurs qui contrôlent les articulations (de types rotoïdes). Le rôle de cette structure est d'amener l'organe terminal dans une situation (position et orientation) donnée. Le tableau suivant donne le type d'articulations et les angles de rotations possibles de notre simulation. [26]

**Tableau II.1** : Types d'articulations et d'angles de rotations possibles du bras réalisé.

Type d'articulations, d'angles et de rotations de notre réalisation			
Nombre d'articulations		3	
Nombre d'actionneurs		4 servomoteurs	
Rotations		Angles	Types
La base		180°	Rotoïde
Le bras	L'épaule	180°	Rotoïde
	Le coude	180°	Rotoïde
Ouverture de la pince (en mm)		80	

### II.2.3.3 Identification des paramètres de DH de notre bras manipulateur

**Figure.II.-6** représente le placement des repères, selon la convention DH, que l'on a utilisé pour élaborer le MGD du bras.

## Chapitre II : modélisation des bras manipulateurs

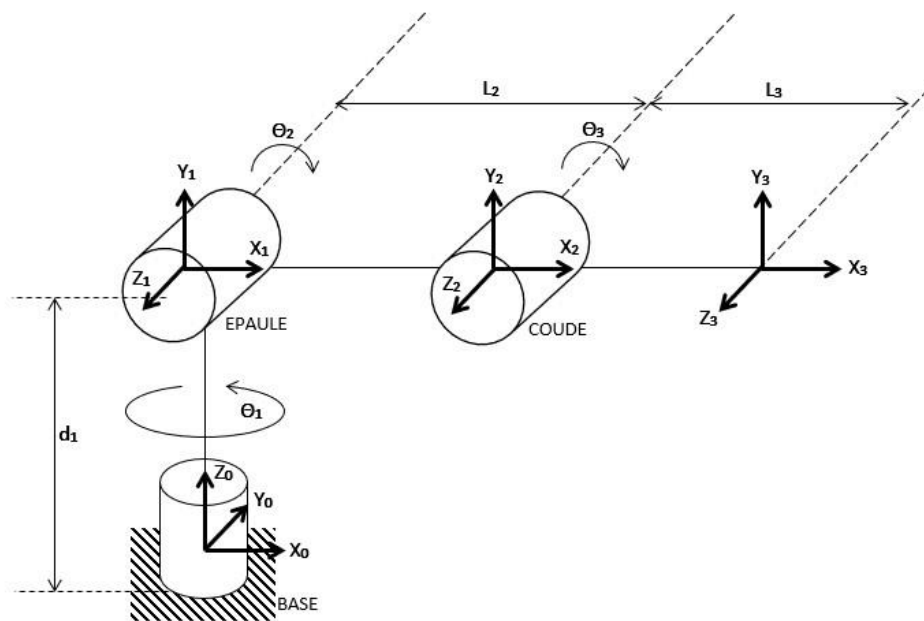


Figure II-6 P lacement des repères selon le modèle DH.

Les paramètres, selon DH, sont donnés par **Tableaux II.2** . [17]

Tableau II.2 : Tableau des paramètres de DENAVIT- HATENBERG de notre simulation.

Segment	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	$d_1$	0	$90^\circ$
2	$\theta_2$	0	$L_2$	0
3	$\theta_3$	0	$L_3$	0

$$T_0^1 = \begin{bmatrix} c\theta_1 & -ca_1s\theta_1 & sa_1s\theta_1 & a_1 \cdot c\theta_1 \\ s\theta_1 & ca_1 \cdot c\theta_1 & -sa_1c\theta_1 & a_1 \cdot s\theta_1 \\ 0 & sa_1 & ca_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

## Chapitre II : modélisation des bras manipulateurs

$$T_1^2 = \begin{bmatrix} c\theta_2 & -c\alpha_2 s\theta_2 & s\alpha_2 s\theta_2 & a_2 \cdot c\theta_2 \\ s\theta_2 & c\alpha_2 \cdot c\theta_2 & -s\alpha_2 c\theta_2 & a_2 \cdot s\theta_2 \\ 0 & S\alpha_2 & c\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_2 & -S_2 & 0 & L_2 C_2 \\ S_2 & C_2 & 0 & L_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

$$T_2^3 = \begin{bmatrix} c\theta_3 & -c\alpha_3 s\theta_3 & s\alpha_3 s\theta_3 & a_3 \cdot c\theta_3 \\ s\theta_3 & c\alpha_3 \cdot c\theta_3 & -s\alpha_3 c\theta_3 & a_3 \cdot s\theta_3 \\ 0 & S\alpha_3 & c\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_3 & -S_3 & 0 & L_3 C_3 \\ S_3 & C_3 & 0 & L_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Le MGD s'obtient à partir de l'équation qui nous donne le résultat suivant :

$$T_0^3 = T_0^1 T_1^2 T_2^3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (L_3 C_{23} + C_2 L_2) \\ S_1 C_{23} & S_1 S_{23} & -C_1 & S_1 (L_3 C_{23} + L_2 C_2) \\ S_{23} & C_{23} & 0 & L_3 S_{23} + L_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

### II.2.4. Le modèle géométrique inverse (MGI)

On a vu que le MGD d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal.

On peut distinguer trois méthodes de calcul du MGI :

- La méthode de Paul qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels,
- La méthode de Pieper qui permet à résoudre le problème des robots à six ddl possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques,

La méthode générale de Raghavan et Roth, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré, au plus, égal à 16. [11]

#### II.2.4.1. Le modèle géométrique inverse de notre réalisation

Pour calculer  $\theta_1$ ,  $\theta_2$  et  $\theta_3$ , on a :

$$T_0^3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (L_3 C_{23} + C_2 L_2) \\ S_1 C_{23} & S_1 S_{23} & -C_1 & S_1 (L_3 C_{23} + L_2 C_2) \\ S_{23} & C_{23} & 0 & L_3 S_{23} + L_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

## Chapitre II : modélisation des bras manipulateurs

On remarque que le calcul de  $\theta_1$  est directement obtenu à partir de  $r_{13}$  et  $r_{23}$ , ce qui implique que :

$$\theta_1 = \text{atan2}(r_{13}, -r_{23}) \quad (2.16)$$

Le calcul de  $\theta_2$  peut se faire en utilisant les composantes  $r_{14}$  et  $r_{34}$ , tel que :

$$r_{14} = (L_3 C_1 C_{23} + L_2 C_1 C_2) \quad (2.17)$$

$$r_{34} = (L_3 S_{23} + L_2 S_2 + d_1) \quad (2.18)$$

Ce qui implique que :

$$(2.19)$$

$$S_2 = \frac{r_{34} - L_3 r_{31} - d_1}{L_2}$$

$$(2.20)$$

$$C_2 = \frac{r_{14} - L_3 C_1 C_{23}}{L_2 C_1}$$



## Chapitre II : modélisation des bras manipulateurs

Ce qui nous donne :

$$\theta_2 = \text{atan2}(S_2, C_2) \quad (2.21)$$

On obtient le calcul de  $\theta_3$  à partir de  $r_{31}$  et  $r_{32}$  comme suit : [24]

On pose :

$$\beta = \text{atan2}(r_{31}, r_{32}) \quad (2.22)$$

Ce qui nous donne :

$$\theta_3 = \beta - \theta_2 \quad (2.23)$$

### II.2.5. Modélisation des actionneurs électriques

Les servomoteurs, comme évoqué précédemment, sont des MCC qui sont asservis en position : ce sont des actionneurs électriques. C'est donc pour cela que l'on va consacrer cette partie à leur modélisation.

Les actionneurs électriques produisent les couples nécessaires aux articulations. Ces couples sont déterminés à partir des équations modélisant ces derniers.

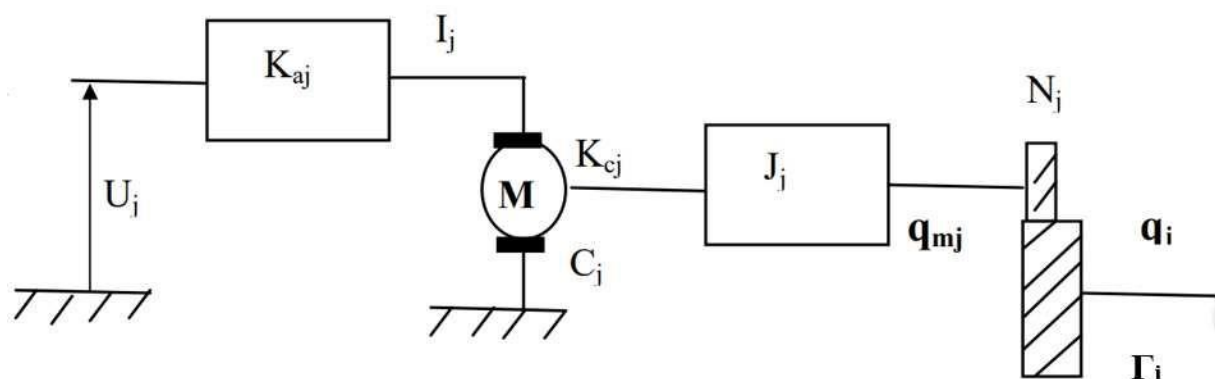


Figure II-7 Schéma fonctionnel d'un actionneur

On peut écrire :

## Chapitre II : modélisation des bras manipulateurs

$$I_j = K_{a_j} \cdot u_j$$

Avec,

$I_j$  : Courant de l'induit du moteur  $j$

$u_j$  : Tension d'entrée

$K_{a_j}$  : Constante proportionnelle

Le courant, traversant l'induit du moteur, produit un couple moteur :

$$C_j = K_{c_j} \cdot I_j = K_{c_j} \cdot K_{a_j} \cdot u_j = J_j \cdot \ddot{q}_{mj} \cdot \frac{\Gamma_i}{N_j} \quad (2.24)$$

Avec,  $\ddot{q}_{mj} = N_j \cdot \ddot{q}_j$  et  $\dot{q}_{mj} = N_j \cdot \dot{q}_j$

$N_j$  : Rapport de réduction,

$K_{c_j}$  : Coefficient du couple moteur  $j$ ,

$J_j$  : Inertie propre du moteur  $j$  ramené à l'arbre du moteur plus celle des organes de transmission,

$\dot{q}_{mj}$  : Vitesse angulaire produite par le moteur  $j$ ,

$\dot{q}_j$  : Vitesse angulaire transmise au corps  $j$  à travers le réducteur,

$\Gamma_i$  : Couple transmis à la  $j^{\text{ème}}$  articulation.

On a alors :

$$N_j \cdot K_{c_j} \cdot K_{a_j} \cdot u_j = (N_j^2 \cdot J_j) \cdot \ddot{q}_j + \Gamma_i = I_{a_j} \cdot \ddot{q}_j + \Gamma_i \quad (2.25)$$

En utilisant la forme matricielle on obtient :

$$N \cdot K_c \cdot K_a \cdot U = I_a \cdot \ddot{q} + \Gamma \quad (2.26)$$

Après avoir obtenu la modélisation des actionneurs électriques, on peut obtenir le modèle dynamique complet du robot, tel que :

$$\Gamma = M(q) \cdot \ddot{q} + C(q, \dot{q}) + G(q) \quad (2.27)$$

## Chapitre II : modélisation des bras manipulateurs

En substituant (1) dans (2), on obtient :

$$N.Kc.Ka.U = [I_a + M(q)].\ddot{q} + C(q,\dot{q}) + G(q) \quad (2.28)$$

Avec,

$M(q)$  : Matrice d'inertie (nxn) symétrique, liée à l'accélération,

$C(q,\dot{q})$  : Vecteur des forces centrifuges et Coriolis non linéaire lié à la vitesse,

$G(q)$  : Vecteur des forces de charges gravitationnelles.

[12]

### II.2.6. Modélisation cinématique direct

Le Modèle cinématique direct(MCD) d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires Il s'écrit :

$$\dot{X} = J(q)\dot{q} \quad (2.29)$$

Ou ,  $J(q)$  désigne la matrice jacobienne de dimension  $(m \times n)$  du mécanisme, égale

$$a \frac{\partial X}{\partial q} \quad (2.30)$$

le calcul de la matrice Jacobienne peut se faire en dérivant le MGD,  $X = f(q)$ , a partir de la relation suivante :

$$J_{ij} = \frac{\partial f_i}{\partial q_j} \quad i = 1, \dots, m; j = 1, \dots, n \quad (2.31)$$

Ou,  $J_{ij}$  est l'élément  $(I,j)$ de la matrice Jacobienne  $J$  .

Cette méthode est pratique pour des robots simples avec un nombre réduit de degrés de liberté. Le calcul de la matrice Jacobienne de base, connu sous le nom de matrice Jacobienne cinématique, est plus pratique pour un robot général de degré de liberté

## Chapitre II : modélisation des bras manipulateurs

égale  $n$ . La matrice Jacobienne obtenue (sans calcul de la dérivée du MGD) relie les vecteurs des vitesses de translation et de rotation  $V_n$  et  $\omega_n$  et les vitesses articulaires:

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J_n \dot{q} \quad (2.32)$$

ou,  $V_n$  et  $\omega_n$  sont les vitesses linéaire et angulaire du repère  $R_n$  respectivement.

