

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 سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière Automatique  
Spécialité Automatique et systèmes

Présenté par

EL BESSEGHI Fatma Zohra

# Asservissement d'un bras manipulateur mobile par vision artificielle

Proposé par : Pr. KAZED Boualem

Année Universitaire 2018-2019

## Remerciements

---

*Louange à « Dieu » le tout puissant de m'avoir permis de réaliser ce modeste travail dans de très bonnes conditions, m'accordant toute l'énergie qu'il me fallait.*

*Je tiens à exprimer ma profonde gratitude à mon promoteur Mr. **B. Kazed**, pour m'avoir orienté et prodigué ses précieux conseils lors de la réalisation de ce projet.*

*De même, je remercie chaleureusement le jury qui a dû consacrer une partie de son temps à examiner ce projet, ainsi que toutes les personnes ayant collaboré de près ou de loin à cet accomplissement.*

*Enfin, je saisis l'occasion pour exprimer ma reconnaissance à tous les enseignants qui ont assuré ma formation durant mon parcours universitaire.*

---

## ملخص :

يتمثل العمل الذي ناقشه في هذه المذكرة، في إنجاز ذراع مناور متحرك ذو ثلاث درجات حرية و التحكم فيه بواسطة الرؤية الاصطناعية.

قمنا باستخدام كاميرا ذكية التي تلتقط الكرة (بفضل خاصية التعرف على الألوان) التي يتوجه نحوها الروبوت المتحرك، بمجرد الوصول إلى الكرة، يقوم الذراع المناور بالبحث عنها ليتمكن من إلتقاطها.

**كلمات المفاتيح:** ذراع مناور، درجات حرية، الرؤية الاصطناعية، كاميرا ذكية ، روبوت متحرك

---

**Résumé :** Le travail qu'on rapporte dans ce mémoire consiste en la réalisation et l'asservissement d'un bras manipulateur mobile à trois degrés de liberté puis en l'élaboration d'une commande par vision artificielle pour ce dernier. On a utilisé une caméra intelligente qui capte la balle (par reconnaissance de couleurs) vers laquelle le robot mobile va se déplacer, une fois arrivé, le bras manipulateur est commandé afin de pouvoir attraper la balle captée par la caméra et la mettre dans une position préprogrammée.

**Mots clés :** Bras manipulateur ; degrés de liberté ; vision artificielle ; caméra intelligente ; robot mobile.

---

**Abstract:** The work that we report in this thesis, consists of the realization and the enslavement of a three freedom's degrees mobile manipulative arm and then by drawing up a command by artificial vision for the latter. We used a smart camera that captures (by colors recognition) the ball to which the mobile robot will move, once arrived, the manipulative arm is controlled in order to catch the ball captured by the camera and put it in a programmed position.

**Keywords:** Manipulator arm; freedom's degrees; artificial vision; smart camera; mobile robot.

---

## Liste des acronymes et abréviations

ATLIF : Analyse et Traitement Informatique de la Langue Française

BM : Bras manipulateur

BO : Boucle Ouverte

DC : Direct current

DDL : Degrés De Liberté

D-H : Denavit-Hartenberg

HSV: (Hue-Saturation-Value)

I<sup>2</sup>C : Inter-Integrated Circuit

ISO : International Standard Organization

JIRA : Japanese Industrial Robotic Association

JPEG : Joint Photographic Experts Group

MCC : Machine/Moteur à Courant Continu

MCD : Modèle Cinématique Direct

MCI : Modèle Cinématique Inverse

MISO : Master Input Slave Output

MOSI : Master Output Slave Input

NEMA : National Electrical Manufacturers Association

PAP : Pas-à-pas

PID : Proportionnal Integral Derivative

PWM : Pulse Width Modulation

RGB : (Red-Green-Blue)

RIA : Robot Institute of America

RVB : (Rouge-Vert-Bleu)

SDI : Serial Data In

SDO : Serial Data Out

SCARA : Selective Compliant Assembly Robot Arm

SCK : Serial Clock

SPI : Serial Peripheral Interface

SS : Slave Select

UART : Universal Asynchronous Receiver Transmitter

# Table des matières

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre 1 Généralités sur la robotique.....</b>	<b>3</b>
1.1 Introduction .....	3
1.1.1 Apparition des robots manipulateurs (historique) .....	3
1.1.2 Définition des robots manipulateurs .....	4
1.2 Catégories des robots .....	4
1.2.1 Les manipulateurs.....	5
1.2.2 Les télémanipulateurs.....	5
1.2.3 Les robots .....	6
1.3 Structure générale d'un robot manipulateur.....	7
1.3.1 Morphologie générale.....	7
1.3.2 Mécanisme .....	8
1.3.3 Les liaisons mécaniques .....	9
1.3.4 Degrés de liberté.....	11
1.4 Caractéristiques d'un robot .....	12
1.5 Programmation des robots .....	12
1.5.1 L'apprentissage.....	12
1.5.2 Utilisation des logiciels.....	13
1.6 Domaines d'application .....	13
1.6.1 Tâches simples.....	13
1.6.2 Tâches complexes .....	13
1.7 Robotique mobile .....	14
1.7.1 Manipulateur mobile .....	14
1.7.2 Exemples d'application .....	15
1.8 Vision artificielle .....	16
1.8.1 Traitement de l'image.....	16
1.8.2 L'image .....	16
1.8.3 Le modèle RVB (RGB).....	17
1.8.4 Le modèle de couleurs HSV .....	18
1.8.5 Système de vision .....	18
1.9 Conclusion.....	19
<b>Chapitre 2 Modélisation du bras manipulateur .....</b>	<b>20</b>
2.1 Introduction .....	20

2.2	Modélisation d'un bras manipulateur .....	20
2.2.1	Définition d'un bras manipulateur.....	20
2.2.2	Nécessité d'un modèle.....	21
2.2.3	Transformations homogènes .....	21
2.2.4	Modélisation cinématique .....	23
2.3	Cas du robot SCARA.....	26
2.3.1	Présentation du bras manipulateur SCARA.....	26
2.3.2	Structure de l'étude .....	27
2.3.3	Simulation avec Matlab.....	30
2.4	Conclusion.....	33
<b>Chapitre 3</b>	<b>Description de la partie matérielle du robot .....</b>	<b>34</b>
3.1	Introduction .....	34
3.2	La partie du bras manipulateur .....	34
3.2.1	Caractéristiques techniques du bras manipulateur .....	34
3.2.2	Actionneurs .....	35
3.3	La partie mobile.....	38
3.3.1	Mécanique du robot mobile.....	38
3.3.3	Asservissement du robot mobile .....	49
3.4	La partie vision .....	51
3.4.1	Présentation de la Pixy2.....	51
3.4.2	Apprentissage de Pixy2 .....	52
3.4.3	Applications de la Pixy2.....	54
3.5	Conclusion.....	55
<b>Chapitre 4</b>	<b>Tests et résultats .....</b>	<b>57</b>
4.1	Introduction .....	57
4.2	Commande du bras manipulateur.....	57
4.2.1	Validation de la cinématique inverse sur Arduino.....	57
4.2.2	Positionnement du BM à des coordonnées préprogrammées .....	60
4.2.3	Calibrage de la caméra.....	64
4.2.4	Calibration des axes $x$ , $y$ et $z$ du bras manipulateur .....	67
4.3	Commande du robot mobile.....	72
4.4	Organigramme du programme utilisé .....	74
4.5	Conclusion.....	75
<b>Conclusion générale</b>	<b>.....</b>	<b>76</b>

**Annexes 78**

**Bibliographie 82**

# Liste des figures

## Chapitre 1 : Généralités sur la robotique

Figure 1.2.2 : Architecture type d'un télémanipulateur maître-esclave.....	5
Figure 1.2.3.a : Robot manipulateur industriel.....	6
Figure 1.2.3.c: Robot mobile.....	7
Figure 1.3.1: Constituants d'un robot.....	8
Figure 1.3.2.a : Structure série.....	8
Figure 1.3.2.b : Structure parallèle.....	9
Figure 1.3.2.c : Structure mixte.....	10
Figure 1.3.3.a : Symbole de l'articulation rotoïde.....	10
Figure 1.3.3.b : Symbole de l'articulation prismatique.....	10
Figure 1.3.4.b : Orientation d'un repère en utilisant les angles d'Euler.....	11
Figure 1.6.1 : Robot de soudage.....	13
Figure 1.6.2.a : Robot de construction.....	14
Figure 1.6.2.b : Robot d'assistance aux handicapés.....	14
Figure 1.7.2 : Exemples de robots mobiles commerciaux ou de recherche.....	15
Figure 1.8.2 : Image en niveaux de gris et sa matrice de valeurs de niveaux d'intensité.....	16
Figure 1.8.3 : Le modèle de couleur RVB représenté sur un cube.....	17
Figure 1.8.4 : La cône du modèle de couleur HSV.....	18

## Chapitre 2 : Modélisation du bras manipulateur

Figure 2.2.1 : Robot à structure simple ouverte.....	20
Figure 2.2.3 : Transformation de repère.....	23
Figure 2.2.4 : Illustration du passage entre l'espace articulaire et cartésien.....	24
Figure 2.2.4.b: Paramètres de Denavit-Hartenberg.....	25
Figure 2.3.1.a : Le robot SCARA.....	27
Figure 2.3.1.b : Le bras SCARA (schéma).....	27
Figure 2.3.2.a.1 : Le placement des repères du robot SCARA.....	28

Figure 2.3.2.a.2 : Illustration des articulations rotoïdes du BM.....	29
Figure 2.3.3.c.1 : Illustration des articulations rotoïdes du BM sur MATLAB.....	31
Figure 2.3.3.c.2 : Allures des angles $\theta_1$ et $\theta_2$ affichées sur MATLAB .....	31
Figure 2.3.3.c.3 : Articulations rotoïdes à l'état initial .....	32
Figure 2.3.3.c.4 : Articulations rotoïdes à l'état final (coordonnées d'un cercle) .....	32
<b>Chapitre 3 : Description de la partie matérielle du robot</b>	
Figure 3.2.1 : Image réelle du bras manipulateur SCARA conçu.....	35
Figure 3.2.2.a : Modèle du moteur PAP utilisé.....	36
Figure 3.2.2.b: Le schéma de câblage pour la connexion d'un microcontrôleur et le driver a4988.....	36
Figure 3.2.2.c.1 : Moteur pap NEMA-17.....	37
Figure 3.2.2.c.2 : Effecteur final .....	37
Figure 3.2.2.d : Micro pompe à air.....	37
Figure 3.3.1 : Les différentes pièces utilisées en la mécanique du robot.....	38
Figure 3.3.1.a : Roues à bille.....	39
Figure 3.3.1.b.1 : Schéma d'un moteur type EMG30.....	39
Figure 3.3.1.b.2 : Moteur DC (EMG30) .....	40
Figure 3.3.1.b.3 : Codeur incrémental.....	40
Figure 3.3.1.c : Codeur incrémental à effet Hall.....	41
Figure 3.3.2.a : Carte de puissance utilisée.....	42
Figure 3.3.2.a.1 : Le pont en H.....	42
Figure 3.3.2.a.2 : Sens du courant en fonction de l'état des interrupteurs.....	43
Figure 3.3.2.a.3 : Pont en H avec transistors NPN, diodes et entrées Arduino.....	44
Figure 3.3.2.a.4 : L298 H bridge.....	44
Figure 3.3.2.b.1 : Cartes utilisée (pour la commande du robot) .....	45
Figure 3.3.2.b.2 : Carte Arduino MEGA.....	45
Figure 3.3.2.c : Carte RAMPS.....	46
Figure 3.3.2.d.1 : Schéma de câblage d'un SPI.....	47

Figure 3.3.2.d.2 : Architecture type d'un bus I2C.....	47
Figure 3.3.3 : Schéma synoptique du système conçu (robot).....	49
Figure 3.4.1.a : Caractéristiques de la Pixy2.....	51
Figure 3.4.1.b : Le capteur caméra Pixy2.....	52
Figure 3.4.3 : L'emplacement de la caméra dans le robot mobile.....	55
<b>Chapitre 4 : Tests et résultats</b>	
Figure 4.2.1.a : Programme Arduino utilisé.....	58
Figure 4.2.1.b : Valeurs des angles $\theta_1$ et $\theta_2$ affichées sur le moniteur série.....	58
Figure 4.2.1.c : Allure du cercle dessiné par le bras manipulateur.....	59
Figure 4.2.1.d : Trajectoire circulaire poursuivie par le BM.....	60
Figure 4.2.3.a : Filtrage médian.....	65
Figure 4.2.3.b : Courbe de la surface en fonction de la distance.....	66
Figure 4.2.3.c : Capture de la balle à une distance de 700mm.....	67
Figure 4.2.3.d : Capture de la balle à une distance de 600mm.....	67
Figure 4.2.4.a : Allure de $x_{calculé}$ et $x_{bras}$ .....	70
Figure 4.2.4.b : Allure de $y_{calculé}$ et $y_{bras}$ .....	71
Figure 4.2.4.c : Allure de $(x_{calculé}, y_{calculé})$ et $(x_{bras}, y_{bras})$ .....	71
Figure 4.3 : Vitesse des roues gauche et droite en fonction de la position de la balle.....	72

## Liste des tableaux

Tableau 2.3.2.b : Paramètres DH du robot SCARA.....	28
Tableau 3.2.1 : Caractéristiques techniques du bras manipulateur SCARA.....	34
Tableau 3.3.1.b.1 : Connecteurs du codeur.....	40
Tableau 3.3.1.b.2 : Caractéristiques du moteur DC de type EMG30.....	40
Tableau 3.3.2 : Avantages et inconvénients des bus SPI et I <sup>2</sup> C.....	48
Tableau 4.2.3 : Coordonnées x, y et z (utilisées dans le programme).....	61

# Introduction générale

---

L'objectif principal de ce projet consiste en la réalisation d'un bras manipulateur mobile. Les robots font partie intégrante de notre monde. Ils ont également conquis notre imaginaire, hantant nos univers de science-fiction jusqu'à se matérialiser sur nos écrans de par des capacités extraordinaires. Cependant, la réalité est toujours ordinaire.

La grande majorité des robots manipulateurs que nous pouvons trouver aujourd'hui restent confinés dans des usines, afin de réaliser des tâches répétitives et dangereuses pour l'homme. Ayant un support fixe, ils agissent dans un environnement limité qui dépend de la taille et des valeurs maximales que peuvent atteindre leurs différentes articulations.

En ce qui concerne ce projet nous allons concevoir un bras manipulateur de type SCARA qui sera monté sur une plateforme mobile afin de lui permettre d'augmenter considérablement ses marges de manœuvre. Le robot proposé sera muni d'une caméra embarquée qui sera utilisée pour indiquer les coordonnées de l'objet à atteindre. Dans un premier temps la position et la taille, de l'objet capté par cette caméra, seront utilisés pour diriger la partie mobile du robot pour lui permettre de s'approcher de l'objet afin que, par la suite, les coordonnées de cet objet seront calculées dans le repère du bras, qui prendra la relève pour faire déplacer son effecteur final vers l'objet en question. Les coordonnées, dans l'espace de l'objet recherché seront déterminés grâce à une calibration adéquate des données issues de la caméra, l'opération suivante sera de calculer les deux angles de rotation du bras ainsi que le déplacement linéaire du troisième axe, ceci sera réalisé en utilisant la cinématique inverse dont les détails seront expliqués dans le chapitre 2. La dernière étape que devra accomplir notre bras sera d'appréhender l'objet ciblé et le déposer à un point, dans l'espace, dont les coordonnées

seront préprogrammées. Pour réaliser cela nous allons utiliser une ventouse qui sera alimentée par une pompe d'aspiration activée et désactivée au moments opportuns.

Ce mémoire s'articule autour de quatre chapitres :

- Pour le chapitre 1, on a parlé sur la robotique en général, on a aussi défini les robots et on a présenté leurs différentes catégories, leurs domaines d'utilisation ...etc. Une brève partie sur la vision artificielle a été présentée dans ce chapitre.
- Le chapitre 2 traite la modélisation des bras manipulateurs. L'utilisation de la modélisation cinématique directe et inverse est expliquée dans cette partie.
- Le chapitre 3 présente l'architecture robotique. La description de la mécanique du robot mobile et du bras manipulateur et l'électronique utilisée sont expliquées dans cette partie.
- Le chapitre 4 contient les tests effectués et les résultats obtenus.

Au final, on a clôturé avec une conclusion générale dans laquelle on porte un regard critique sur l'ensemble des travaux réalisés et on finit cet exposé par quelques propositions de travaux.

# Chapitre 1 Généralités sur la robotique

---

## 1.1 Introduction

La **robotique** peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles.

Ce chapitre traite d'une façon générale la robotique, les robots manipulateurs et leurs intérêts dans différents domaines : spatial, médical, industriel...etc.

### 1.1.1 Apparition des robots manipulateurs (historique)

- 1920 : Le mot robot dérivé du tchèque « *robota* » (travailleur) est apparu dans la littérature scientifique et technique de la pièce de théâtre de l'écrivain tchèque Karel Capel, RUR (Rossum's Universal Robot). [1]
- 1947 : Premier manipulateur électrique téléopéré.
- 1954 : Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société **UNIMATION** (USA), sur une chaîne de montage de **General Motors**.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot. [2]

Dès 1971, de grands laboratoires de recherches commencent à se développer tels que la **JIRA** (Association Japonaise de Robotique Industrielle) au Japon, la **RIA** (Robot Institute of America) aux USA en 1975. En 1998, on comptait 700 000 robots industriels dans le monde.

De nos jours, les bras manipulateurs constituent les composants les plus importants des processus de fabrication et de contrôle. Ils ont comme impact l'amélioration, la productivité, l'accroissement de la qualité des produits fabriqués ainsi que la réduction du coût du travail. [3]

### **1.1.2 Définition des robots manipulateurs**

L'**ATILF** (Analyse et Traitement Informatique de la Langue Française) définit le robot comme « un appareil effectuant, grâce à un système de commande automatique à base de micro-processeur une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique » [1]. Cette définition décrit le principe de fonctionnement de tous les robots existants, mais ne donne aucune information sur les tâches effectuées ou sur leur aspect visuel. Les robots se présentent sous des formes diverses et variées, les morphologies choisies, étant à *priori* adaptées aux tâches qu'ils devront effectuer.

Selon la **RIA** (Robot Institute of America) c'est un manipulateur qui doit être reprogrammable, multifonctionnel conçu pour déplacer des matériaux, des pièces, des outils ou tout autre dispositif spécialisé au moyen d'une série de mouvements programmés et d'accomplir une variété d'autres tâches. L'**ISO** (International Standard Organization) l'a défini comme étant une machine mue 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 dispositifs d'inspection.

Pour le sens commun, un robot est un dispositif mécanique articulé capable d'imiter certaines fonctions humaines telles que la manipulation d'objets ou la locomotion, dans le but de remplacer l'homme pour la réalisation de certaines tâches matérielles. [3]

## **1.2 Catégories des robots**

La classification des robots est délicate selon leurs fonctionnalités et leurs potentialités. Les robots peuvent être classés en trois grandes catégories :

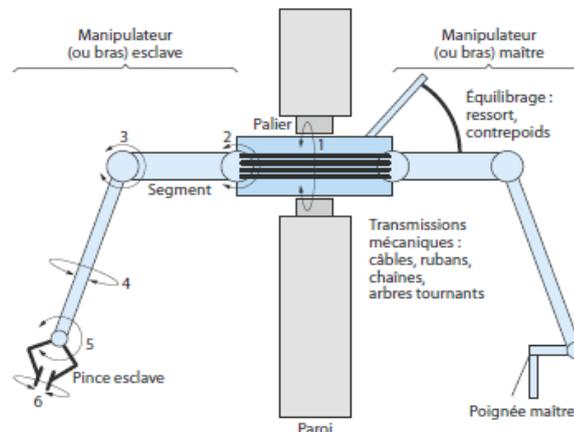
### 1.2.1 Les manipulateurs

Ce sont des systèmes mécaniques destinés à exécuter de manière autonome des tâches répétitives dans un environnement fixe et ordonné et dont les mouvements enregistrés dans une mémoire se répètent de manière cyclique, aucune fonction de décision ne lui permet de réagir dans le cas d'un événement inattendu. [4]

### 1.2.2 Les télémanipulateurs

Un télémanipulateur ou robot de téléopération est un système mécanique poly-articulé et multifonctionnel capable d'assister l'homme dans les opérations effectuées en milieu hostile ou suppléer un handicapé. Il est constitué de 2 entités :

- Un bras maître et un bras esclave. Le bras esclave reproduit instantanément le mouvement imposé par un opérateur à bras maître. Ce dernier est constitué d'une boîte à boutons, d'une poignée de commande ou d'une structure articulée identique à celle du bras esclave.



**Figure 1.2.2 :** Architecture type d'un télémanipulateur maître-esclave

Malgré la présence indispensable d'un opérateur, ce système est doté de potentialité comparable aux robots évolués : [4]

- Prise de décision lors de vision réduite.
- Retour d'effort vers le bras maître.
- Amplification ou atténuation des mouvements.
- Autonomie de l'esclave pour alléger la charge du travail à l'homme.

- Création de mouvements élémentaires autonomes.

L'application des télémanipulateurs est répandue dans les domaines nucléaires (manipulations des produits radioactifs), spatiaux (exploration des planètes), sous-marins (forage), militaires (déménages) et médicaux (prothèses).

### 1.2.3 Les robots

- a. Les robots manipulateurs industriels* : ils constituent l'association d'un manipulateur et d'organes aptes à acquérir, à traiter et à gérer les informations issues de l'environnement. Ils sont dotés de sens artificiels (vision, tactile...) et de ressources méthodologiques leur permettant de résoudre des problèmes de difficultés toutefois limitées. [5]

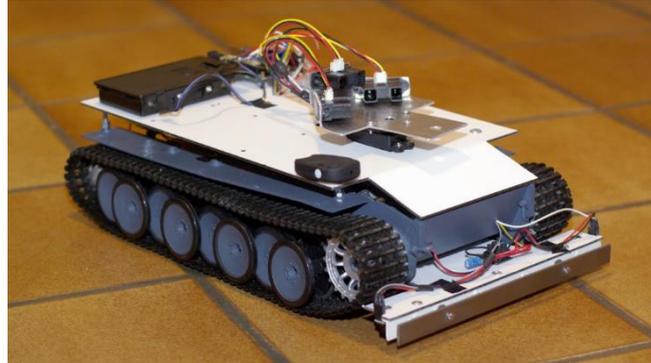


**Figure 1.2.3.a** : Robot manipulateur industriel

Ces robots sont essentiellement orientés vers les applications nécessitant une adaptabilité à la tâche : assemblage, soudage à l'arc, vérification, test ...etc.