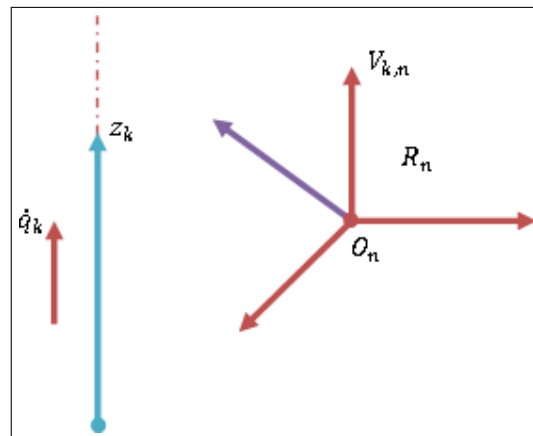
La vitesse  $\dot{q}_k$  de l'articulation  $k$  produit une vitesse linéaire ( $V_{k,n}$ ) et une vitesse angulaire ( $\omega_{k,n}$ ) sur le repère terminal  $R_n$ . Deux cas se présentent :

1. Si l'articulation  $k$  est prismatique ( $\sigma_k = 1$ , Fig 1.1) :

$$V_{k,n} = a_k \dot{q}_k \quad (2.33)$$

$$\omega_{k,n} = 0 \quad (2.34)$$

où,  $a_k$  est le vecteur unitaire porté par l'axe  $z_k$  de l'articulation  $k$ .



**Figure II-8** Cas d'articulation prismatique

Si l'articulation  $k$  est rotoïde ( $\sigma_k = 0$ , Fig 1.2) :

## Chapitre II : modélisation des bras manipulateurs

$$V_{k,n} = a_k \dot{q}_k \times L_{k,n} = (a_k \times L_{k,n}) \dot{q}_k \quad (2.35)$$

$$\omega_{k,n} = a_k \dot{q}_k \quad (2.36)$$

le terme  $L_{k,n}$  désigne le vecteur  $\overrightarrow{O_k O_n}$

De façon générale, les vecteurs  $V_{k,n}$  et  $\omega_{k,n}$  s'écrivent sous la forme :

(2.36)

$$\begin{aligned} V_{k,n} &= [\sigma_k a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ \omega_{k,n} &= \sigma_k \dot{q}_k \end{aligned} \quad (2.37)$$

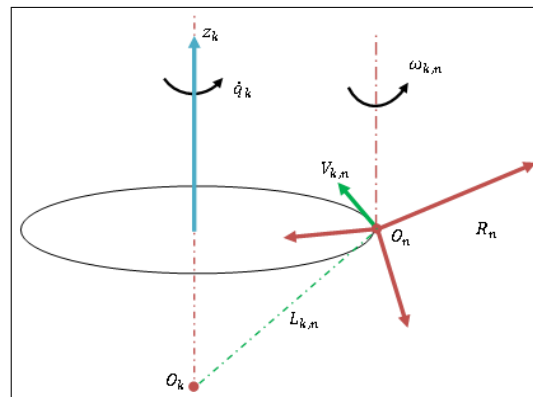


Figure II-9 Cas d'articulation rotoïde

Les vitesses linéaires et angulaires de l'organe terminal peuvent être écrites comme :

$$V_n = \sum_{k=1}^n V_{k,n} = \sum_{k=1}^n [\sigma_k a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \quad (2.38)$$

$$\omega_n = \sum_{k=1}^n \omega_{k,n} = \sum_{k=1}^n \sigma_k \dot{q}_k$$

L'écriture de l'équation (2) sous forme d'une matrice en utilisant l'équation (1),

donne :

$$J_n = \begin{bmatrix} \sigma_1 a_1 + \bar{\sigma}_1 (a_1 \times L_{1,n}) & \cdots & \sigma_n a_n + \bar{\sigma}_n (a_n \times L_{n,n}) \\ \bar{\sigma}_1 a_1 & \cdots & \bar{\sigma}_n a_n \end{bmatrix} \quad (2.39)$$

Se référant les vecteurs de  $J_n$  par rapport au repère  $R_i$ , on obtient la matrice jacobienne  ${}^i J_n$ , telle que :

## Chapitre II : modélisation des bras manipulateurs

$${}^i \begin{bmatrix} V_n \\ \omega_n \end{bmatrix}_n = {}^i J_n \dot{q} \quad (2.40)$$

En général, on calcul  $V_n$  et  $\omega_n$  dans  $R_n$  et  $R_0$ , la matrice jacobéenne correspondante est  ${}^n J_n$  ou  ${}^0 J_n$  respectivement. ces matrices peuvent être aussi calculées en utilisant une matrice  ${}^i J_n$ ,  $j=0, \dots, n$ , grâce à l'expression suivante : [8]

$${}^s J_n = \begin{bmatrix} {}^s A_i & 0_3 \\ 0_3 & {}^s A_i \end{bmatrix} {}^i J_n \quad (2.41)$$

ou  ${}^s A_i$  est la matrice d'orientation du repéré  $R_i$  dans  $R_s$ . En général on obtient la matrice simple  ${}^i J_n$  lorsqu'on prend  $i = \text{entier}(n/2) + 1$ .

On note que les matrices  ${}^i J_n$  ayant les mêmes positions singulières.

Calcul de la matrice  ${}^i J_n$

Le produit  $a_k \times K_{k,n}$  peut être calculé par  $a_k L_{k,n}^2$ , la  $k^{\text{ime}}$  colonne de la matrice  ${}^i J_n$  notée par  ${}^i J_{n:k}$  devienne :

$${}^i J_{n:k} = \begin{bmatrix} \sigma_k {}^i a_k + \bar{\sigma}_k {}^i A_k {}^k \hat{a}_k {}^k L_{k,n} \\ \bar{\sigma}_k {}^i a_k \end{bmatrix} \quad (2.42)$$

On pose  $a_k^k = [0 \ 0 \ 1]^t$  et  ${}^k L_{k,n} = {}^k P_n$ , donc on obtient :

$${}^i J_{n:k} = \begin{bmatrix} \sigma_k a_k^i + \bar{\sigma}_k (-p_{ny}^k s_k^i + p_{nx}^k n_k^i) \\ \bar{\sigma}_k a_k^i \end{bmatrix} \quad (2.43)$$

ou  ${}^k P_{ny}$  et  ${}^k P_{nx}$  sont les composants du vecteur  ${}^k P_n$ . a partir de cette expression, la  $k^{\text{ime}}$  colonne de la matrice  ${}^i J_n$  est :

## Chapitre II : modélisation des bras manipulateurs

$${}^i J_{n:k} = \begin{bmatrix} \sigma_k {}^i a_k + \bar{\sigma}_k {}^i \hat{a}_k ({}^i P_n - {}^i P_k) \\ \bar{\sigma}_k {}^i a_k \end{bmatrix} \quad (2.44)$$

qui donne pour  $i = 0$  :

$${}^0 J_{n:k} = \begin{bmatrix} \sigma_k {}^0 a_k + \bar{\sigma}_k {}^0 \hat{a}_k ({}^0 P_n - {}^0 P_k) \\ \bar{\sigma}_k {}^0 a_k \end{bmatrix} \quad (2.45)$$

dans ce cas on est besoin de calculer les matrices  ${}^0 T_k$  pour  $k=1, \dots, n$ .

### II.2.7. Modèle cinématique inverse

Le modèle cinématique inverse (MCI) donne les vitesses articulaires correspondants a une vitesse désirée  $\dot{X}$  de l'organe terminal. le modèle cinématique inverse s'obtient par la solution d'un syst`eme d'equations linéaires soit analytiquement, soit numériquement. Les solutions analytiques diminuent le nombre d'opérations de façon remarquable par rapport au solutions numériques, mais il faut traiter les cas singuliers séparément. Les solutions numériques sont plus générales et traitent tous les cas de la même façon.

Soit  $X = [X_p^T X_r^T]^T$  une représentation de la situation du repère  $R_n$  dans le repère  $R_0$ . Ou désignent respectivement la position et l'orientation opérationnelles.

Les relations  $X_p^T X_r^T$  entre les vitesses  $\dot{X}_p$  et  $\dot{X}_r$  et les vecteurs  ${}^0 V_n$  et  ${}^0 \omega_n$  sont :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \begin{bmatrix} \Omega_p & 0_3 \\ 0_3 & \Omega_r \end{bmatrix} \begin{bmatrix} {}^0 V_n \\ {}^0 \omega_n \end{bmatrix} = \Omega \begin{bmatrix} {}^0 V_n \\ {}^0 \omega_n \end{bmatrix} \quad (2.46)$$

La matrice  $a$  est défini par :  $a = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (2.47)$

A partir de l'équation (1) le MCD s'écrit sous la forme :  $\dot{x} = j\dot{q}$  [12]

La méthode la plus générale consiste a calculer  $J^{-1}$  la matrice inverse de  $J$ , qui permet de déterminer les vitesses articulaires  $\dot{q}$  grace a la relation :  $\dot{q} = J^{-1}\dot{X}$

## Chapitre II : modélisation des bras manipulateurs

Lorsque la matrice  $J$  a la forme :  $J = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix}$ . Les matrices  $A$  et  $C$  étant carrées

inversibles, il est facile de montrer que l'inverse de cette matrice s'écrit :

$$J^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -C^{-1}BA^{-1} & C^{-1} \end{bmatrix} \quad (2.48)$$

La résolution du problème se ramène donc à l'inversion de la matrice  $J$ .

### II.3 Conclusion

Dans ce second chapitre, on a donné une modélisation aussi simpliste que possible de son modèle géométrique. Enfin, on a terminé par la modélisation des actionneurs électriques afin d'obtenir le modèle dynamique complet généralisé d'un bras manipulateur.



## Chapitre III : Programmation et interface graphique

### Chapitre III Programmation et interface graphique

#### III.1.Introduction

#### III.2.Présentation de la carte Arduino

**Arduino** est une plate-forme de prototypage rapide : un ensemble d'outils développés pour faciliter la conception de montages à base de microcontrôleur, sans perdre trop de temps à en apprendre les tenants et aboutissants.

Les circuits à microcontrôleur, grâce à la grande facilité de leur mise en œuvre et la diminution de leur coût sont de plus en plus fréquemment employés et remplacent avantageusement les circuits électroniques classiques.[36]

La plate-forme **Arduino** comporte :

- du **matériel** : une collection de cartes à microcontrôleurs,
- du **logiciel** : permettant la programmation, la communication et intégrant de nombreuses bibliothèques de fonctions,
- un **site Internet** : information, téléchargements, documentation, forums, ...[18]

# Chapitre III : Programmation et interface graphique

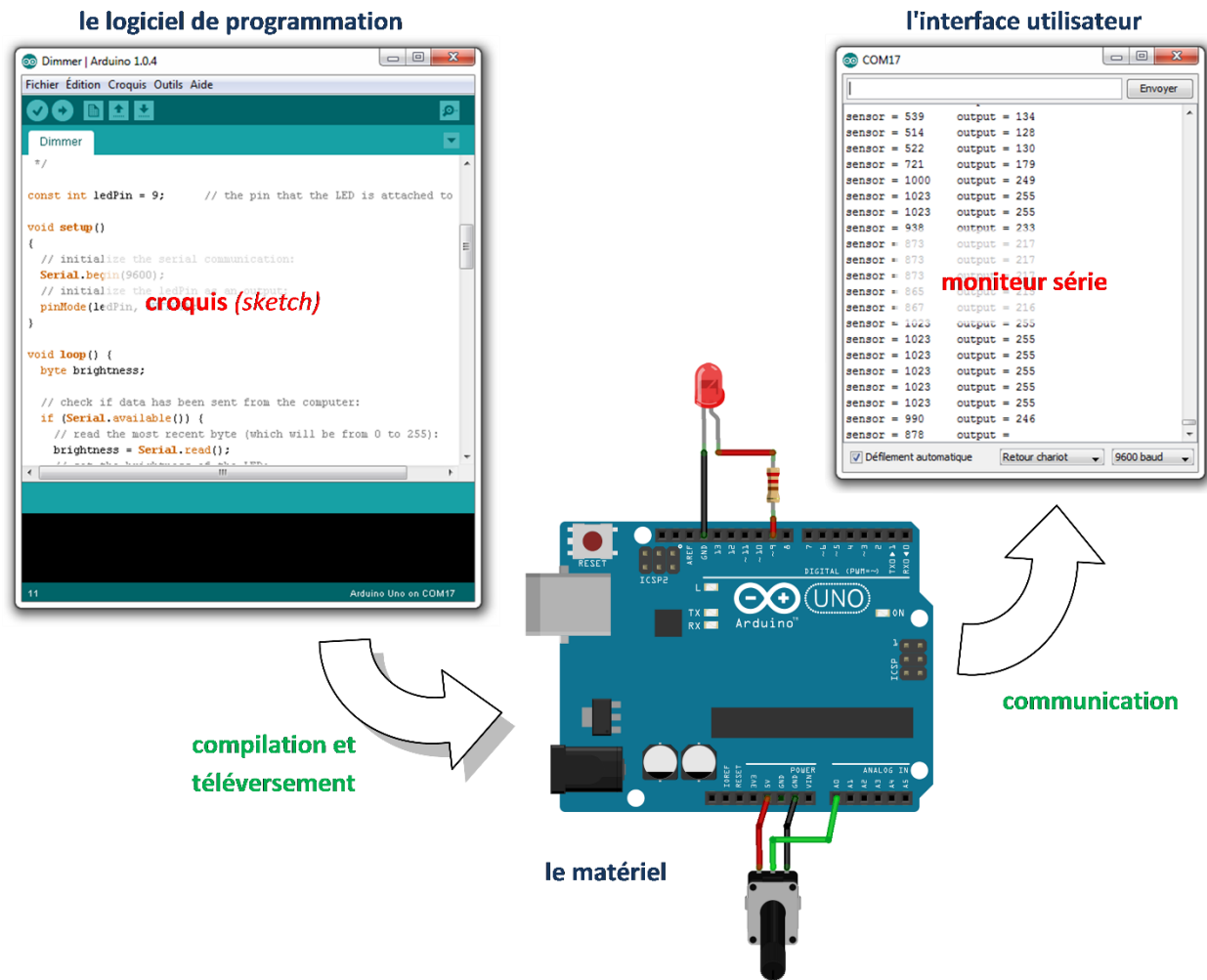


Figure III-1. Communication du logiciel IDE vers le matériel

## III.2.1 La carte Arduino

La carte Arduino Uno est une carte à microcontrôleur. Pour pouvoir l'utiliser, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB, l'alimentation étant fournie par le port USB (on peut aussi l'alimenter avec un adaptateur secteur ou un jeu de piles via le connecteur jack).

Le microcontrôleur est donc un véritable petit ordinateur : [5]

- microprocesseur cadencé à 16MHZ
- mémoire flash  $\Leftrightarrow$  disque dur
- mémoire RAM  $\Leftrightarrow$  mémoire vive (RAM)
- mémoire EEPROM  $\Leftrightarrow$  mémoire morte (ROM)
- broches Entrée/Sortie  $\Leftrightarrow$  ports de communication

## Chapitre III : Programmation et interface graphique

### III.2.2 Les caractéristiques de la carte Arduino

Voyons qu'est-ce qu'il se cache derrière la meilleure carte pour débiter en électronique et commencer à coder ! Certaines cartes vendues sur le marché peuvent vous sembler différentes à la carte Arduino originale. C'est tout simplement car Arduino est un projet Open Source. N'importe qui peut donc assembler et vendre des cartes Arduino.

La majorité des cartes Arduino ont ces composants en communs : [37]

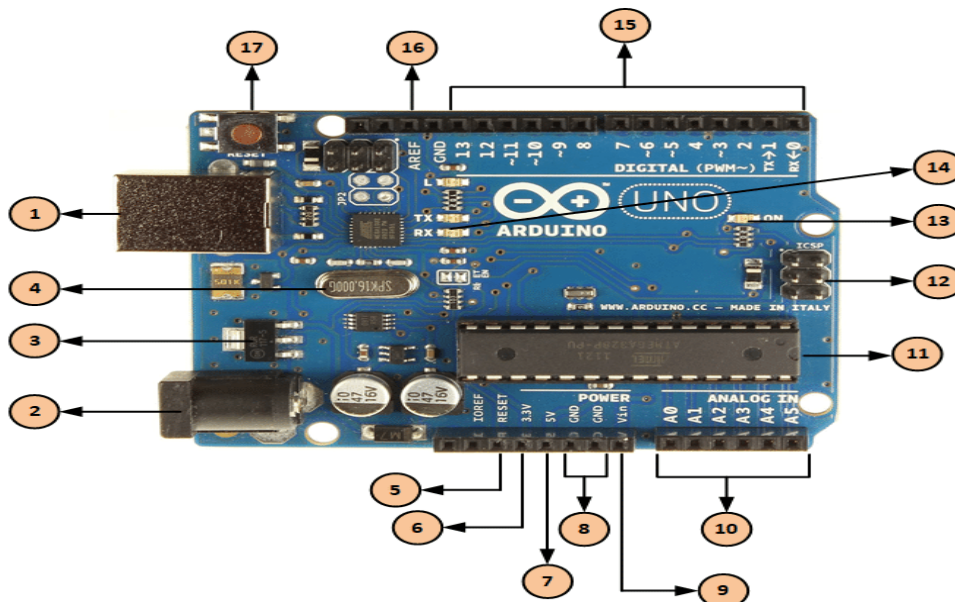






Figure III-2 Composent de la carte arduino.

## Chapitre III : Programmation et interface graphique





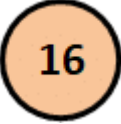
Tableau III.1 : Tableau des composants de la carte arduino.

<p style="text-align: center;"><b>1</b></p>	<p><b>Alimentation / Programmation par USB</b>          La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur.          Tout ce dont vous avez besoin, c'est de connecter votre carte Arduino à votre ordinateur avec le câble USB type A/B.</p>
<p style="text-align: center;"><b>2</b></p>	<p style="text-align: right;"></p> <p><b>Alimentation via connecteur Jack DC</b>  <i>Diamètre interne 2.1mm, externe 5.5mm</i>          La carte Arduino peut être directement alimentée par ce connecteur Jack DC.          Ce connecteur (2) est relié au régulateur de tension intégré à la carte.          L'alimentation via ce connecteur (2) doit être comprise entre 5 et 12 V.</p>
<p style="text-align: center;"><b>3</b></p>	<p><b>Régulateur de tension</b>          La fonction du régulateur de tension (3) est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.</p>
<p style="text-align: center;"><b>4</b></p>	<p><b>Oscillateur à quartz</b>          Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise.          Cet élément aide l'Arduino UNO à calculer les données de temps. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16,000,000 Hertz, soit 16 MHz.</p>
<p style="text-align: center;"><b>5,17</b></p>	<p><b>Arduino Reset</b>          Vous pouvez redémarrer un Arduino avec un "Reset". Cela aura pour effet de redémarrer votre programme depuis le début. Vous pouvez redémarrer l'Arduino UNO de deux manières : soit en utilisant le bouton "Reset" (17), soit en connectant un bouton externe sur la broche de la carte Arduino</p>

## Chapitre III : Programmation et interface graphique

	mentionnée "RESET" (5).
	<p><b>Broches (3.3, 5, GND, Vin)</b></p> <ul style="list-style-type: none"><li>• 3.3V (6) – Broche d'alimentation de tension 3.3 Volts</li><li>• 5V (7) – Broche d'alimentation de tension 5 Volts</li><li>• La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts.</li><li>• GND (8) (Ground / Masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour vos circuits.</li><li>• Vin (9) – Cette broche permet d'alimenter l'Arduino depuis une source de tension extérieure. Elle est relié au circuit d'alimentation principale de la carte Arduino.</li></ul>
	<p><b>Broches analogiques</b></p> <p>L'Arduino UNO possède 5 broches d'entrées analogiques numérotée de A0 jusqu'à A5.</p> <p>Ces broches permettent de lire un signal analogique d'un capteur comme un capteur d'humidité ou de température. La carte Arduino utilise un convertisseur analogique/numérique (convertisseur CAN) pour permettre la lecture du signal par le microcontrôleur.</p> <p>Un signal sera converti sur 10 bits. La valeur pourra être lue sur une échelle 1024 points.</p>
	<p><b>Microcontrôleur principal</b></p> <p>Chaque carte Arduino possède son propre microcontrôleur (11). Vous pouvez le considérer comme le cerveau de la carte Arduino. Le microcontrôleur sur l'Arduino est légèrement différent d'une carte à l'autre. Les microcontrôleurs sont généralement de la société ATMEL.</p> <p>Vous devez savoir quel est le microcontrôleur de votre carte avant de charger un nouveau programme depuis l'IDE Arduino. Cette information est disponible directement sur le composant. Pour plus de détails sur la construction et les fonctions du microcontrôleur, vous pouvez vous référer à la fiche technique (<i>data sheet</i>).</p>

## Chapitre III : Programmation et interface graphique

	<p><b>Connecteur ICSP</b></p> <p>Avant tout, le connecteur <b>ICSP</b> (<i>In-Circuit Serial Programming</i>) est une connectique AVR comprenant les broches MOSI, MISO, SCK, RESET, VCC et GND. Il s'agit d'un connecteur de programmation. Ce connecteur permet entre autre de programmer directement le microcontrôleur sur les couches les plus basses (bootloader, code ASM...).</p> <p>C'est aussi un port appelé port <b>SPI</b> (Serial Peripheral Interface), qui permet de dialoguer avec d'autres composants SPI (écrans, capteurs, etc...). On ne va pas se préoccuper de ce connecteur au début des tutoriels.</p>
	<p><b>Indicateur LED d'alimentation</b></p> <p>Ce voyant doit s'allumer lorsque vous branchez votre Arduino sur une source d'alimentation pour indiquer que votre carte est correctement alimentée. Si cette lumière ne s'allume pas, il y a un problème avec votre alimentation, et je ne parle pas de nourriture ici.</p>
	<p><b>LEDs TX et RX</b></p> <p>Sur votre carte, vous trouverez deux indicateurs : <b>TX (émission)</b> et <b>RX (réception)</b>.</p> <p>Ils apparaissent à deux endroits sur la carte Arduino UNO. Tout d'abord, sur les broches numériques 0 et 1, pour indiquer les broches responsables de la communication série.</p> <p>Deuxièmement, les LEDs TX et RX (13). Le voyant TX clignote à une vitesse variable lors de l'envoi des données série. La vitesse de clignotement dépend de la vitesse de transmission utilisée par la carte. RX clignote pendant le processus de réception. La vitesse de transmission s'exprime en bauds, soit l'équivalent du bits/seconde si le signal est binaire.</p>
	<p><b>Entrées/Sorties numériques</b></p> <p>La carte Arduino UNO possède 14 broches d'Entrées / Sorties numériques (15), dont 6 peuvent fournir une sortie PWM (Pulse Width Modulation). Ces broches peuvent être configurées pour fonctionner comme des broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. Elles peuvent également être utilisées comme des broches de sortie pour piloter différents modules comme des LEDs, des relais, etc. Les broches étiquetées “~” peuvent être utilisées pour générer des PWM.</p>
	<p><b>Broche AREF</b></p> <p>AREF est l'acronyme anglais de “référence analogique”. Cette broche est parfois utilisée pour définir une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.</p>

### III.2.3 L'environnement de programmation Arduino (IDE)

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

## Chapitre III : Programmation et interface graphique

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.

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C. [20]

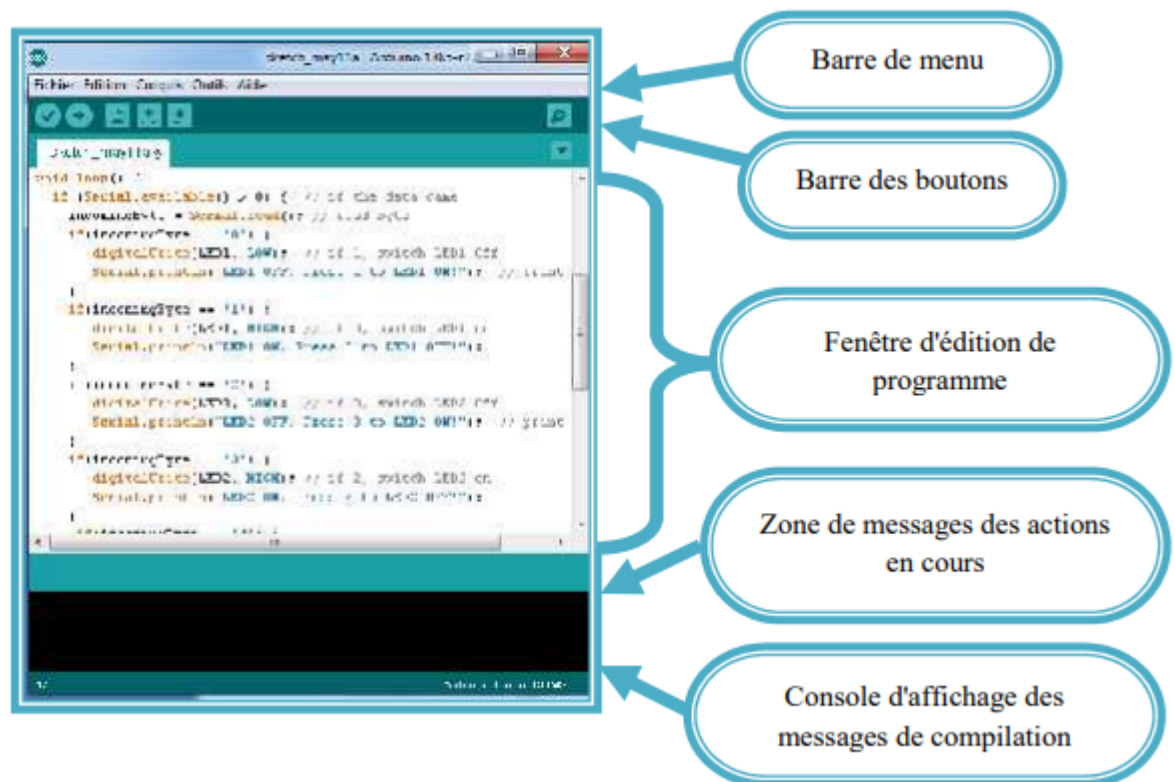


Figure III-3 Interface Logiciel Arduino IDE.

### III.3. Les actionneurs (servomoteurs)

Dans une machine ou un système de commande, semi-automatique ou automatique, un actionneur est l'organe de la partie opérative qui, dès qu'il reçoit un ordre de la partie commande via un éventuel capteur ou pré-actionneur, convertit l'énergie qui lui est fournie

## Chapitre III : Programmation et interface graphique

en un travail utile à l'exécution de tâches, éventuellement programmées, d'un système automatisé. En d'autres termes, un actionneur est l'organe fournissant la force nécessaire à l'exécution d'un travail ordonné par une unité de commande. Il existe trois types d'actionneur :

- Actionneur électrique.
- Actionneur pneumatique.
- Actionneur hydraulique.

Un servomoteur est un actionneur (moteur) capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure.