- b. Les robots didactiques* : ils sont de format réduit, ces robots ont été conçus ces dernières années afin de mieux encadrer les apprentissages, la robotique pédagogique s'inscrivant dans une démarche d'accompagnement des apprenants pour un engagement optimal.
- c. Les robots mobiles autonomes* : ils peuvent être utilisés en zones dangereuses (nucléaire, incendie, sécurité civile ...etc.), inaccessibles (océanographie, spatial). De tels robots font appel à

des capteurs et à des logiciels sophistiqués. On peut distinguer deux types de locomotion : les robots marcheurs qui imitent la démarche humaine et les robots mobiles qui ressemblent plus à des véhicules.



*Figure 1.2.3.c:* Robot mobile

## **1.3 Structure générale d'un robot manipulateur**

### **1.3.1 Morphologie générale**

Un robot manipulateur est l'ensemble formé par :

- Une structure mécanique qui supporte l'organe terminal à situer.
  - Des actionneurs qui servent à agir sur la structure mécanique pour changer la situation de l'organe terminal.
  - Des capteurs divers nécessaires à la commande.
  - Un système de commande qui pilote les actionneurs du robot manipulateur.
- 
- Un système décisionnel qui assure la fonction de raisonnement et élabore le mouvement du robot.
  - Un système de communication qui gère le message transmis entre le système décisionnel et l'opération via une console de visualisation. [6]

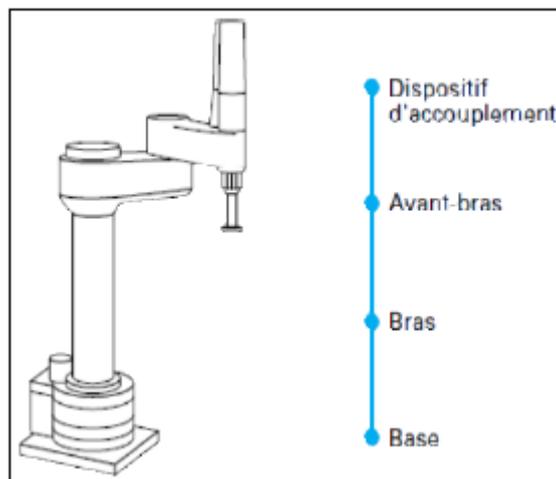


**Figure 1.3.1:** Constituants d'un robot

### 1.3.2 Mécanisme

On appelle *mécanisme* un ensemble de solides reliés 2 à 2 par des liaisons. Deux corps extrêmes ont des rôles particuliers : [6]

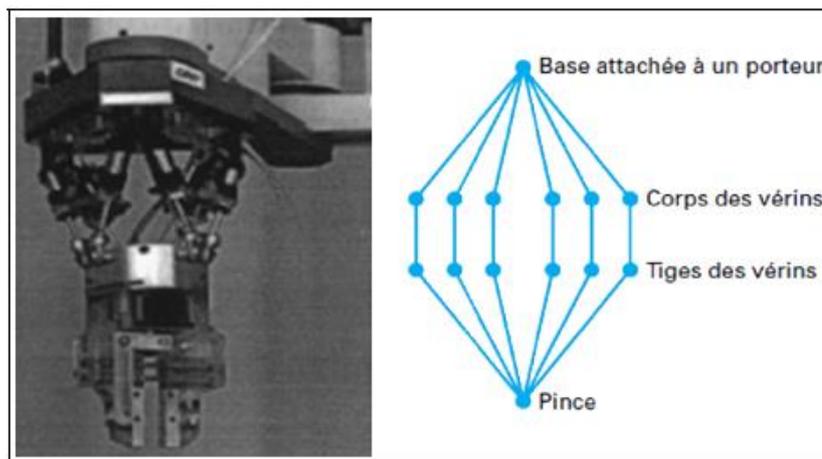
- La base : fixée au sol ou sur un véhicule.
- L'organe terminal qui porte l'outil (l'effecteur final).



**Figure 1.3.2.a :** Structure série

En partant de la base pour aller vers l'effecteur, on pourra distinguer :

- Les structures séries (Figure 1.3.2.a), ou sérielles (mécanisme en chaîne simple), pour lesquelles le graphe est arborescent, la base étant la racine et les feuilles étant les organes terminaux.
- Les structures parallèles (Figure 1.3.2.b), pour lesquelles toutes les chaînes partent de la base pour aller vers l'organe terminal.
- Les structures mixtes (Figure 1.3.2.c), présentant des boucles cinématiques, par exemple des parallélogrammes ou des motorisations par vérins linéaires. [6]

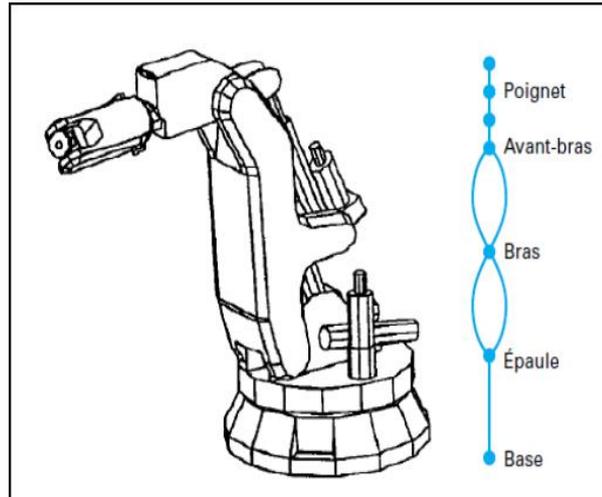


**Figure 1.3.2.b** : Structure parallèle

### 1.3.3 Les liaisons mécaniques

Dans l'assemblage des robots manipulateurs, les liaisons les plus courantes sont :

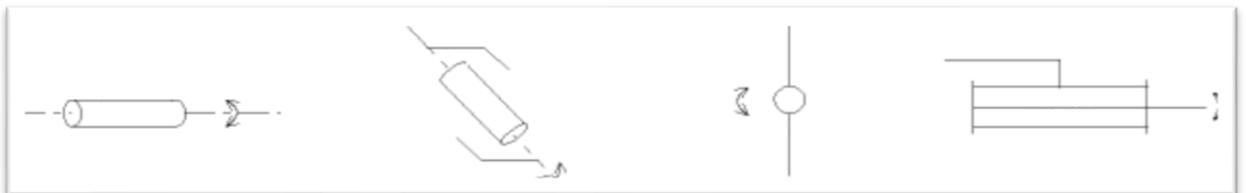
- La liaison rotoïde (ou pivot).
- La liaison prismatique (ou glissière).
- La liaison rotule S (ou sphérique).
- La liaison cardan U (joint universel).



**Figure 1.3.2.c** : Structure mixte

Certaines liaisons sont motorisées, on parlera de liaisons actives. Ce sera généralement le cas des liaisons rotoïdes ou prismatiques.

- a. Articulation rotoïde** : il s'agit d'une articulation de type pivot, notée **R**, réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle de cet axe. [3] (Figure 1.3.3.a)



**Figure 1.3.3.a** : Symbole de l'articulation rotoïde

- b. Articulation prismatique** : Il s'agit d'une articulation de type glissière, notée **P**, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance de cet axe. [3] (Figure 1.3.3.b)



**Figure 1.3.3.b** : Symbole de l'articulation prismatique

### 1.3.4 Degrés de liberté

#### a. Positionnement d'un solide dans l'espace :

Le positionnement complet d'une pièce dans l'espace nécessite six paramètres indépendants, appelés *degrés de liberté*. Parmi les différents choix possibles de ces paramètres :

- 3 paramètres indépendants définissent la position d'un point, noté  $P$ , du solide (coordonnées cartésiennes, cylindriques, sphériques, ..., dans la base du repère fixe).
- 3 paramètres indépendants déterminent l'orientation du solide autour du point  $P$  (angles d'Euler, paramètres d'Euler, ...).

#### b. Exemple d'orientation d'un repère (angles d'Euler) :

L'orientation d'un repère orthonormé (lié au solide) par rapport au repère fixe est spécifiée par 3 angles  $\psi$  (psi),  $\theta$  (thêta) et  $\varphi$  (phi), correspondant à une séquence de 3 rotations (Figure 1.3.4.b). De tels angles sont très utilisés en mécanique. Ils permettent une définition minimum de l'orientation (3 paramètres)

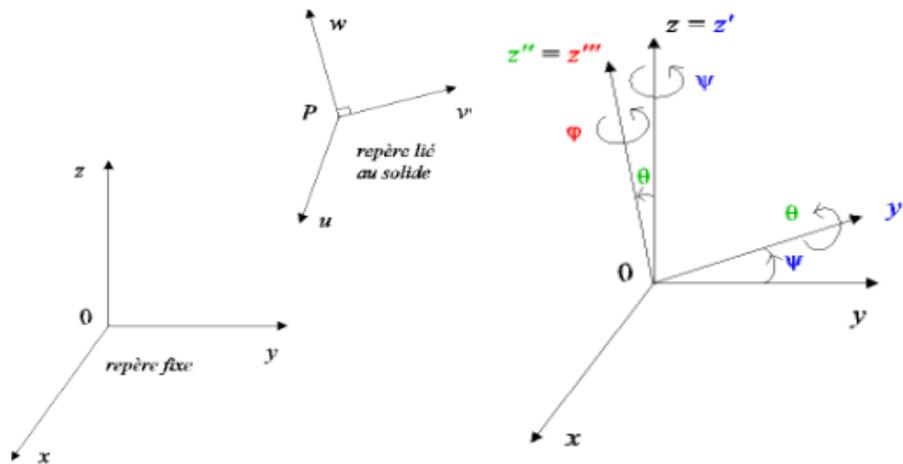


Figure 1.3.4.b : Orientation d'un repère en utilisant les angles d'Euler

Les angles sont définis dans le cas présent, selon la convention (z, y, z), comme suit :

- On fait une rotation d'un angle  $\psi$ , appelé *lacet* (*yaw* en anglais), autour de l'axe  $Oz$ ,
- On fait une rotation d'un angle  $\theta$ , appelé *tangage* (*pitch* en anglais), autour de l'axe  $Oy'$ ,
- On fait une rotation d'un angle  $\varphi$ , appelé *roulis* (*roll* en anglais), autour de l'axe  $Oz''$ . [7]

## 1.4 Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Les paramètres à prendre en considération sont essentiellement les suivants :

- La charge utile transportable par le robot.
- L'espace de travail qui est l'ensemble des stations de l'espace que l'organe terminal du robot peut atteindre.
- Les vitesses et les accélérations maximales qui conditionnent le temps du cycle.
- Les performances, l'écart entre une station commandée et la moyenne des situations atteintes.
- La masse du robot.
- Le coût du robot. [8]

## 1.5 Programmation des robots

Classiquement, deux étapes sont utilisées pour faire en sorte qu'un robot connaisse la tâche à exécuter [8] :

### 1.5.1 L'apprentissage

Enregistrement dans la mémoire de la trajectoire à exécuter, sous contrôle d'un opérateur humain.