Le servomoteur est composé de plusieurs éléments visibles :

- Les fils.
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boîtier qui le protège.
- Aussi de plusieurs éléments que l'on ne voit pas qui se trouvent à l'intérieur du boîtier.
- Un moteur à courant continu.
- Des engrenages pour former un réducteur.
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent).
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

La position est définie avec une limite de débattement d'angle de 180 degrés. Souvent abrégé en « servo » provenant du latin servus qui signifie « esclave ».

Il existe plusieurs servomoteurs, dans notre application en a utilisé un servomoteur de faible puissance (9 Gramme) SG90.

Il contient trois fils, un pour l'alimentation, l'autre pour la masse (GND) et la dernière pour le signal d'entrée (la commande) . [30]



## Chapitre III : Programmation et interface graphique

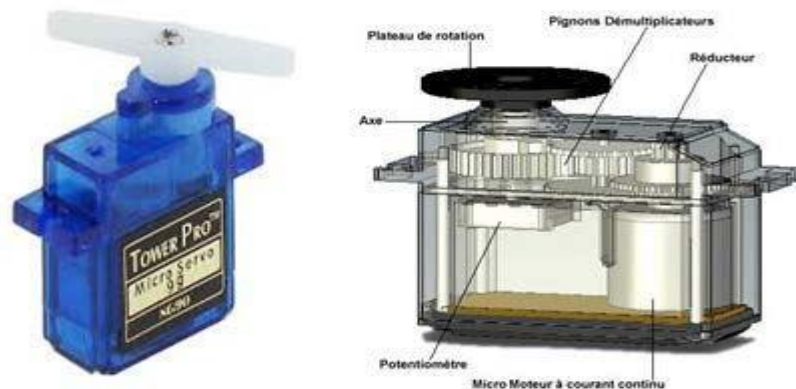


Figure III-4 Servomoteur

### III.3.1 Les paramètres techniques des servomoteurs

Le tableau illustre les paramètres des servomoteurs qu'on a utilisé.

Tableau III.2 : Paramètres techniques des servomoteurs utilisés. [17]

Références / Spécifications	MG995	MG996R	.SG90
Poids	55 g	55 g	9g
Couple d'arrêt	8.5 kgf·cm (4.8 V), 10kgf·cm(6 V)	9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)	1.8 kgf·cm
Tension de fonctionnement	4.8 V à 7.2 V	4.8 V à 7.2 V	4.8 V (~5V)
Vitesse	0.2 s/60° (4.8 V), 0.16 s/60° (6 V)	0.17 s/60° (4.8 V), 0.14 s/60° (6 V).	0.1 s/60°
Rotation angle	180°	180°	180°

### III.4. Présentation du logiciel ISIS (Intelligent Schematic Input System)

Le logiciel ISIS de PROTEUS est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus

## Chapitre III : Programmation et interface graphique

grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect

[24]

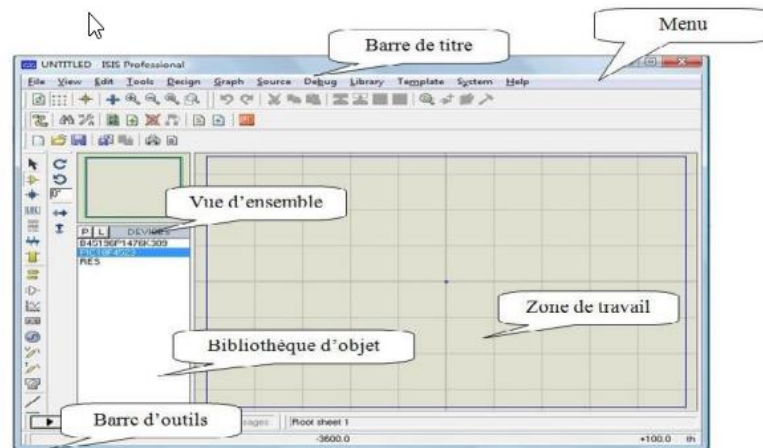


Figure III-5 graphique des circuits

### III.5. Définition de LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un logiciel de développement d'applications mis au point par la société américaine National Instruments. Ce logiciel est utilisé dans un grand nombre de domaines plus particulièrement ceux destinés à l'acquisition de données et au traitement du signal. En effet, il offre de larges possibilités de communication entre l'ordinateur et le monde physique par cartes d'acquisitions, il fournit aussi de multitudes d'outils et de bibliothèques permettant de réaliser de divers traitements sur les signaux mesurés.

LabVIEW utilise un langage de programmation graphique (le langage G) pour créer un programme sous forme de diagramme, au lieu des lignes de textes (programmation par script) utilisées en C par exemple pour écrire des programmes.[35]

### III.6. Description du programme

Lors de la mise sous tension du capteur, on lance le programme par le bouton d'exécution des VIs les DAQ assistant 1 et 2 envoi les données acquises respectivement aux bornes du capteur et du shunte vers les unités de mesure d'amplitude afin d'avoir les valeurs efficace des deux signaux.

## Chapitre III : Programmation et interface graphique

Étant donné que la résistance shunte n'ayant pas une valeur précise de  $1\Omega$ , sa valeur efficace été divisé via l'outil formula par la valeur réel mesuré par un LCR mètre ( $1.1\Omega$  on se qui nous concerne).

Une fois que les valeurs efficaces de la tension et du courant obtenues, ces deux grandeurs seront divisé l'une sur l'autre par un diviseur, nous obtenons au final le module de l'impédance du capteur, le tout sera affiché via des indicateurs numériques. [35]

### III.6.1 Le diagramme

Le diagramme est chargé de mettre en relation les entrées et les sorties ainsi que les différents éléments fonctionnels du programme, Il se compose de : [38]

- **nœuds**

Les nœuds sont des objets sur le diagramme. Ils possèdent des entrées et/ou des sorties et effectuent des tâches spécifiques lorsqu'un VI fonctionne. Ils sont équivalents à des fonctions dans les langages textuels.

- **terminaux**

Les objets situés dans la face avant apparaissent comme des terminaux dans le diagramme. Les terminaux sont des ports de communication entre la face avant et le diagramme reflétant le type de données du contrôle ou de l'indicateur. Ils sont équivalents aux paramètres et aux constantes dans les langages textuels.

- **fil**

L'ensemble des terminaux et des nœuds sont mis en relation par des fils pour transférer les données dans le diagramme. Chaque fil provient d'une seule source mais peut être distribué à plusieurs VIs ou de fonctions destinées à les traiter. Les fils sont analogues aux variables dans les langages textuels.

- **Structures**

Plus communément connues sous l'appellation BOUCLE. Comme dans beaucoup de langages, un programme ne peut être efficace qu'à conditions de disposer de mécanismes d'itération, dans le cas du LabVIEW les plus utilisés :

- Des boucles : For et While

## Chapitre III : Programmation et interface graphique

- Des structures conditionnelles
- Des structures de séquence : Elles sont spécifiques à LabVIEW et permettent de déterminer l'ordre d'exécution de certaines parties de code.

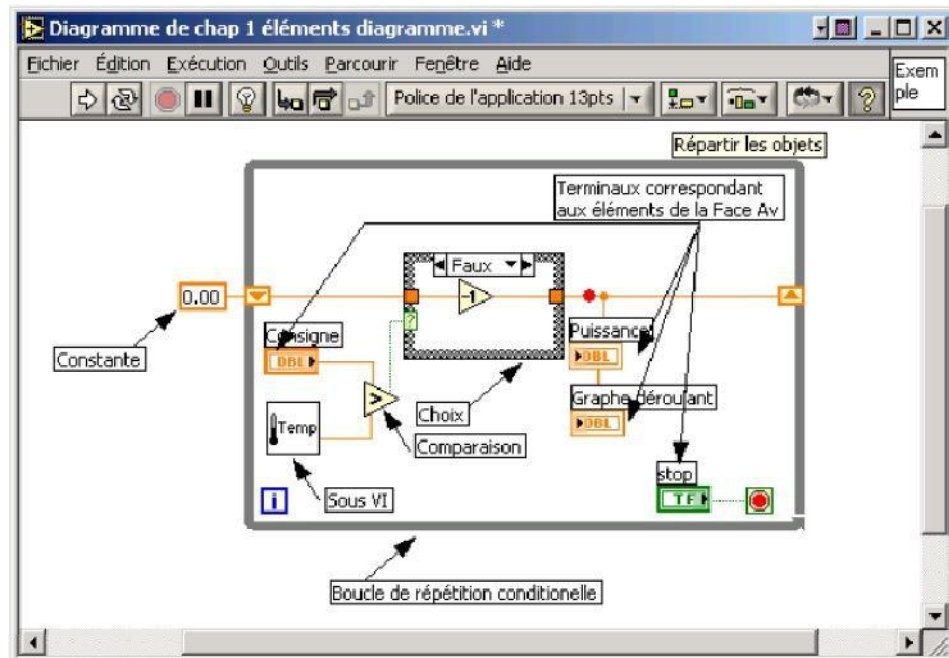
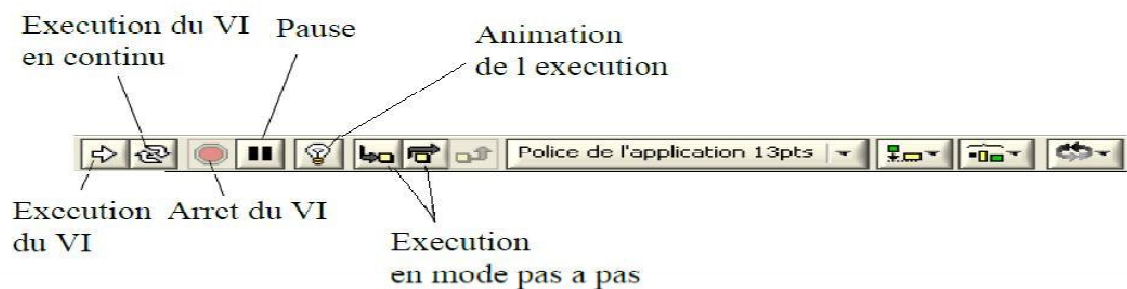


Figure III-6 Fenêtre diagramme.

### III.6.2 Les Outils

Barres d'outils de la face avant et du diagramme.

Les barres d'outils donnent accès aux outils d'exécution et de présentation du programme.





## Chapitre III : Programmation et interface graphique

- Booléens
- Chaînes de caractères
- Tableaux
- Clusters
- Graphiques
- Types spéciaux

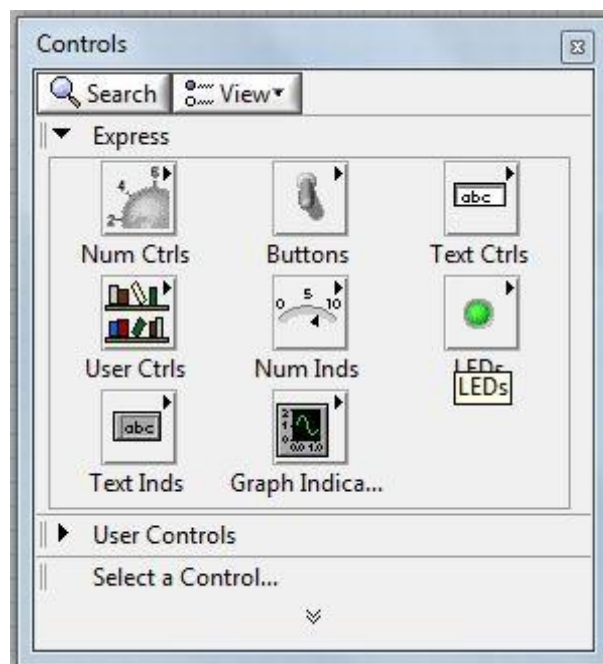


Figure III-9 La palette de commande.

### ➤ Palette de fonctions

La palette Fonctions est seulement accessible sur le diagramme, Elle permet de créer toutes les fonctions du LabVIEW. Chaque icone de la palette correspond à une rubrique qui permet d'accéder à une fenêtre secondaire contenant l'ensemble des éléments relatifs à cette rubrique utilisables sur le diagramme par exemple.

- Structures
- Opérations numériques
- Opérateurs logiques
- Comparaisons
- Fonctions temporelles

## Chapitre III : Programmation et interface graphique

- Entrées/Sorties
- Acquisition
- Traitement du signal

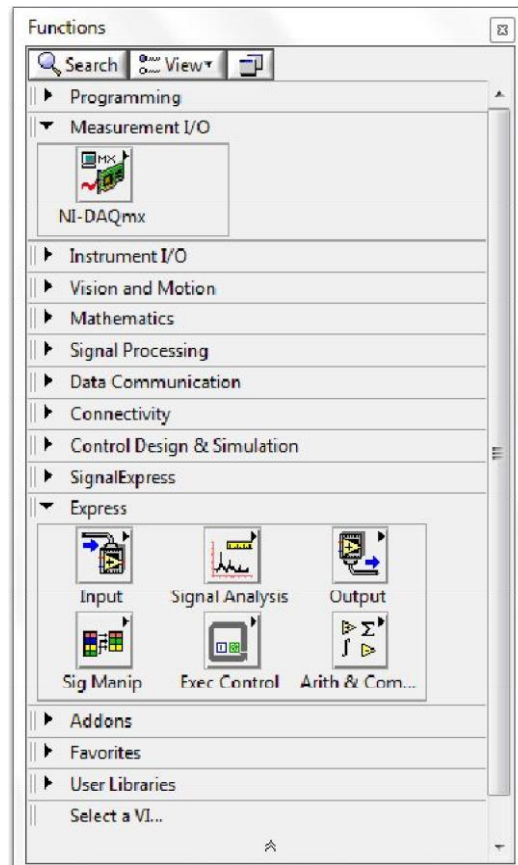


Figure III-10 palette de fonction.