- **Pantin** : Structure mécanique identique à celle du robot, qui est déplacée et qui mémorise les points « *pertinents* ».
- **Syntaxeur** : Une manche de pilotage (*joystick*) qui commande les déplacements de l'organe terminal.
- **Boite à boutons** : C'est un interrupteur par actionneur.

## 1.5.2 Utilisation des logiciels

Pour effectuer les opérations utilisées lors de la réalisation des trajectoires (tâche à réaliser) il est fait appel à un logiciel qui, à partir d'un modèle du robot et des trajectoires à réaliser élabore la succession des commandes des actionneurs.

## 1.6 Domaines d'application

Les bras manipulateurs peuvent être utilisés dans plusieurs domaines et peuvent accomplir des tâches différentes.

### 1.6.1 Tâches simples

La majorité des robots est utilisée pour des tâches simples et répétitives. Ils sont programmés une fois au cours de la procédure d'apprentissage.

Exemples de robots utilisés pour des tâches simples : robots soudeurs par points et robots soudeurs à l'arc. (Figure 1.6.1)



*Figure 1.6.1* : Robot de soudage

### 1.6.2 Tâches complexes

On les trouve généralement en robotique de service :

- Robot pompiste
- Robot de construction (Figure 1.6.2.a)
- Robot d'assistance aux personnes handicapées (Figure 1.6.2.b)



**Figure 1.6.2.a** : Robot de construction



**Figure 1.6.2.b** : Robot d'assistance aux handicapés

## 1.7 Robotique mobile

Au-delà des applications industrielles classiques, les robots sont de plus en plus présents dans notre quotidien et dans de grands domaines d'applications tels que la médecine, l'agriculture, la sécurité ou l'assistance à domicile. Ces robots sont également de plus en plus mobiles, capables d'évoluer aussi bien dans des milieux aériens ou maritimes que terrestres. La robotique mobile terrestre occupe une place historique importante notamment les robots mobiles à roues qui empruntent un mode de locomotion par roulement particulièrement efficace. La fonction principale d'un robot mobile est de transporter une charge utile – qui peut être un simple capteur ou un système de manipulation – à un endroit donné de l'environnement. Les robots terrestres, grâce à leur liaison au sol, créent une force de propulsion à cette interface qui leur permet de réaliser un déplacement. [9]

### 1.7.1 Manipulateur mobile

Un manipulateur mobile est un système composé de deux sous-systèmes, un ou plusieurs bras robotiques et une plateforme mobile. Un tel système combiné est capable

d'effectuer la tâche de manipulation dans un grand espace de travail comparé aux manipulateurs à base fixe.

### **a. Taille du robot**

La taille du robot représente un critère de sélection important. Dans des environnements fortement encombrés par exemple, les petits robots peuvent être d'un grand secours. Par contre, s'il y a nécessité de transport d'objets lourds, les grands robots conviennent mieux.

## **1.7.2 Exemples d'application**

Aujourd'hui, le marché commercial de la robotique mobile est relativement restreint à l'exception des robots aspirateurs vendus à plusieurs millions d'exemplaires. Cependant, il existe de nombreuses perspectives de développement qui en feront probablement un domaine important dans le futur. Les applications des robots peuvent se trouver dans de nombreuses activités "ennuyeuses, salissantes ou dangereuses" (3 D's en anglais pour **Dull, Dirty, Dangerous**), et dans des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées. [10]



**Figure 1.7.2 :** Exemples de robots mobiles commerciaux ou de recherche

## 1.8 Vision artificielle

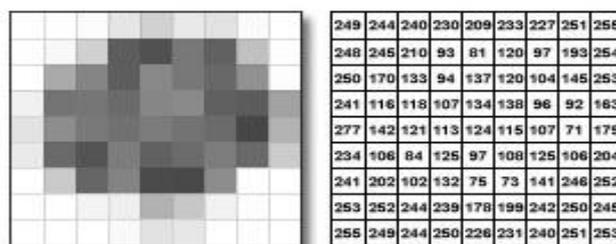
Dans de nombreuses branches, le mariage de la robotique et de la vision artificielle permet une fabrication et une assurance qualité irréprochables, rapides, fiables et économiques de produits les plus divers. La vision artificielle est la science de l'extraction d'informations d'images provenant d'une caméra, allant de points dans l'image à la reconstruction tridimensionnelle de la scène observée. La richesse des informations apportées par une caméra en fait un capteur de choix pour les applications de robotique autonome. Contrairement à d'autres capteurs dont la mesure donnée est, par exemple, directement une distance, l'image (donnée de la caméra), quant à elle doit être traitée.

### 1.8.1 Traitement de l'image

Le traitement d'image (*image processing*) consiste à appliquer des transformations mathématiques sur des images dans le but d'améliorer leur qualité ou d'en extraire une information. On peut citer à titre d'exemple : la création d'effets spéciaux au cinéma, l'amélioration de la qualité des photos en corrigeant les conditions de prise de vue, la détection de lettres ou de visages, l'identification des zones cancéreuses en imagerie médicale, la compression **JPEG** ...etc.

### 1.8.2 L'image

Une image est une fonction dimensionnelle  $f(x, y)$  où  $x$  et  $y$  sont des coordonnées spatiales (plan). L'intensité ou le niveau de gris de l'image au point de coordonnées  $(x, y)$  est l'amplitude de  $f$  à ce point. Cette image est appelée image en niveaux de gris ou image de niveau d'intensité.



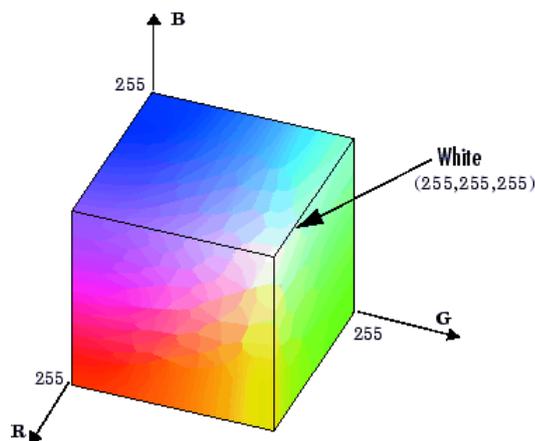
**Figure 1.8.2** : Image en niveaux de gris et sa matrice de valeurs de niveaux d'intensité

**a. Pixel :** Le pixel est souvent abrégé *p* ou *px*. C'est l'unité de base permettant de mesurer la définition d'une image numérique matricielle. Son nom provient de la locution anglaise *Picture element*, qui signifie « élément d'image ». [1]

### 1.8.3 Le modèle RVB (RGB)

L'image RVB ou RGB (*Red*, *Green* et *Bleu* en Anglais) a 3 niveaux d'intensité ; rouge, vert et bleu, de sorte que chaque pixel de l'image a 3 valeurs et ces valeurs décrivent la couleur du pixel.

Le modèle RVB est le modèle idéal pour expliquer l'addition de la luminosité dans les mélanges de couleur, encore appelée « la synthèse additive ». Le modèle RVB repose sur trois couleurs primaires, qui par leurs mélanges vont restituer toutes les autres. Il a été largement exploité dans l'industrie dès lors qu'on s'est rendu compte que l'œil utilise ce modèle colorimétrique. Le principe est simple : si on éclaire le projecteur rouge, l'écran est rouge ; si on éclaire à la fois le projecteur rouge et le projecteur vert, l'écran est jaune (et deux fois plus lumineux). Et si les trois projecteurs (rouge, vert et bleu) sont éclairés, la neutralité de la couleur s'impose avec du blanc (ou avec du gris si les projecteurs ne sont pas très puissants).



**Figure 1.8.3 :** Le modèle de couleur RVB représenté sur un cube

Les couleurs RVB sont généralement représentées comme les axes d'un cube 3D, comme illustré à la figure 1.8.3, le cube représente toutes les couleurs possibles.

### 1.8.4 Le modèle de couleurs HSV

Le HSV est une représentation alternative du modèle des couleurs RVB, conçue dans les années 1970 par des chercheurs en infographie pour mieux correspondre à la vision de la couleur. Le HSV utilise comme variables les notions de :

- ✓ **Teinte** (en anglais *Hue*)
- ✓ **Saturation** ou **Chroma** (idem en anglais)
- ✓ **Luminosité** ou **Valeur** ou **Intensité** (en anglais *Ligntness* ou *Value* ou *Intensity*).

C'est un espace tridimensionnel (dans la mesure où il comporte trois variables), mais il ne s'agit certainement pas d'une représentation en cube, mais d'un cône comme le montre la figure 1.8.4.

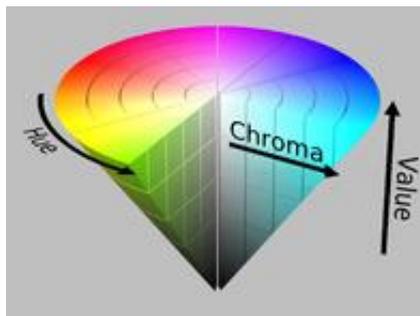


Figure 1.8.4 : La cône du modèle de couleur HSV

### 1.8.5 Système de vision

L'expression « *vision naturelle* » désigne le processus de perception par l'œil. Ce processus est en fait composé à la fois de la vue et de la perception. La vue décrit le chemin suivi par la lumière (élément essentiel en vision) de la scène jusqu'à l'œil. La perception décrit la manière dont la lumière est convertie en signal électrique puis transmise par le nerf optique au système nerveux qui le traite pour l'interpréter.

On retrouve tous ces éléments en *vision artificielle*, qui réalisent finalement, un mimétisme saisissant de la vision naturelle, avec des caractéristiques différentes. En vision artificielle, la **caméra** remplace l'œil, la transmission se réalise via diverses technologies avec ou sans fil et, enfin, le traitement et l'interprétation sont réalisés

par une unité de calcul informatique. Ce qui ne change pas dans le passage de la vision naturelle à la vision artificielle, du moins dans une approche intuitive, c'est le chemin que suit la lumière, de sa source jusqu'à son entrée dans la caméra, que l'on peut diviser en trois étapes :

- Emission de la lumière à partir d'une source
- Interaction de la lumière avec l'environnement
- Réflexion de la lumière dans l'environnement

#### **a. Caméra**

Une caméra est un appareil de prise de vues destiné à enregistrer ou à transmettre des images photographiques successives afin de restituer l'impression de mouvement pour le cinéma, la télévision, la recherche, la télésurveillance, l'imagerie industrielle et médicale, ou bien pour d'autres applications, professionnelles ou domestiques. [1]

#### **b. Composition d'une caméra**

Trois éléments de base sont à noter : l'élément mécanique (*le corps de la caméra* lui-même), l'élément optique (*la lentille*) et l'élément chimique (*le Film*). Toutes les autres parties et composants fonctionnent simplement pour supporter ou améliorer l'une des fonctions principales de la caméra.