### ➤ La fenêtre d'aide contextuelle

Donne des informations relatives à l'objet situé sous le curseur de la souris, peut être une information sur la nature d'une liaison, le type de données à fournir à un indicateur, le fonctionnement d'un sous VI ou d'une fonction.

## Chapitre III : Programmation et interface graphique

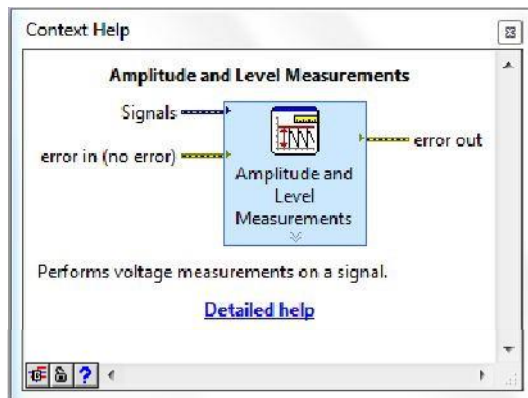


Figure III-11 fenêtre d'aide contextuelle

### III.6.4 Les types de données utilisés dans LabVIEW

Comme la grande majorité des langages de programmation, LabVIEW possède une large palette de variables (entier signés, non-signés, flottants, booléens, chaînes de caractères,...etc.). Afin de les identifier plus facilement dans le diagramme, les différentes familles de variables possèdent chacune une couleur. [38]

<b>Entiers signés</b> (0)	8 bits		<b>Booléen</b> (Faux)			<b>Waveform</b>		
	16 bits			<b>Chaîne</b> (chaîne vide)				<b>Waveform numérique</b>
	32 bits					<b>Chemin</b> (<Pas un chemin>)		
<b>Entiers non signés</b> (0)	8 bits		<b>Refnum</b>				<b>Nom d'E/S</b>	
	16 bits		<b>Type énumération</b>			<b>Horodatage</b>		
	32 bits			<b>Cluster</b>	<b>Numérique</b>			<b>Données dynamiques</b>
<b>Nombres à virgule flottante</b> (0)	Simple précision		<b>Données mixtes</b>				<b>Image</b>	
	Double précision		<b>Tableau</b>		1D			<b>Variant</b>
	Précision étendue			2D				
<b>Complexes à virgule flottante</b> (0 + 0i)	Simple précision							
	Double précision							
	Précision étendue							



## Chapitre III : Programmation et interface graphique

**Figure III-12** Les différents types de données utilisées dans l'environnement LabVIEW

### III.6.5 Les Bibliothèques

L'environnement LabVIEW offre des bibliothèques étendues de fonctions et de routines répondant à la plupart des besoins en programmation. Pour l'analyse et le traitement des données, la bibliothèque d'analyse contient les fonctions pour la génération et le traitement de signaux (les filtres, les statistiques, la régression, l'algèbre linéaire et l'arithmétique matricielle). LabVIEW contient également des bibliothèques de fonctions spécifiques à l'acquisition de données utilisant les différentes portes d'entrée ainsi que des bibliothèques dédiées à la présentation et le stockage des données. [38]

### III.6.6 Logiciel LabVIEW en acquisition de données

L'acquisition des données s'effectue en utilisant une fonction offerte par le logiciel qui nous permet de communiquer avec des instruments reliés à l'ordinateur. Cette fonction est appelée **DAQmx-Data Acquisition**, elle contient des VIs pour le contrôle des cartes d'acquisitions (DAQ), figure (III.10). Parmi ses VIs on trouve DAQ Assistant qui nous permet d'accéder au panneau de configuration de la carte [18].

## Chapitre III : Programmation et interface graphique

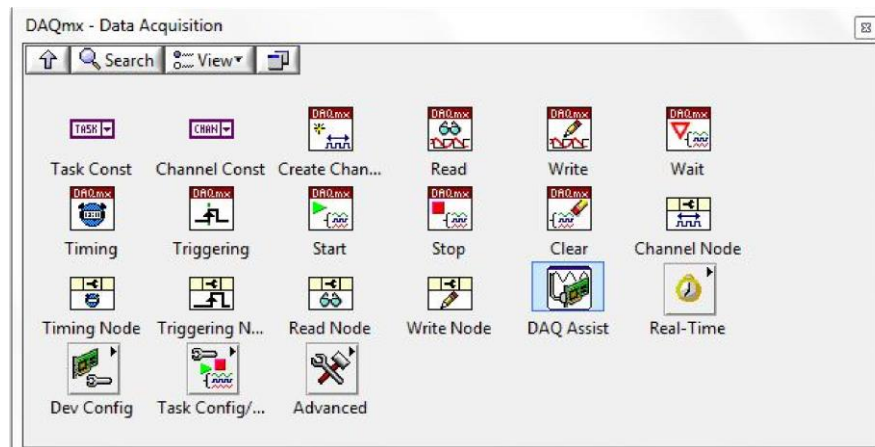


Figure III-13 Palette d'acquisition de données.

### ➤ DAQ Assistant

Nous permet la configuration pas à pas de l'acquisition de données (configuration des entrées/sorties), il permet également de tester cette dernière avant de l'injecter dans le programme et corriger d'éventuelles erreurs signalées.



Figure III-14 DAQ assistant

Une fois posé sur le diagramme une fenêtre de configuration va s'ouvrir pour créer une nouvelle tâche et on aura le choix entre d'acquérir ou générer des signaux.

### III.8.Conclusion

Dans ce chapitre on a déterminé la présentation de l'arduino et ses caractéristique et son environnement de programmation aussi qu'on a déterminé l'actionneur et on décrits le programme labview avec son interface et ses outils.

## **Chapitre IV : Commande du Bras Manipulateurs**

### **Chapitre IV Commande du Bras Manipulateur**

#### **IV.1 Introduction**

Dans ce chapitre, après la présentation de chaque logiciel utilisée, on explique notre simulation et commande appliquée au bras. Enfin on présente la simulation sur MATLAB avec la régulation PI

#### **IV.2 Présentation de la carte Arduino**

La carte Arduino est un microcontrôleur open source, qui a été adopté par la communauté des concepteurs. Il s'agit d'un système numérique à processeur entouré d'une mémoire et des interfaces avec le monde extérieur.[39]

Ces microcontrôleurs ont des performances réduites, mais sont de faible taille et consomment peu d'énergie, les rendant indispensables dans toute solution d'électronique embarquée (voiture, porte de garage, robots, ...). La carte Arduino n'est pas le microcontrôleur le plus puissant, mais son architecture a été publiée en open-source [39].

##### **IV.2 .1 La carte Arduino UNO**

L'Arduino UNO est une carte de développement avec un microcontrôleur Atmel ATMEGA328P qui est couramment utilisé par le grand public.

L'avantage de cette carte est d'éviter d'utiliser des composants nus. Les composants périphériques nécessaires sont déjà présents, il n'y a donc pas besoin de les souder. De plus, l'ATMEGA est assez complet et facile à utiliser. Enfin, elle est également peu chère et connaît un grand succès.

## Chapitre IV : Commande du Bras Manipulateurs

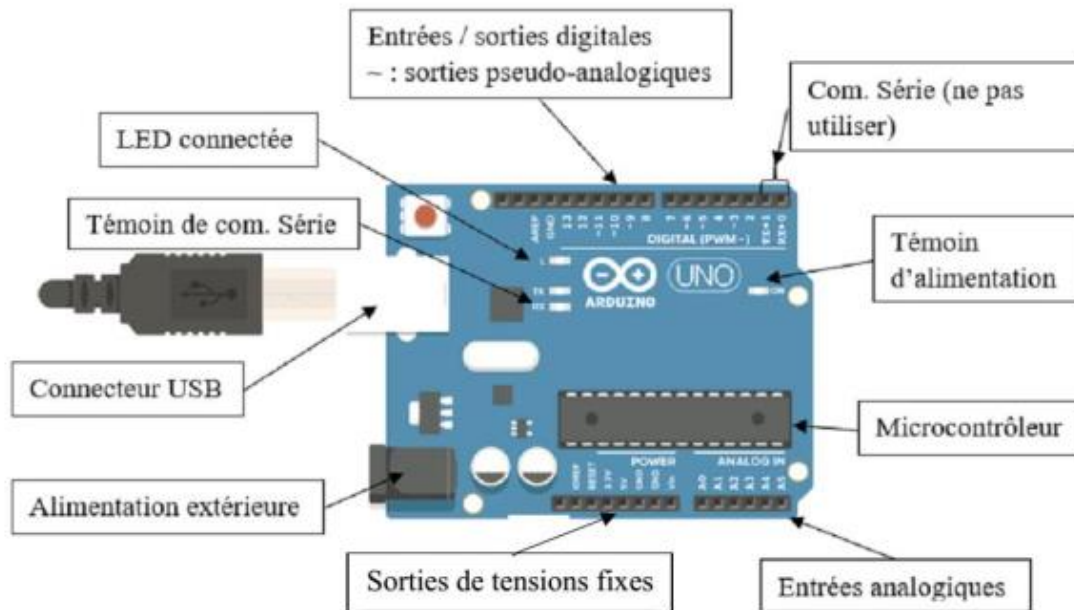


Figure.IV.1. La carte Arduino UNO

### IV.3 Présentation du logiciel ISIS

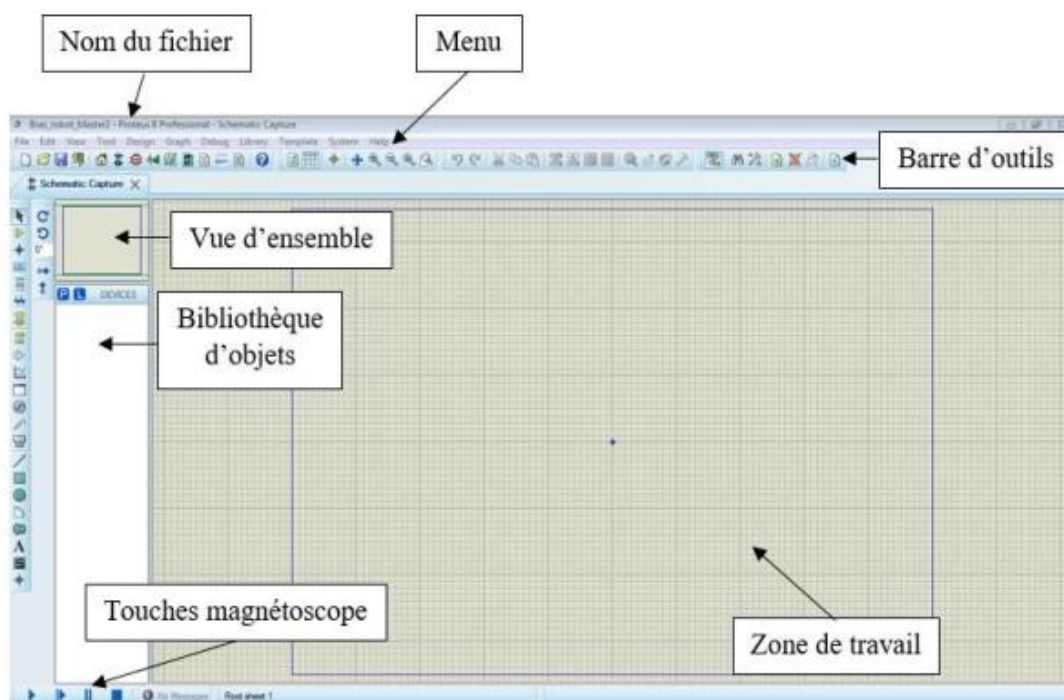
Proteus est une suite logicielle destinée à l'électronique. Développé par la société « Labcenter Electronics », les logiciels inclus dans Proteus permettent la conception assistée par ordinateur dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES [40]. Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de détecter certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits. [40]

## Chapitre IV : Commande du Bras Manipulateurs



**Figure.IV.2.** Ecran du chargement de Proteus.

Isis Proteus est un logiciel de développement et de simulation largement utilisé ces dernières années comme outil pédagogique, en particulier pour sa simplicité due à son



Environnement graphique et interactif. Les différents outils mis à la disposition de l'utilisateur

permettent, outre la simulation, des analyses dans des domaines différents (temporel, fréquentiel), ainsi que des régimes différents (statique, dynamique) [34].

## Chapitre IV : Commande du Bras Manipulateurs

### IV.4 Schéma de circuit sur ISIS

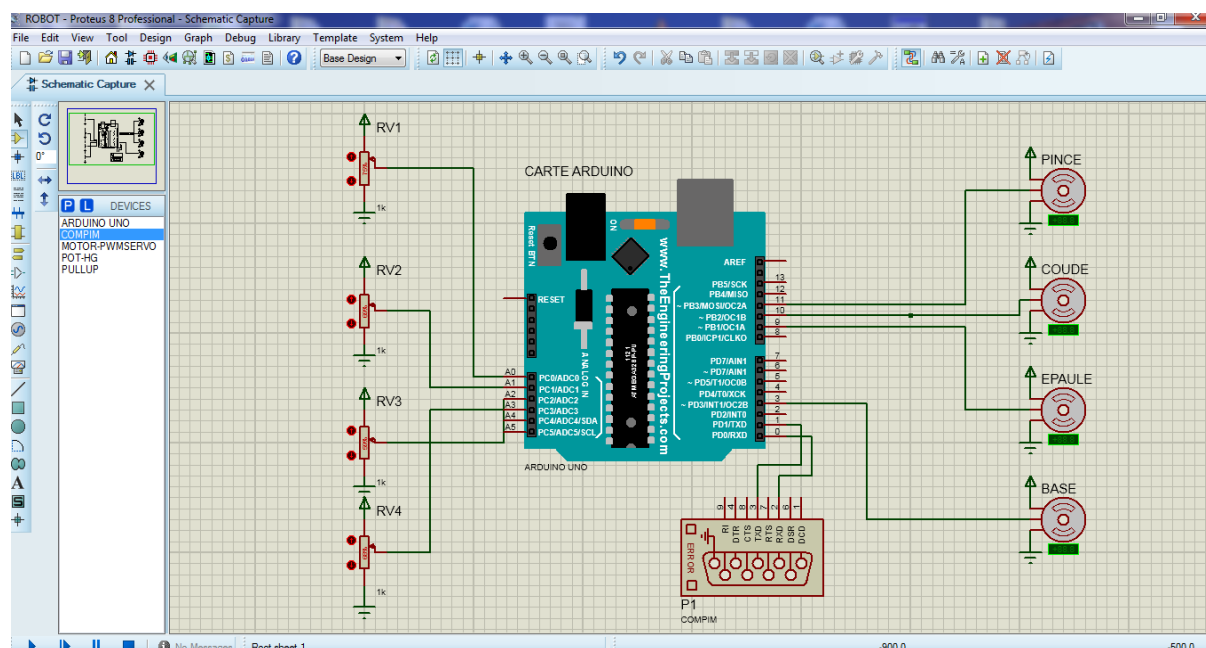


Figure.IV.3 Interface ISIS.

La carte de commande est équipée d'une carte Arduino de type UNO, de quatre potentiomètres qui contrôlent manuellement le bras, de quatre servomoteurs et un compim arduino pour indiquer le port série

### IV.5 Arduino IDE :

Le logiciel Arduino IDE est un environnement de développement associé avec les cartes Arduino. Il est créé en JAVA, open source, et gratuit; peut être téléchargé ou utilisé en ligne depuis le site officiel ([www.arduino.cc](http://www.arduino.cc)).

Le langage de programmation utilisé est dérivé de C et C++, donc il respecte la même syntaxe de ces langues. Il permet de créer, éditer, compiler et de le télé-verser un programme au mémoire de la carte Arduino; comme il facilite la communication avec cette dernière grâce à son terminal et moniteur série.

## Chapitre IV : Commande du Bras Manipulateurs



The image shows the Arduino IDE interface. At the top, the window title is "sketch\_jun06a | Arduino 1.8.19". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, undo, redo, and other functions. The main editor area displays the following C++ code:

```
void loop() {  
  val = analogRead(pin1);  
  val = map(val, 0, 1023, 0, 180);  
  EEPROM.write(data, val);  
  servol.write(val);  
  delay(1);  
  
  val = analogRead(pin2);  
  val = map(val, 0, 1023, 0, 180);  
  EEPROM.write(data, val);  
  servo2.write(val);  
  delay(1);  
  
  val = analogRead(pin3);  
  val = map(val, 0, 1023, 0, 180);  
  EEPROM.write(data, val);  
  servo3.write(val);  
  ...  
}
```

Below the code editor, a status bar indicates "Done compiling." Below that, a terminal window shows the compilation command and output:

```
" rcs "C:\\Users\\Toshiba\\AppData\\Local\\Temp\\arduino_build_98963\\core\\core.a" "C:\\Users\\Toshiba\\Ap  
" rcs "C:\\Users\\Toshiba\\AppData\\Local\\Temp\\arduino_build_98963\\core\\core.a" "C:\\Users\\Toshiba\\Ap  
ab0d2ac1c7ef603c8489eb8dafc.a  
  
w -Os -g -flto -fuse-linker-plugin -Wl,--gc-sections -mmcu=atmega328p -o "C:\\Users\\Toshiba\\AppData\\Loc  
y" -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-warnings --change-section-lma .eep  
y" -O ihex -R .eeprom "C:\\Users\\Toshiba\\AppData\\Local\\Temp\\arduino_build_98963\\sketch_jun06a.ino.elf"  
  
\\libraries\\EEPROM  
-A "C:\\Users\\Toshiba\\AppData\\Local\\Temp\\arduino_build_98963\\sketch_jun06a.ino.elf"
```

Figure.IV.4 Environnement de développement Arduino.

### IV.6 Logiciel « LabVIEW »

L'interface graphique, appelée communément la GUI (pour Graphical User Interface), est le lien entre l'utilisateur et la machine. [41]

De nos jours, les outils de réalisation d'interface graphiques ne manquent pas, comme le DELPHI, MATLAB, LABVIEW, VISUEL BASIC, mais il faut bien savoir choisir le logiciel avec quoi on va travailler. En ce qui nous concerne, nous avons choisi d'utiliser « LabVIEW [41]

# Chapitre IV : Commande du Bras Manipulateurs

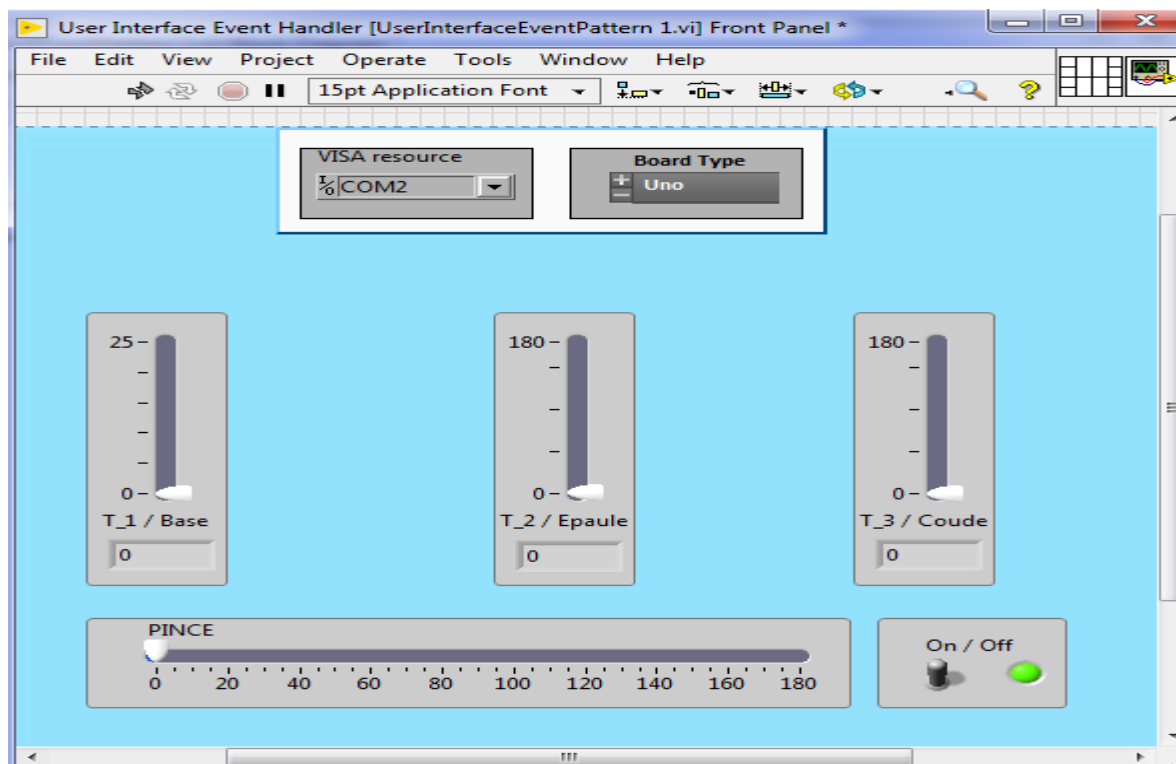
## IV.6.1 Programmation et interface graphique



Figure.IV.5. Interface de LabVIEW

## IV.6.2 Interface de commande sur LabVIEW

### IV.6.2.1 Panneau avant (front panel) :





## Chapitre IV : Commande du Bras Manipulateurs

Figure.IV.6. Panneau avant

### IV.6.2.2 Fonctionnement :

Les deux menus en haut sont utilisé pour sélectionner le porte série et le type de carte de commande avant de commencer la commande par l'appui sur l'interrupteur. Pour la commande, on a utilisé quatre curseurs. Les trois verticales contrôlent la base, l'Epaule, le coude ; le curseur horizontal contrôle l'ouverture et fermeture de pince.

### IV.6.2.3 Block de diagramme :

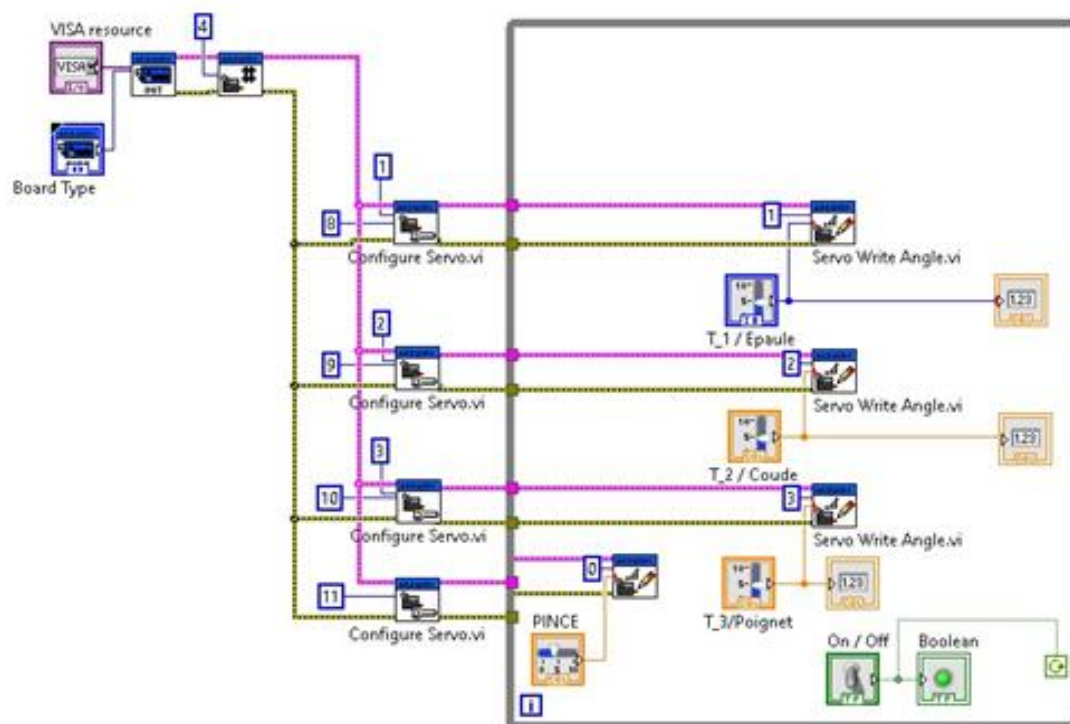


Figure.IV.7. Block de diagramme sur LabVIEW

### IV.6.2.4 Description du programme

Pour voir une bon description de la commande, on étudie le block de diagramme en quatre parties comme il montre la figure suivante ou A, B, C, et D signifie aussi l'ordre d'exécution.

- A) Configuration
- B) Définition des broches

## Chapitre IV : Commande du Bras Manipulateurs

- C) Démarrage de boucle
- D) Commande des servomoteurs

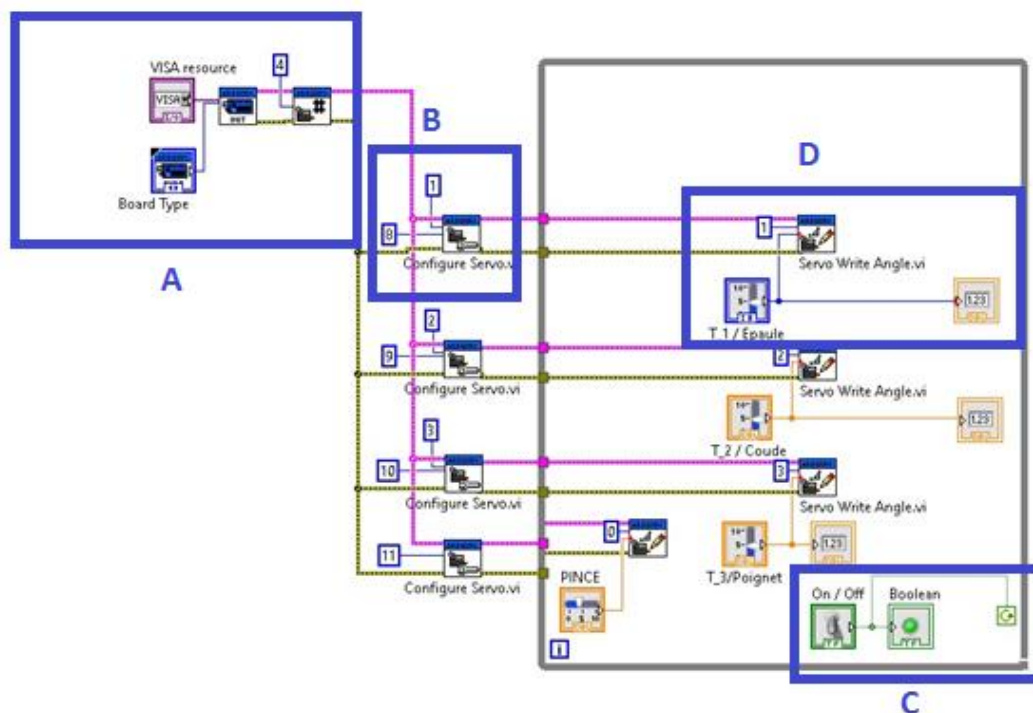


Figure.IV.8. Description de block de diagramme.