## **1.9 Conclusion**

Dans ce chapitre, on a donné un aperçu général sur la robotique : l'historique des robots, leurs structures, leurs utilisations, leurs différentes catégories, leurs classifications ainsi que leurs domaines d'application. On a également parlé de la robotique mobile et d'une brève partie portant sur la vision artificielle, ce qui va nous servir à la construction de notre bras manipulateur mobile.

# Chapitre 2 Modélisation du bras manipulateur

---

## 2.1 Introduction

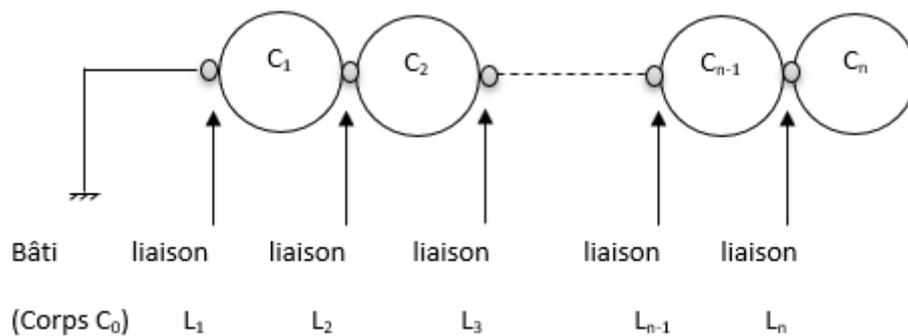
Pour développer une stratégie de commande performante pour un bras manipulateur, il est impératif de connaître la cinématique du robot considéré. Comme les bras manipulateurs sont des systèmes mécaniques poly-articulés, non linéaires, ils ont des paramètres variables et sont soumis à des perturbations externes.

Dans ce chapitre, on élabore le modèle d'un bras manipulateur à 3 *ddl* (degrés de liberté) à travers la mise en forme des équations mathématiques régissant le mouvement de ce bras.

## 2.2 Modélisation d'un bras manipulateur

### 2.2.1 Définition d'un bras manipulateur

C'est la composition de  $n$  corps mobiles rigides reliés par  $n$  liaisons rotoïdes et/ou prismatiques.



**Figure 2.2.1 :** Robot à structure simple ouverte

## 2.2.2 Nécessité d'un modèle

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques tels que :

- Les modèles de transformation entre l'*espace opérationnel* (dans lequel est définie la situation de l'organe terminal) et l'*espace articulaire* (dans lequel est définie la configuration du robot). Parmi ces modèles, on distingue :
- Les modèles **géométriques directs** et **inverses** qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement.
- Les modèles **cinématiques directs** et **inverses** qui expriment la vitesse de l'organe terminal en fonction de la vitesse articulaire et inversement.
- Les modèles **dynamiques** définissant les équations du mouvement du robot et qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, ainsi que vitesses et accélérations des articulations.

## 2.2.3 Transformations homogènes

Soient  $({}^iP_x, {}^iP_y, {}^iP_z)$  les coordonnées cartésiennes d'un point  $P$  arbitraire, mesurées dans le repère  $R_i(O_i, x_i, y_i, z_i)$ . Les *coordonnées homogènes* du point  $P$  sont  $(\omega \cdot {}^iP_x, \omega \cdot {}^iP_y, \omega \cdot {}^iP_z)$  où  $\omega$  est le facteur d'échelle, dans la robotique  $\omega = 1$ . Les coordonnées homogènes du point  $P$  sont représentées par le vecteur colonne :

$${}^iP = \begin{bmatrix} {}^iP_x \\ {}^iP_y \\ {}^iP_z \\ 1 \end{bmatrix} \quad (1.1)$$

Les coordonnées homogènes d'un plan  $Q$  de l'équation :

$${}^i\alpha x + {}^i\beta y + {}^i\gamma z + {}^i\delta = 0 \quad (1.2)$$

Exprimée selon le repère  $R_i$ , sont données par le vecteur ligne :

$${}^iQ = [{}^i\alpha \ {}^i\beta \ {}^i\gamma \ {}^i\delta] \quad (1.3)$$

Si un point  $P$  appartient à un plan  $Q$ , le produit matriciel est  ${}^iQ \cdot {}^iP$  est nul :

$${}^i Q \cdot {}^i P = [{}^i \alpha \ {}^i \beta \ {}^i \gamma \ {}^i \delta] \cdot \begin{bmatrix} {}^i P_x \\ {}^i P_y \\ {}^i P_z \\ 1 \end{bmatrix} \quad (1.4)$$

La transformation (*translation et/ou rotation*) d'un repère  $R_i$  au repère  $R_j$  est représentée par la matrice de *Transformation homogène*  ${}^i T_j$  ( $4 \times 4$ ) où :

$${}^i T_j = [{}^i s_j \ {}^i n_j \ {}^i a_j \ {}^i P_j] = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Où  ${}^i s_j, {}^i n_j, {}^i a_j$  contiennent les composants des vecteurs unitaires suivants  $x_i, y_i$  et  $z_i$  respectivement du repère  $R_j$  exprimées dans le repère  $R_i$  et  ${}^i P_j$  représente les coordonnées de l'origine  $O_j$  du repère  $R_j$  exprimées dans  $R_i$ .

On peut dire également que la matrice  ${}^i T_j$  définit le repère  $R_j$  dans le repère  $R_i$ .

La matrice de transformation s'écrit aussi sous la forme :

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

Finalement la matrice de transformation  ${}^i T_j$  :

- Est considérée comme la représentation du repère  $R_j$  dans le repère  $R_i$
- Permet le passage du repère  $R_i$  au repère  $R_j$

Soit **Trans**( $a, b, c$ ) la transformation d'une translation pure où  $a, b$ , et  $c$  sont les translations le long des axes  $x, y$  et  $z$  respectivement. Quand l'orientation est conservée, la matrice de transformation de cette translation a la forme suivante :

$${}^i T_j = \mathbf{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

La notation **Trans**( $u, d$ ) indique la translation d'une valeur  $d$  le long de l'axe  $u$ . Ainsi la matrice **Trans**( $a, b, c$ ) peut être décomposée en produit de trois matrices : **Trans**( $x, a$ )**Trans**( $y, b$ )**Trans**( $z, c$ ), par n'importe quel ordre de multiplication.

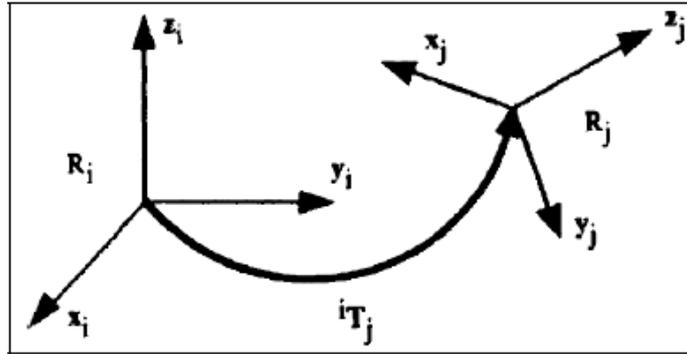


Figure 2.2.3 : Transformation de repère

Soit  $\mathbf{Rot}(x, \theta)$  la transformation correspondant à une *rotation pure* d'angle  $\theta$  autour de l'axe  $x$ , la matrice de transformation de cette rotation s'écrit :

$${}^i T_j = \mathbf{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(x, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

Où,  $\mathbf{Rot}(x, \theta)$  désigne la matrice d'orientation de dimension  $(3 \times 3)$ .

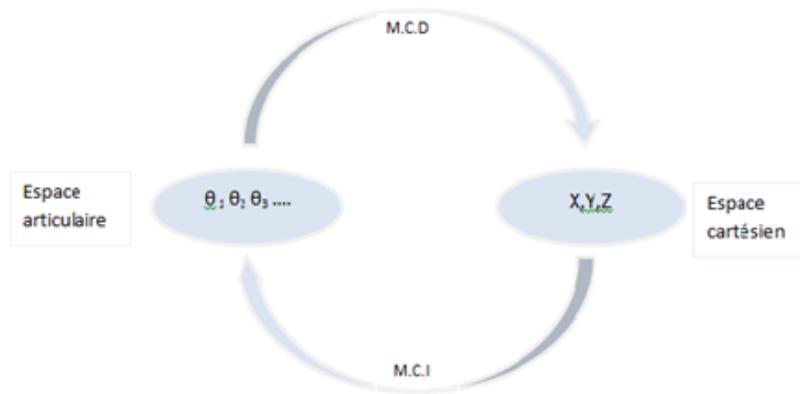
De la même façon, on définit les matrices de transformation des rotations autour des axes  $y$  et  $z$ .

$${}^i T_j = \mathbf{Rot}(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(y, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.9)$$

$${}^i T_j = \mathbf{Rot}(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{rot}(z, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.10)$$

## 2.2.4 Modélisation cinématique

Afin de calculer l'équation géométrique directe d'un bras manipulateur à chaîne ouverte simple, une méthode systématique et générale doit être appliquée pour définir la position et l'orientation relative de deux corps consécutifs. La méthode de *Denavit-Hartenberg* (DH) est la plus répandue.



**Figure 2.2.4** : Illustration du passage entre l'espace articulaire et cartésien

Le problème est comment déterminer deux repères attachés aux deux corps ? Cependant, il est commode de placer quelques hypothèses pour la définition des repères des corps.

Se reportant à la figure (Figure 2.2.1), la structure est composée de  $n$  corps  $C_i$ ,  $i = 1$  à  $n$ , plus la base ( $C_0$ ), reliés entre eux par des liaisons rotoïdes ou prismatiques, l'articulation  $i$  relie le corps  $C_i$  au corps  $C_{i-1}$ . Afin de définir la relation entre les positions du corps on associe à chaque corps  $C_i$  le repère  $R_i(O_i, x_i, y_i, z_i)$  tels que :

- L'axe  $z_i$  est porté par l'axe de l'articulation  $i$ .
- L'axe  $x_i$  est porté par la perpendiculaire commune aux axes  $z_i$  et  $z_{i+1}$ .
- L'axe  $y_i$  est formé par la règle de la main droite pour compléter les coordonnées du système  $(x_i, y_i, z_i)$ .

#### **a. Modèle cinématique direct (MCD)**

C'est l'ensemble des relations permettant de définir la position de l'organe terminal du robot en fonction de ses coordonnées articulaires.

#### **b. Convention de DENAVIT-HARTENBERG (D-H)**

D-H est la convention utilisée pour choisir le système de référence en robotique.