### IV.7 Logiciel Virtual Serial Port Driver :

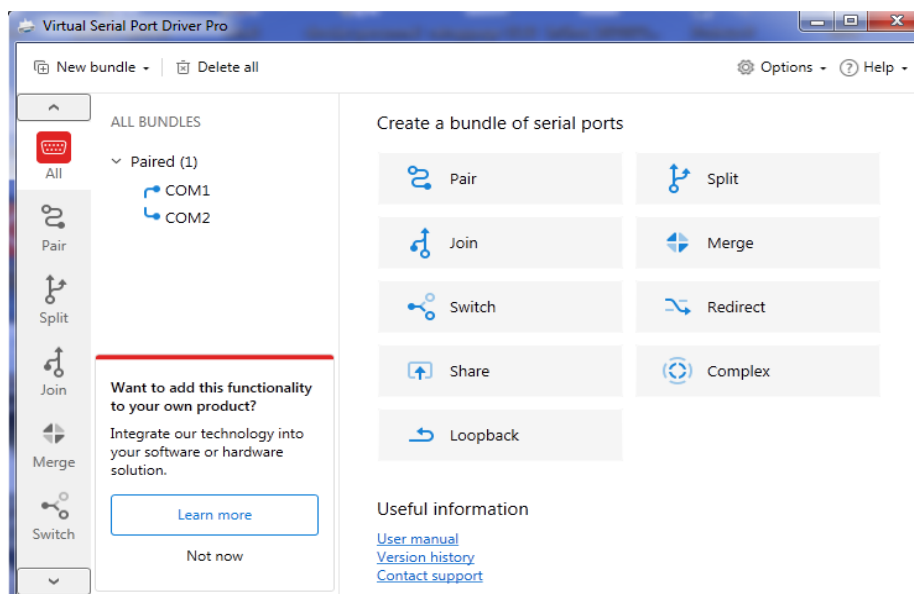
C'est un logiciel a petit taille qui permet de créer et configurer des ports séries virtuels, et aussi de faire des connections entre eux.

Cette fonctionnalité est très usuelle lors de simulation en rendant possible d'inter-changer les données entre deux ou plus logiciels.

#### IV.7.1 Configuration VSPK :

Après de créer deux port virtuel ('COM1' et 'COM2'), on a les connecté pour fournir un moyen d'interfacé LabVIEW avec ISIS comme il montre la figure ci-dessus :

## Chapitre IV : Commande du Bras Manipulateurs



**Figure.IV.9.** L'interface de logiciel Virtual Serial Port Driver

### IV.8 Simulation du bras manipulateur avec la commande PD

Toutes les commandes exposées et implémentées pratiquement sont réalisées en boucle ouverte. La commande en boucle fermée, plus utile en présence des perturbations, constitue une extension intéressante de ce travail. Néanmoins, on a abordé le sujet dans ce mémoire, par une étude théorique et une simulation sous (MATLAB SIMULINK). Avec simulation en 3D

Une fois le modèle dynamique du robot établi, le plus intéressant reste la commande du bras manipulateur. Cette matière a fait l'objet de nombreuses études. L'objet principal de notre travail, on a opté pour un régulateur PI.

#### IV.8.1 Le régulateur PD

Le régulateur PD (Proportionnel, Dérivé) est un régulateur, en boucle fermée, qui a pour rôle de diminuer l'erreur entre la consigne et la mesure et d'atteindre la valeur souhaitée pour l'une des variables du système (vitesse, position...) [42].

La régulation sert à maintenir des grandeurs constantes, en minimisant rapidement l'effet des perturbations en le compensant. La poursuite sert, quant à elle, à s'adapter rapidement aux nouvelles consignes. C'est ce qu'on appelle l'asservissement.

## Chapitre IV : Commande du Bras Manipulateurs

### IV.8.2 Commandes Dynamiques De Robot Manipulateur

Dans cette section on présente le résultat de simulation effectuée, afin d'évaluer la loi de commande PD

Les positions Désirées Sont :  $q_d = [0.5 \quad 1 \quad -0.5]$

La figure suivante présente le Schéma bloc de la commande P D

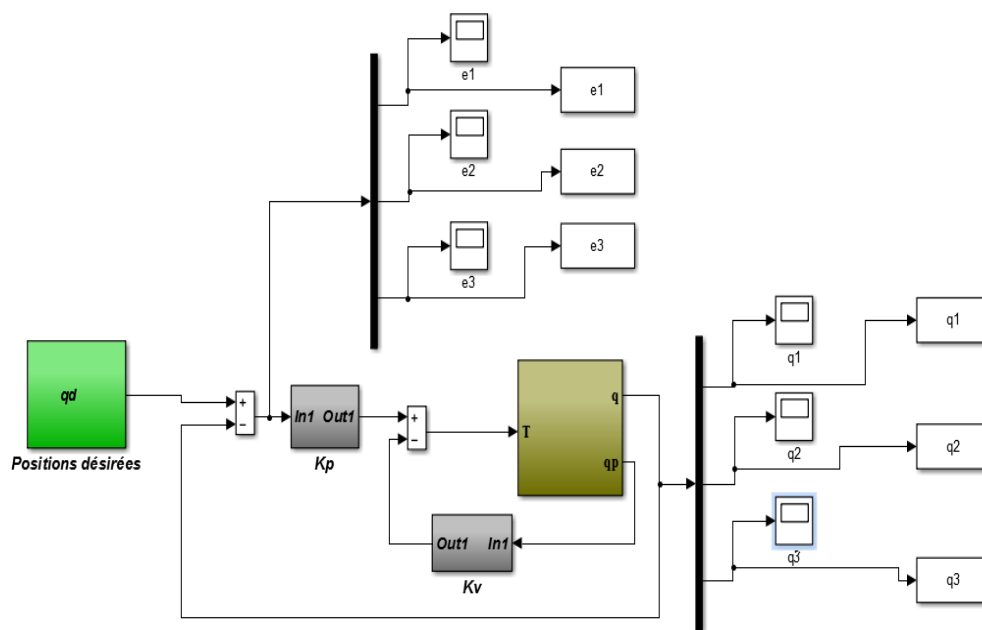


Figure.IV.10 Schéma bloc de la commande P D

Les paramètres de simulation sont les matrices des gains  $K_p$  et  $K_v$  données par :

$$K_p = \begin{bmatrix} 3500 & 0 & 0 \\ 0 & 3500 & 0 \\ 0 & 0 & 3500 \end{bmatrix} \quad K_v = \begin{bmatrix} 250 & 0 & 0 \\ 0 & 350 & 0 \\ 0 & 0 & 190 \end{bmatrix}$$

## Chapitre IV : Commande du Bras Manipulateurs

La figure suivante montre les Positions et les erreurs des différents segments par la loi de commande PD

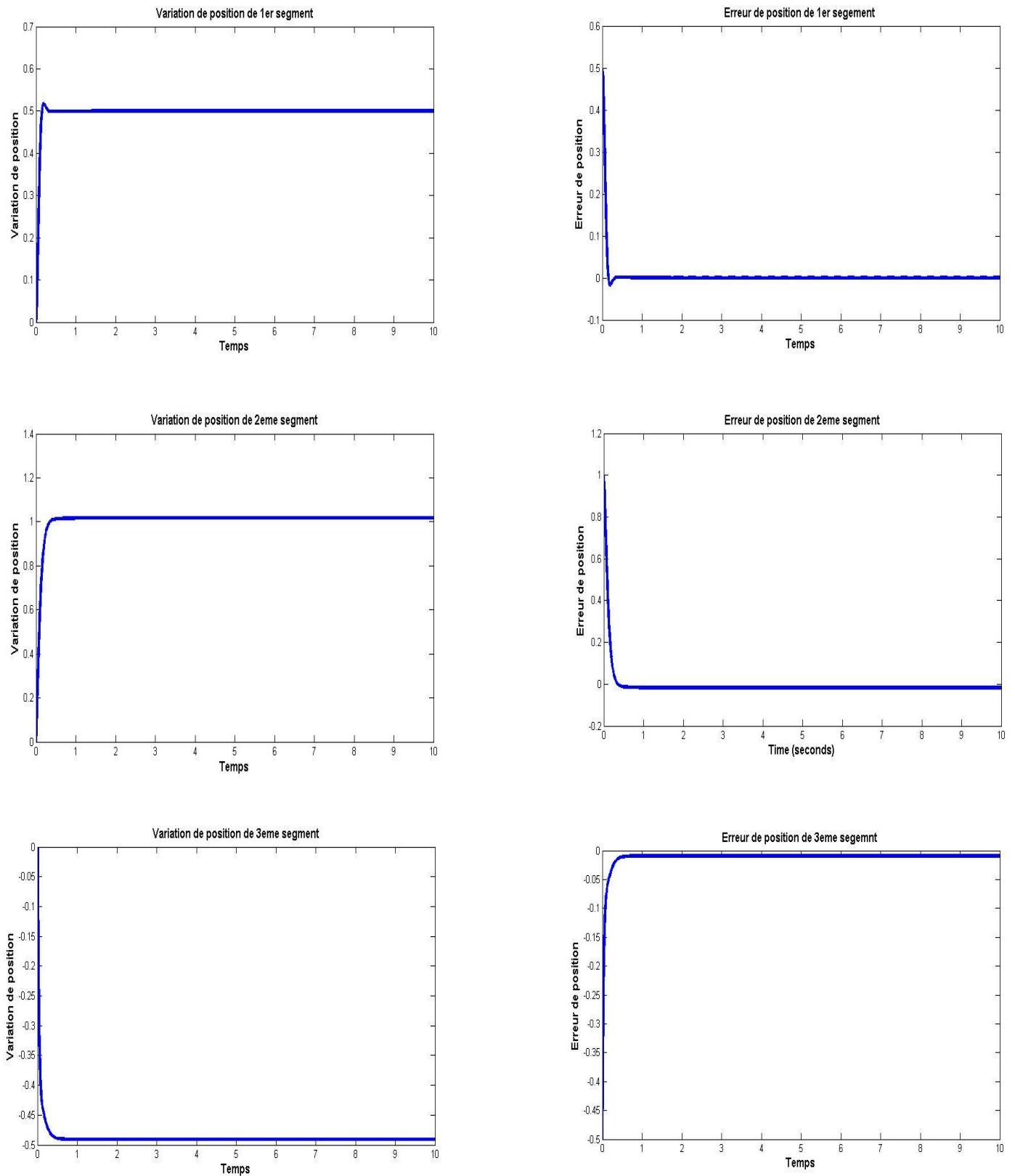
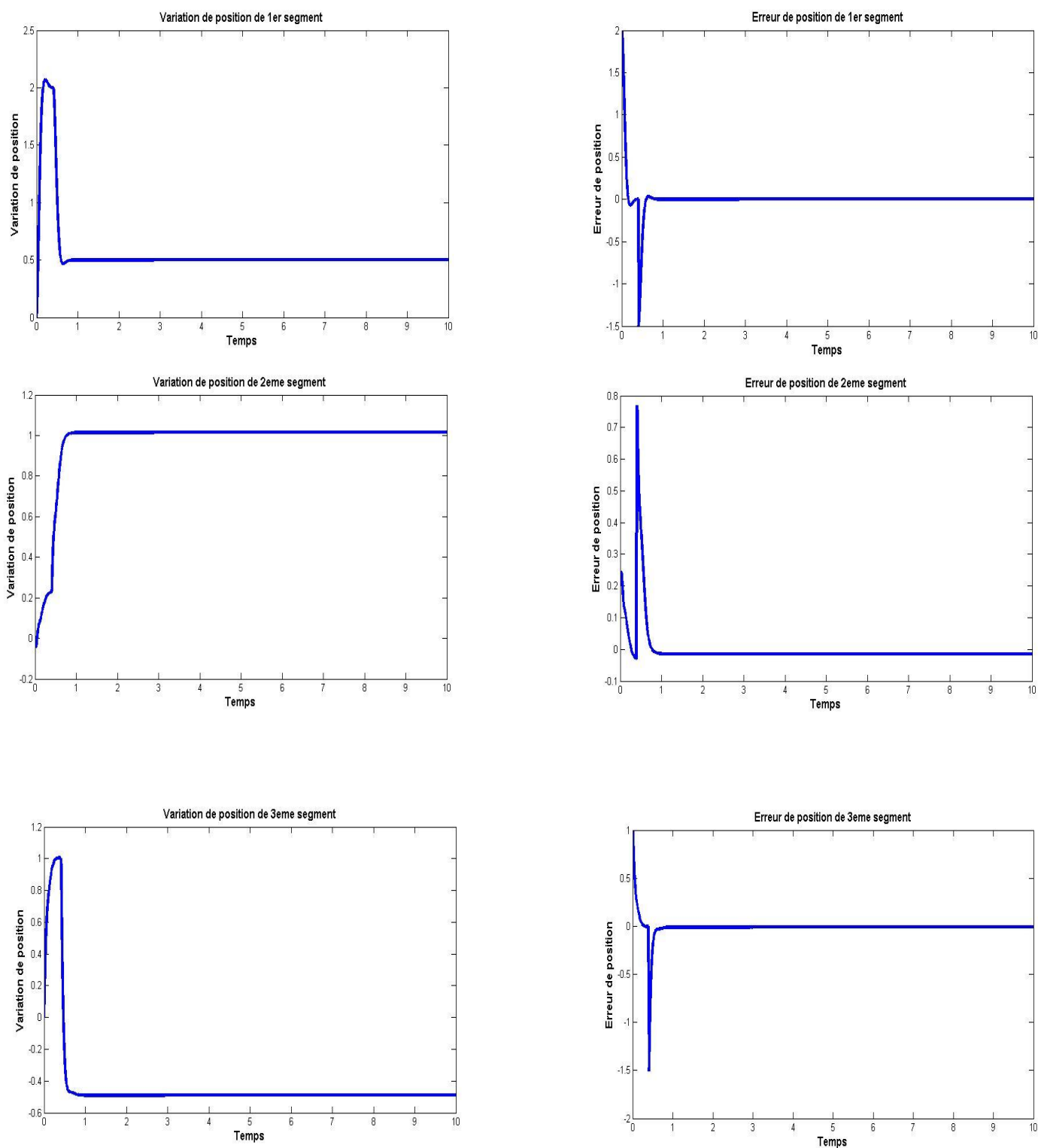