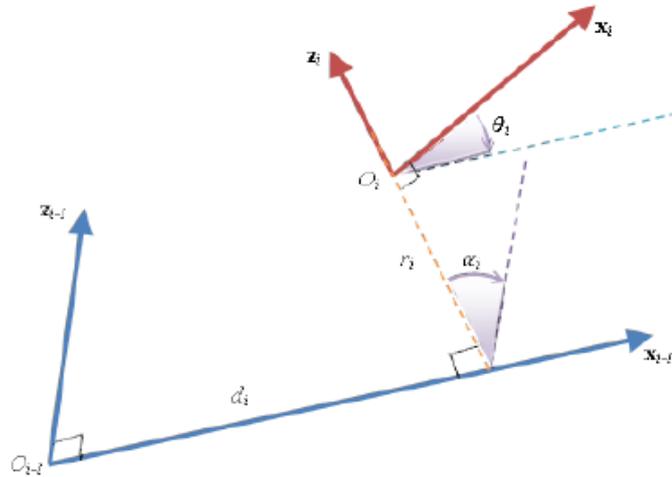
La matrice de transformation du repère  $R_{i-1}$  au repère  $R_i$  s'exprime en fonction des paramètres de D-H suivants :

$\alpha_i$  : l'angle de rotation entre l'axe  $z_{i-1}$  et l'axe  $z_i$  autour de l'axe  $x_{i-1}$ .

$d_i$  : La distance entre l'axe  $z_{i-1}$  et l'axe  $z_i$  le long de l'axe  $x_{i-1}$ .

$\theta_i$  : L'angle de rotation entre l'axe  $x_{i-1}$  et l'axe  $x_i$  autour de l'axe  $z_i$ .

$L_i$  : La distance entre l'axe  $x_{i-1}$  et l'axe  $x_i$  le long de l'axe  $z_i$ .



**Figure 2.2.4.b:** Paramètres de Denavit-Hartenberg

La variable  $q_i$  de l'articulation  $i$ , définissant l'orientation ou la position relative entre les articulations  $i - 1$  et  $i$ , est soit  $\theta_i$ , soit  $L_i$ , selon le type d'articulation (rotoïde ou prismatique respectivement). Ceci est défini par la relation :

$$q_i = \bar{\sigma}_i \theta_i + \sigma_i L_i \quad (1.11)$$

Avec :

- $\sigma_i = 0$  si l'articulation  $i$  est rotoïde
- $\sigma_i = 1$  si l'articulation  $i$  est prismatique
- $\bar{\sigma}_i = 1 - \sigma_i$

La matrice de transformation définissant  $R_i$  dans  $R_{i-1}$  est donnée par :

$${}^{i-1}T_i = Rot(x, \alpha_i) Trans(x, d_i) Rot(z, \theta_i) Trans(z, L_i) \quad (1.12)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -L_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & L_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.13)$$

### c. *Modèle cinématique inverse (MCI)*

Le modèle cinématique inverse consiste à déterminer les variables articulaires correspondant à une situation spécifique de l'organe terminal. [7]

## 2.3 Cas du robot SCARA

Afin d'appliquer les différentes notions et techniques de modélisation présentées dans ce chapitre, on présente dans cette section une application de calcul des modèles cinématiques (directe et inverse) sur un bras manipulateur SCARA.

### 2.3.1 Présentation du bras manipulateur SCARA

L'un des plus grands classiques de la robotique, le robot **SCARA**. **SCARA** signifie « **S**elective **C**ompliant **A**ssembly **R**obot **A**rm », ce qui peut se traduire par « Bras Robotisé d'Assemblage à Mobilité Sélective ». Comme son nom l'indique, c'est un robot de manutention. Il est généralement choisi pour sa grande précision, sa rapidité d'action, son faible encombrement et son prix.



Figure 2.3.1.a : Le robot SCARA

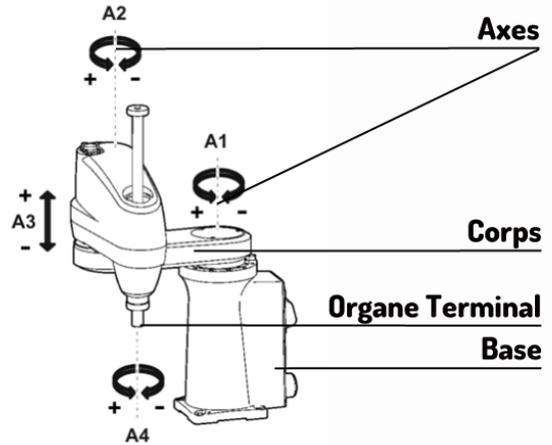


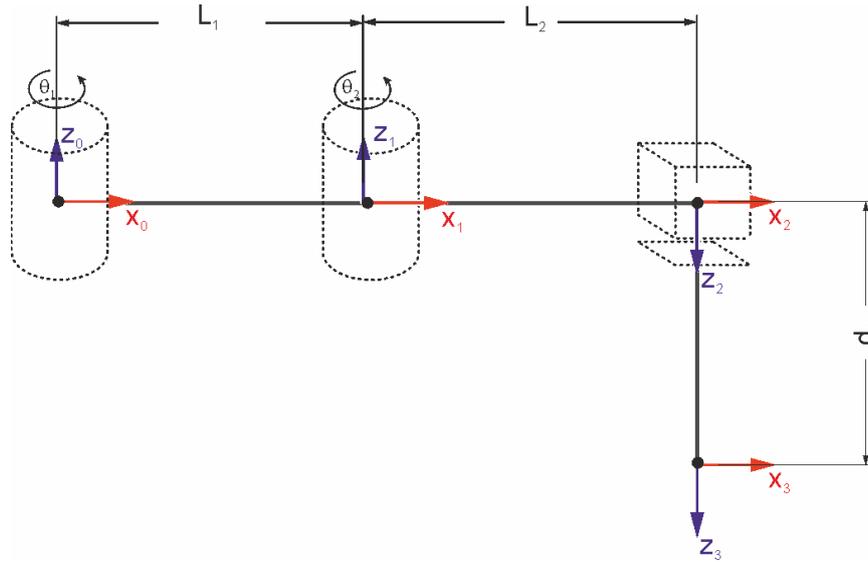
Figure 2.3.1.b : Le bras SCARA (schéma)

Dans ce cas, on a fait la conception d'un bras manipulateur (BM) de type **SCARA** constitué de 3 articulations (corps rigides) de longueurs  $L_1$ ,  $L_2$  et  $L_3$  respectivement (non nulles). Ce bras se plie dans le plan  $(x, y)$  (plan horizontal) grâce à deux liaisons pivots, il possède également un axe coulissant verticalement au biais d'une liaison prismatique l'axe  $z$  (axe vertical) contenant une ventouse et une pompe aspirante (l'effecteur final). C'est un robot à chaîne simple ouverte, articulé à **3 ddl** dont la structure est **RRP**.

## 2.3.2 Structure de l'étude

### a. Modèle Cinématique Direct du BM SCARA

Le placement des repères du bras manipulateur -en utilisant la convention de Denavit-Hartenberg- de trois degrés de liberté (**3 ddl**) est donné par la figure (Figure 2.3.2.a) qui permet de tirer les paramètres cinématiques.



**Figure 2.3.2.a.1** : Le placement des repères du robot SCARA

En utilisant l'équation (1.13), on obtient les matrices de transformation suivantes :

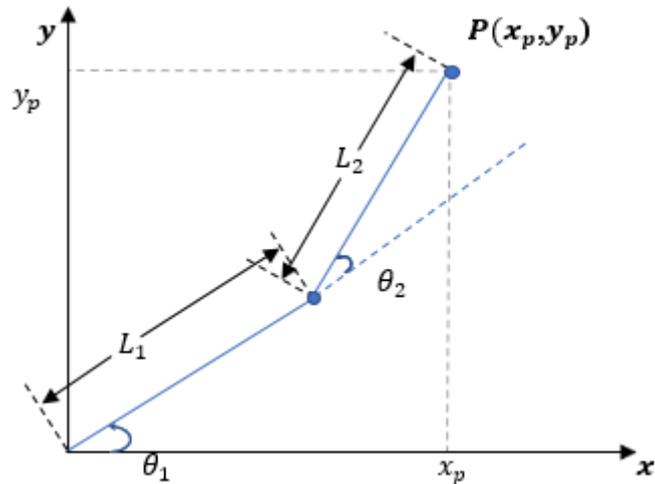
$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i & -L_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i & L_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**b. Paramètres de DENAVIT-HARTENBERG du BM**

Articulation $i$	$\theta$	$d$	$a$	$\alpha$
1	$\theta_1$	0	$L_1$	0
2	$\theta_2$	0	$L_2$	$\pi$
3	0	$d_1$	0	0

**Tableau 2.3.2.b** : Paramètres DH du robot SCARA

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ -\sin \theta_2 & -\cos \theta_2 & 0 & 0 \\ 0 & 0 & -1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



**Figure 2.3.2.a.2** : Illustration des articulations rotoïdes du BM (vue de dessus)

- **MCD** : on définit les coordonnées du point **P**

$$\begin{cases} x_p = L_1 \times \cos(\theta_1) + L_2 \times \cos(\theta_1 + \theta_2) & (2.1) \\ y_p = L_1 \times \sin(\theta_1) + L_2 \times \sin(\theta_1 + \theta_2) & (2.2) \end{cases}$$

Avec :  $\theta_p = \theta_1 + \theta_2$

### c. **Modèle cinématique inverse du BM SCARA**

- On élève au carré les équations (2.1) et (2.2), on obtient :

$$\begin{cases} x_p^2 = (L_1 \times \cos(\theta_1))^2 + (L_2 \times \cos(\theta_1 + \theta_2))^2 + 2 \times L_1 \times L_2 \times \cos(\theta_1) + \cos(\theta_1 + \theta_2) & (2.3) \\ y_p^2 = (L_1 \times \sin(\theta_1))^2 + (L_2 \times \sin(\theta_1 + \theta_2))^2 + 2 \times L_1 \times L_2 \times \sin(\theta_1) + \sin(\theta_1 + \theta_2) & (2.4) \end{cases}$$

- On fait la sommation des équations (2.3) et (2.4), on obtient :

$$x_p^2 + y_p^2 = L_1^2 + L_2^2 + 2 \times L_1 \times L_2 \times \cos \theta_2 \quad (2.5)$$

- De l'équation (2.5), on tire :

$$\theta_2 = \pm \cos^{-1} \left( \frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times L_2} \right) \quad (2.6)$$

- Et l'équation de l'angle  $\theta_1$  est la suivante :

$$\theta_1 = \tan^{-1}\left(\frac{y_p}{x_p}\right) \pm \cos^{-1}\left(\frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times \sqrt{x_p^2 + y_p^2}}\right) \quad (2.7)$$

Ainsi le modèle cinématique inverse pour les articulations **RR** est *théoriquement* donné par les équations :

$$\begin{cases} \theta_1 = \tan^{-1}\left(\frac{y_p}{x_p}\right) \pm \cos^{-1}\left(\frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times \sqrt{x_p^2 + y_p^2}}\right) \\ \theta_2 = \pm \cos^{-1}\left(\frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times L_2}\right) \end{cases}$$

### 2.3.3 Simulation avec Matlab

#### a. Boîte à outils robotique (*Robotics Toolbox*)

*Robotics Toolbox* est une boîte à outils créée par Peter Corke (Queensland University of Technology, Australia), dans des problèmes de localisation et navigation des robots mobiles.

C'est une boîte à outils MATLAB qui fournit de nombreuses fonctions qui sont utiles pour l'étude et la simulation des robots industriels (cinématique, dynamique, génération de trajectoires ...etc.) avec une méthode de représentation très générale.