Figure.IV.11 Positions et erreurs des différents segments par la loi de commande PD



## Chapitre IV : Commande du Bras Manipulateurs

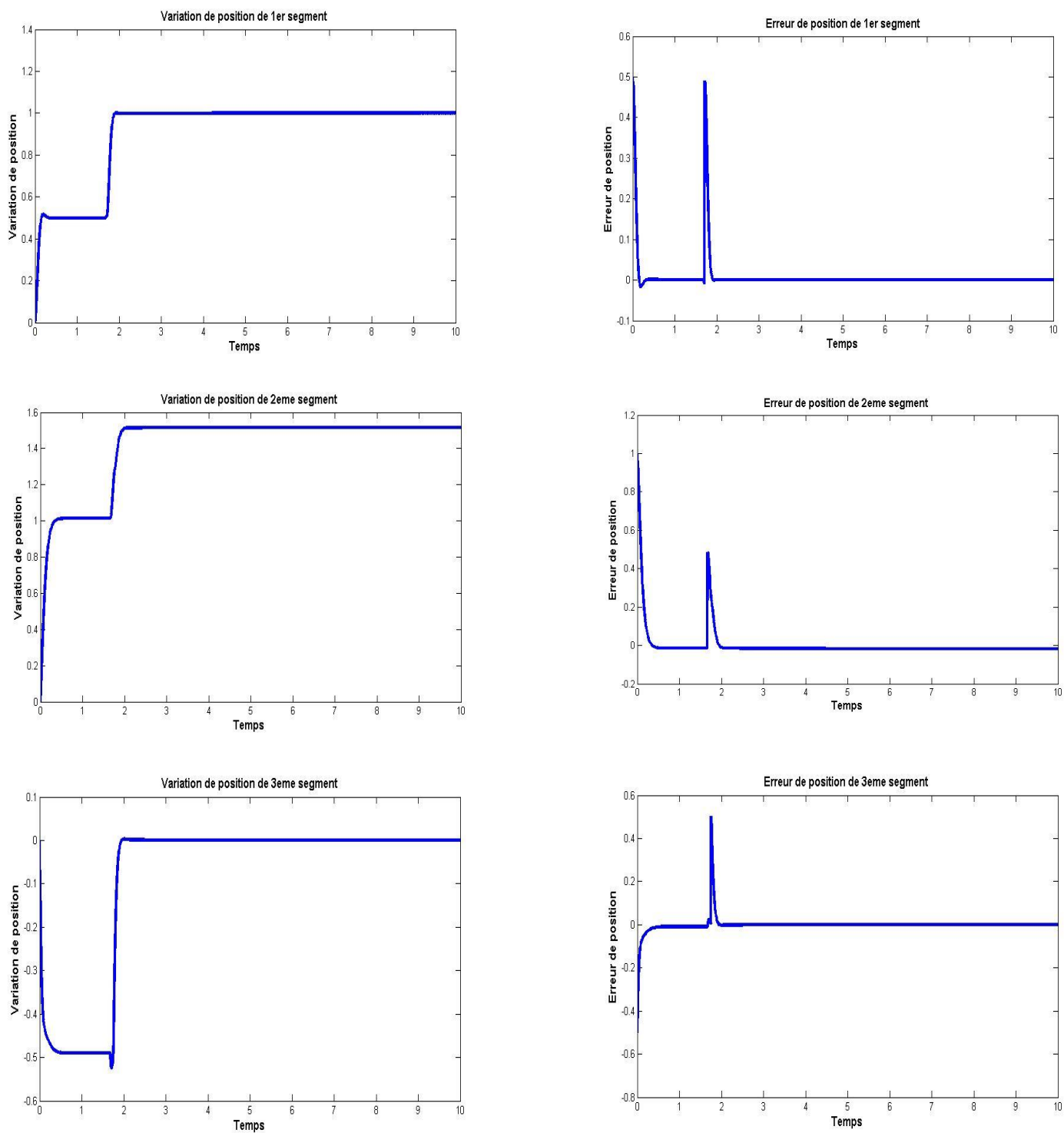


**Figure.IV.13** Positions et erreurs des différents segments par la loi de commande PD en temps réel

La figure suivante montre les Positions et les erreurs des différents segments par la loi de commande PD en temps réel avec changement de repère

## Chapitre IV : Commande du Bras Manipulateurs

Les positions Désirées Sont :  $q_{d1} = [0.5 \quad 1 \quad -0.5]$   $q_{d2} = [1 \quad 1.5 \quad 0]$



**Figure.IV.14** Positions et erreurs des différents segments par la loi de commande PD en temps réel avec changement de repère



## **Chapitre IV : Commande du Bras Manipulateurs**

### **IV.9 Conclusion**

Dans la première partie de ce chapitre, on s'est intéressé à la carte de commande Arduino et à son fonctionnement en décrivant les types de programmes implémentés à notre bras manipulateur, puis on a exposé leurs tests et leurs résultats de simulation. Dans la seconde partie, on s'est penché sur la commande du bras manipulateur en boucle fermée par une simulation avec la commande PD sous matlab/simulink en temps réel

## **Conclusion Générale**

### **Conclusion Générale**

Ce mémoire présente notre travail qui concerne l'étude et la simulation d'un bras manipulateur à trois degrés de liberté, avec l'hypothèse d'un environnement de travail sans obstacles. Après une exploration générale du domaine de la robotique, on a abordé le vif du sujet en étudiant les différents organes constituant notre bras manipulateur. Par la suite, on s'est penché sur la modélisation et l'élaboration des modèles géométrique, cinématique et dynamique, qui ont pour rôle respectivement de générer des trajectoires de références, d'analyser les singularités et de commander et simuler le robot.

Durant notre travail, on a gagné beaucoup de connaissance sur le domaine de modélisation et commande des robots travaillons sur ce projet a aussi nos introduit aux plusieurs ressources de recherche disponible sur ce champ.

## Bibliographie

### Bibliographie

- [1] Dictionnaire Reverso, 2010, Edition en ligne.
- [2] Dictionnaire Larousse, 2008, Edition en ligne.
- [3] EYRAUD, Charles-Henri. « Horloges astronomiques au tournant du XVIIIe siècle : de l'à-peu-près à la précision ». Thèse de doctorat. Lyon 2, 2004.
- [4] Official website IFR INTERNATIONAL FEDERATION OF ROBOTICS.
- [5] RICHA Braham BOUYEKHF mouhamed Redha . « Etude et réalisation d'un bras robot à 2DDL » Mémoire Master. Université Djilali Bounaama KHEMIS MILIANA.2017
- [6] Fodil Rosa et Sayah Kenza « Réalisation et Commande d'un Bras Manipulateur à Quatre Degrés de Liberté », Mémoire de Master UNIVERSITE M'HAMED BOUGARABOUMERDES. Juin 2017
- [7] EL HAMZAOUI Oussama « Localisation et Cartographie Simultanées pour un robot mobile équipé d'un laser à balayage : CoreSLAM », Thèse de Doctorat, École nationale supérieure des mines de Paris, 25 septembre 2012.
- [8] BOSTELMAN Roger « PERFORMANCE MEASUREMENT OF MOBILE MANIPULATORS », Thèse de Doctorat, UNIVERSITY OF BURGUNDY, 16 mars 2018.
- [9] BOIMOND, Jean-Louis. « ROBOTIQUE » Cours, ISTIA, Université Angers, 2017.
- [10] AMOURI AMMAR « modélisation dynamique d'un robot parallèle forme de plusieurs modules empilés » mémoire de MAGISTER, Université Larbi Ben M'hidi Oum-ElBouaghi, Faculté Des Sciences et Sciences Appliquées, 2011.
- [11] BOUKLI HACENE Lotfi Fazil « Commande et supervision d'un ensemble de robots via internet », Mémoire en vue de l'obtention de Magister, Université Des Sciences Et De La Technologie D'oran, 26 avril 2012.
- [12] BAHLOUL Abdelkrim « Sur la commande des robots manipulateurs industriels en comanipulation robotique », Thèse de Doctorat, Université Paris-Saclay, 7 décembre 2018.

## Bibliographie

- [13] LE BARS Fabrice « Analyse par intervalles pour la localisation et la cartographie simultanées ; Application à la robotique sous-marine », Thèse de Doctorat, UNIVERSITÉ DE BRETAGNE OCCIDENTALE, 23 novembre 2011.
- [14] FATOUX Julien « Mise en œuvre d'un robot humanoïde et contribution à la génération de marches dynamiques optimales », Thèse de Doctorat, UNIVERSITE DE POITIERS, 7 août 2006.
- [15] FISETTE P., BUYSE H., SAMIN J.C. « Introduction à la robotique » Cours, 19 février 2004.
- [16] RASHIDIFAR, Mohammad Amin, RASHIDIFAR, Ali Amin, et AHMADI, Darvish. «Modeling and Control of 5DOF Robot Arm Using Fuzzy Logic Supervisory Control », Article, vol. 2, no 2, 2089-4856, Janvier 2013, International Journal of Robotics and Automation (IJRA).
- [17] CRAIG, John J. « Introduction to robotics: mechanics and control », Pearson Education International, Livre, 2009.
- [18] EL-HUSSIENY Haitham « Robotics Engineering », Cours, Benha University, 2017.
- [19] GANGLOFF Jacques « Cours de Robotique », Cours, ENSPS 3A Master ISTI, 2018.
- [20] ANGELES Jorge « Fundamentals of robotic mechanical systems », Springer, Livre, 2002.
- [21] MENDIL B. « Cours de Robotique », Cours, Université Abderrahmane Mira, Béjaia, 2017.
- [22] FERRERO F. « Moteurs et transistors MOS », Cours, Université de Nice, 2017.
- [23] DATASHEET « MG90S Metal Gear Servo », Components 101, 2019.
- [24] DATASHEET « SG90 Servo Motor », Components 101, 2017.
- [25] BETTAIEB Mohamed Nizar « Modélisation du comportement mécanique d'engrenages. Couplages entre engrènement et éléments de structures flexibles », Thèse de Doctorat, Ecole Doctorale Des Sciences Pour L'Ingénieur De Lyon, 16 juin 2006.

## Bibliographie

- [26] DATASHEET « HC-05 Bluetooth to Serial Port Module », Components 101, 2018.
- [27] GAUTRON Laurent « PHYSIQUE TOUT LE COURS EN FICHES », Dunod, Livre, 2015.
- [28] PASQUIER Claude « Mécanique », Cours, Polytech Paris-Sud,2012.
- [29] LAFOND Roger « Analyse dynamique du mouvement », Cours,2018.
- [30] BOUQUET Frédéric, BOBROFF Julien « Microcontrôleur arduino », Cours, Université Paris Sud, printemps 2015.
- [31] RAO Michael « Introduction à l'Arduino », Cours, Université de Lyon,2016.
- [32] DATASHEET « Arduino UNO », Components 101, 2018.
- [33] « MG996R High Torque MetalGear Dual Ball Bearing Servo » [En ligne]. Disponible : [http://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](http://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf)
- [34] LAKHDARIF « Introduction à la simulation et routage des circuits avec le logiciel PROTEUS V7 et V8 », Cours, Université Des Sciences Et De La Technologie D'oran Faculté Du Génie Electrique Département D'électronique Licence, 2016/2017.
- [35]. [www.ni.com](http://www.ni.com) « site officiel de LABVIEW ».
- [36] [<https://arduino.blaisepascal.fr/presentation/>]
- [37] UNO[<https://bentek.fr/2-arduino-uno/>]
- [38] BENNAI Abdallah, « Etude et réalisation d'un bras de robot à base de carte Arduino piloter par interface graphique en labview », mémoire master , université Bechar , 2022
- [39 ] BOUQUET Frédéric, BOBROFF Julien « Microcontrôleur arduino », Cours, Université Paris Sud, printemps 2015.
- [40] : « Proteus (ISIS et ARES) » dans Elektronique. [En ligne]. Disponible: <http://www.elektronique.fr/logiciels/proteus.php>

## **Bibliographie**

[41] Fodil Rosa et Sayah Kenza « Réalisation et Comman de d'un Bras Manipulateur à Quatre Degrés de Liberté », Mémoire de Master UNIVERSITE M'HAMED BOUGARABOUMERDES. Juin 2017

[42] DURAND-ESTEBE Baptiste « Optimisation Energétique du Rafraichissement des Datacenters », Thèse de Doctorat, Ecole Doctorale Des Sciences Physiques Et De L'Ingénieur, 4 juillet 2014