Pour utiliser la boîte à outils, il suffit de copier le répertoire « **rvctools** » dans l'espace de travail et lancer le fichier « **startup\_rvc.m** » qui placera les bons répertoires dans le chemin MATLAB. [11]

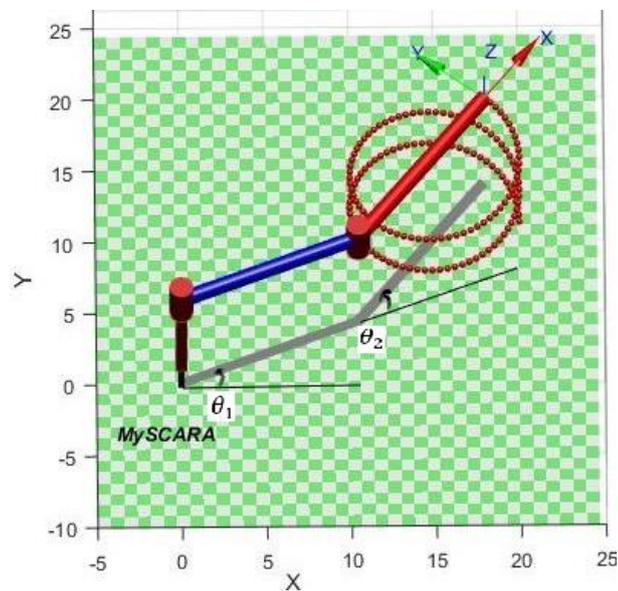
#### b. Modélisation du bras manipulateur sur MATLAB

Grâce à la boîte à outils robotique, on peut utiliser une méthode très générale pour la représentation de la cinématique ou la dynamique d'un BM de structure sérielle, comme objet MATLAB. Les objets du

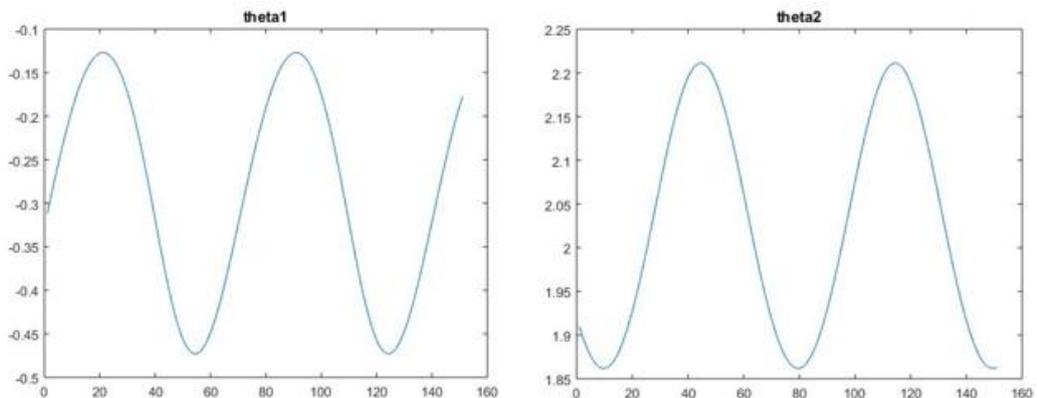
robot peuvent être créés par l'utilisateur pour n'importe quel BM de structure sérielle.

**c. Modèle cinématique inverse du bras manipulateur sur MATLAB**

Il existe une fonction sur MATLAB « **ikine** » qui calcule la cinématique inverse du bras manipulateur. Cependant, on peut aussi calculer les angles ( $\theta_1, \theta_2$ ) que forment les articulations du bras en utilisant leurs formules mathématiques (comme expliqué dans la partie précédente 2.3.2.c) et on les intègre dans un script MATLAB.



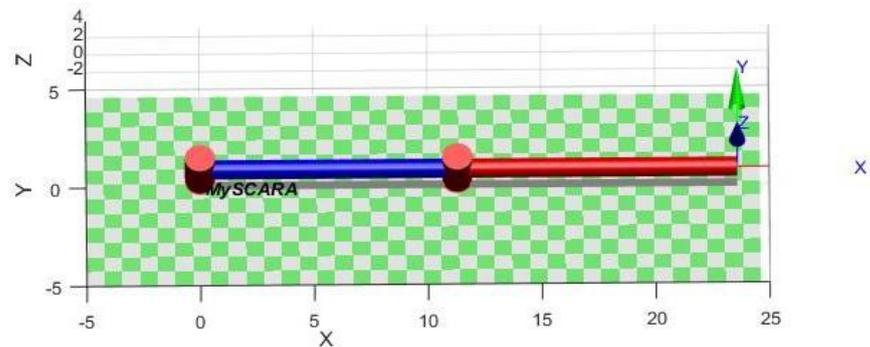
**Figure 2.3.3.c.1 :** Illustration des articulations rotoïdes du BM sur MATLAB



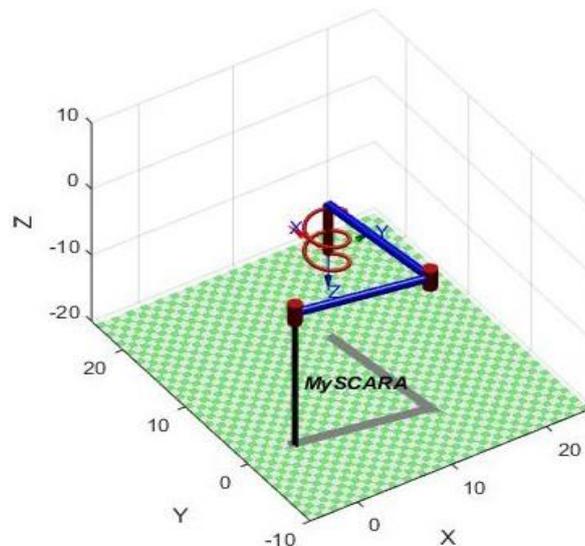
**Figure 2.3.3.c.2 :** Allures des angles  $\theta_1$  et  $\theta_2$  (respectivement) affichées sur MATLAB

Même si la programmation du bras manipulateur se fait sur Arduino (programme téléversé à la carte), on a quand même créé un programme MATLAB (script) dans le but d'établir une comparaison entre les résultats de la simulation et les résultats des tests réalisés concrètement. On a créé une fonction dans laquelle on a inséré les formules mathématiques des angles  $\theta_1$  et  $\theta_2$  puis, dans le programme on fait appel à cette fonction pour calculer ces angles (Figure 2.3.3.c.2).

Comme on a la possibilité d'avoir une animation en 3D du bras manipulateur, on a créé une boucle dans laquelle on insère des coordonnées  $x, y, z$  (choisis aléatoirement) d'un cercle pour bien visualiser les mouvements des articulations du robot manipulateur.



**Figure 2.3.3.c.3 :** Articulations rotoïdes à l'état initial (avec les coordonnées nulles)



**Figure 2.3.3.c.4 :** Articulations rotoïdes à l'état final (avec les coordonnées d'un cercle avec un mouvement continu sur l'axe z)

## **2.4 Conclusion**

Dans ce chapitre nous avons étudié le modèle cinématique du bras manipulateur SCARA. En effet, pour notre cas le modèle cinématique suffit largement parce qu'on cherche à commander la position seulement, de plus, les mouvements sont lents donc les effets dynamiques ne sont pas importants.

# Chapitre 3 Description de la partie matérielle du robot

---

## 3.1 Introduction

Dans la constitution d'un robot, la partie matérielle est aussi importante que la partie logicielle et parmi les difficultés rencontrées lors de la conception d'un robot, les problèmes de la mécanique sont les plus fréquents, afin de les éviter, on doit faire le bon choix du matériel et réaliser un travail bien fini. Dans ce chapitre, on va présenter les différents outils utilisés pour la réalisation du robot.

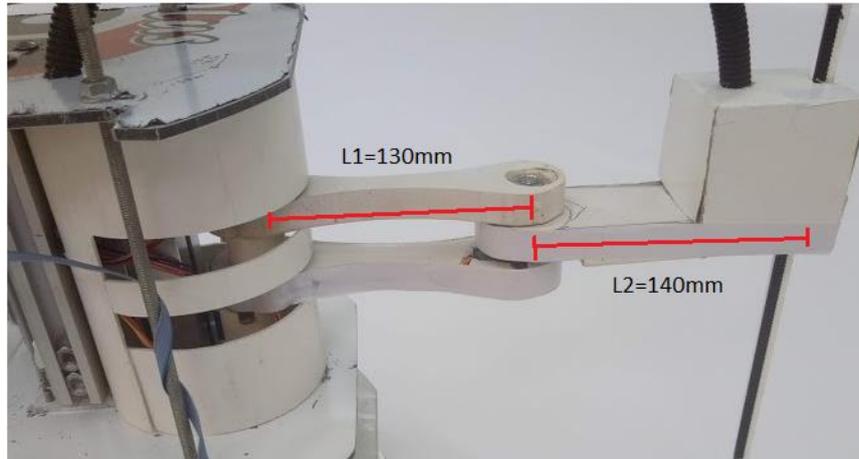
## 3.2 La partie du bras manipulateur

Avec une description générale, il ne serait pas difficile d'imaginer comment est la forme d'un bras manipulateur, cependant, bien décortiquer le robot permet de mettre les choses aux claires.

### 3.2.1 Caractéristiques techniques du bras manipulateur

Nombre d'articulation	3
Articulation ( L1 = 130mm)	Rotation de 180°
Articulation ( L2 = 140mm)	Rotation de 360°
Articulation (L'axe z)	Translation
Actionneurs	Moteurs pas-à-pas

**Tableau 3.2.1** : caractéristiques techniques du bras manipulateur SCARA



**Figure 3.2.1** : Image réelle du bras manipulateur SCARA (conçu)

### 3.2.2 Actionneurs

Les actionneurs sont les « muscles » des manipulateurs. Le contrôleur envoie des signaux aux actionneurs, qui, à leur tour, mobilisent les articulations des robots. Les types communs des actionneurs sont : les servomoteurs, les moteurs pas-à-pas, les actionneurs pneumatiques, les vérins hydrauliques...etc.

**a. Moteurs pas-à-pas** : Les moteurs pas-à-pas sont des moteurs spéciaux utilisés pour commander avec une grande précision le déplacement et la position d'un objet. Ce sont des convertisseurs électromécaniques qui assurent la transformation d'un signal électrique (impulsionnel) en un déplacement mécanique (angulaire ou linéaire). En variant la fréquence des impulsions, on peut faire tourner le moteur très lentement, d'un pas à la fois, ou rapidement à des vitesses aussi élevées que **4000tr/min**.

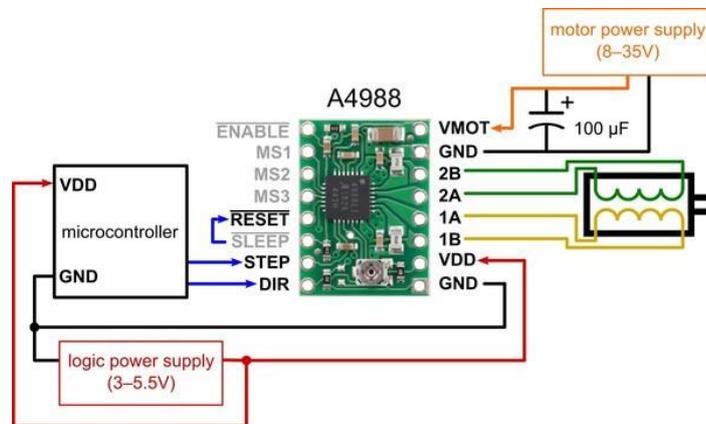
Contrairement au moteurs DC, ils ne nécessitent pas de boucle d'asservissement et sont plus simple à commander (commande en BO). Il existe trois types de moteurs : à aimant permanent, à réluctance variable et hybrides. [12]

Pour les articulations rotoïdes (les axes  $x$  et  $y$ ) du bras manipulateur conçu, on a utilisé le moteur pas-à-pas (2 pour la première articulation et 1 pour la deuxième) de référence **28BYJ-48**, C'est un unipolaire (Utilisé en bipolaire) à 4 enroulements avec réducteur et driver permettant de contrôler directement le moteur avec Arduino. Ce type de moteur peut effectuer **2048 pas/tour**.



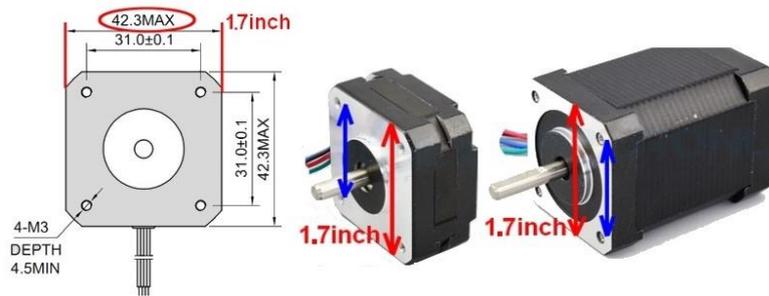
**Figure 3.2.2.a** : modèle du moteur pap utilisé

**b. Le support de pilote (Driver a4988)** : Grâce à ces drivers spécialisés, il est facile de commander ce type de moteur à l'aide d'un simple microcontrôleur (Arduino, par exemple).

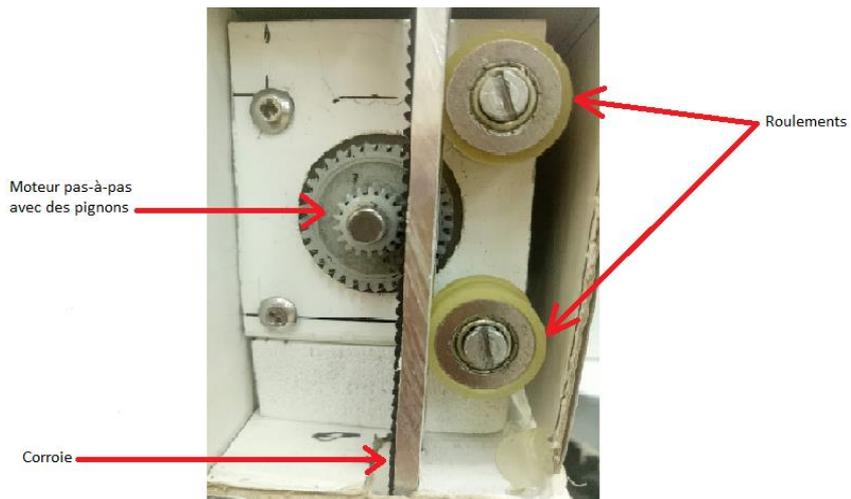


**Figure 3.2.2.b** : Le schéma de câblage pour la connexion d'un microcontrôleur et le driver a4988

**c. Le moteur pas-à-pas NEMA 17** : National Electrical Manufacturers Association, 17 signifie 1.7 pouces. On a ajouté des pignons au moteur afin de mobiliser la corroie, on a également utilisé des roulements pour fluidifier le mouvement. (Voir Figure 3.2.2.c.2)



**Figure 3.2.2.c.1** : Moteur pap NEMA-17



**Figure 3.2.2.c.2** : Effecteur final

**d. Pompe aspirante** : C'est un dispositif gérant l'aspiration et le refoulement d'un objet (une balle dans notre cas) entraîné par un moteur DC (12V). C'est l'effecteur final de notre robot.

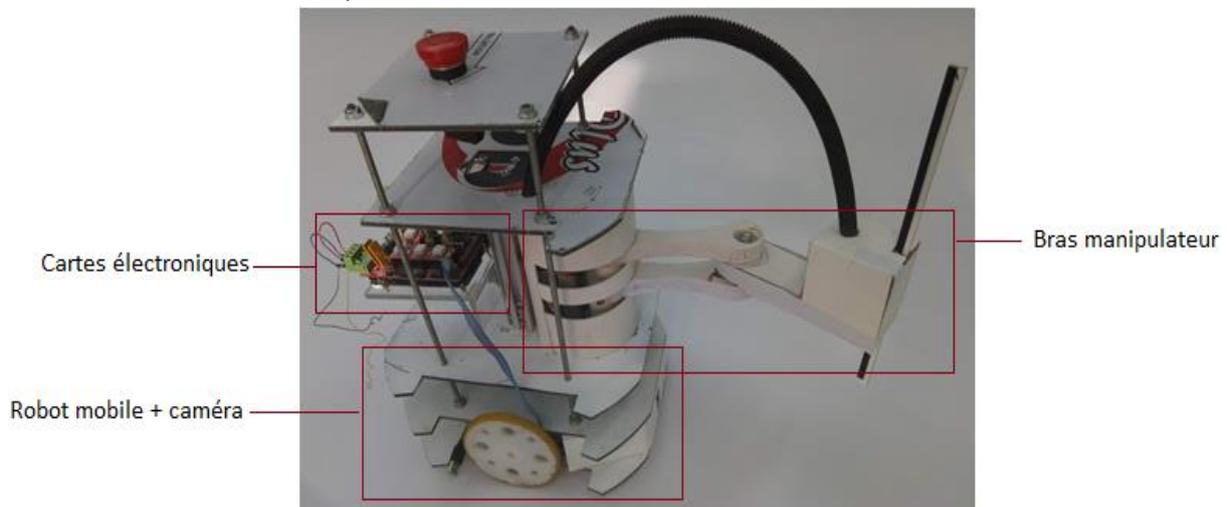


**Figure 3.2.2.d** : Micro pompe à air

## 3.3 La partie mobile

### 3.3.1 Mécanique du robot mobile

On utilise un robot mobile non holonome à **2 ddl**, c'est-à dire qu'il peut effectuer une marche avant, une marche arrière et des déviations pour changer de position (il est incapable de faire des mouvements latéraux) et cela grâce aux actionneurs utilisés (moteurs DC dans ce cas).

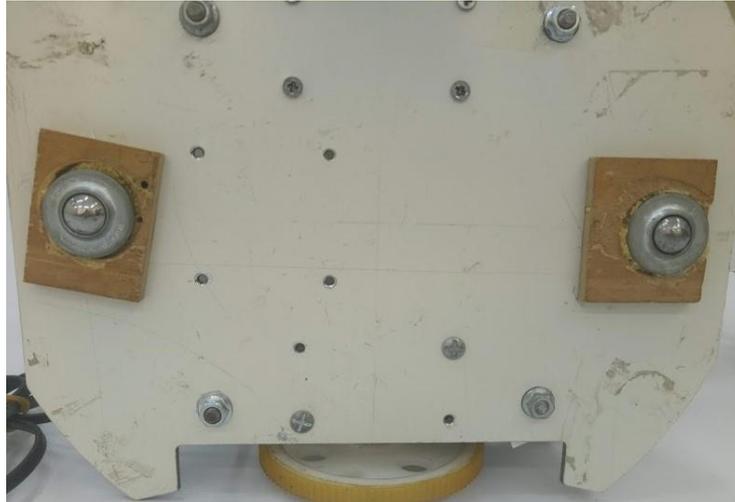


**Figure 3.3.1** : Les différentes pièces utilisées en la mécanique du robot

**La base roulante (plateforme)** : La fonctionnalité première de la base roulante est de permettre au robot d'avancer.

Les roues motrices sont reliées chacune à un moteur DC, qui reçoit des ordres de rotation (que ce soit le sens ou la vitesse directement).

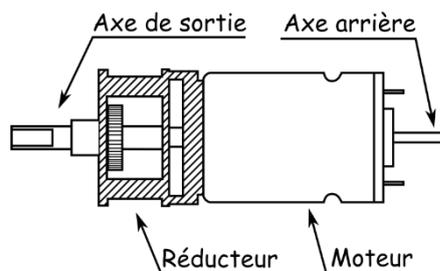
On a utilisé deux roues à bille pour garder l'équilibre de la plateforme.



**Figure 3.3.1.a** : roues à bille

**a. Moteurs DC (MCC)** : Un moteur à courant continu est constitué de deux parties électriques : le stator et le rotor. Lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur, il tourne en sens inverse.

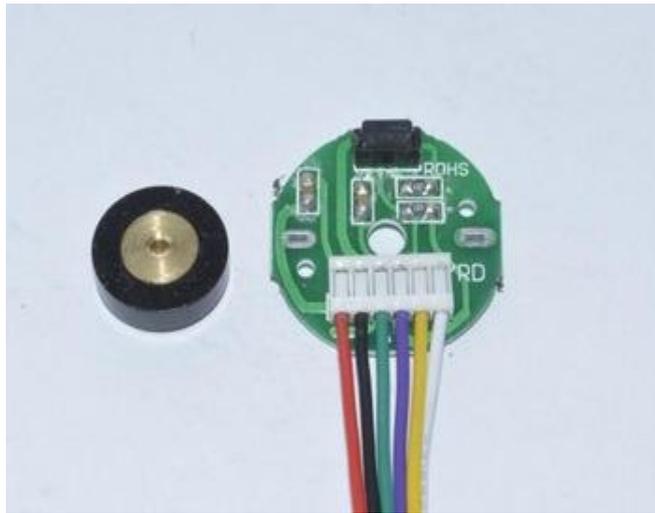
Pour les deux roues du robot mobile, on a utilisé des moteurs DC de type EMG30 (Figure 3.3.1.b.2). L'**EMG30** (moteur, codeur, boîte de vitesse 30 :1) est un moteur 12v entièrement équipé de codeur et d'un réducteur de vitesse 30 :1. Il est idéal pour les applications robotique petites ou moyennes, offrant un entraînement et un retour d'informations économiques pour l'utilisateur. Il comprend également un condensateur de suppression de bruit standard sur les enroulements du moteur.



**Figure 3.3.1.b.1** : Schéma d'un moteur type EMG30



**Figure 3.3.1.b.2** : Moteurs DC (EMG30)



**Figure 3.3.1.b.3** : Codeur incrémental

Couleur de fils	Connection
Violet (1)	Capteur Hall B $V_{out}$
Bleu (2)	Capteur Hall A $V_{out}$
Vert (3)	Capteur Hall terre
Marron (4)	Capteur Hall $V_{cc}$
Rouge (5)	+ Moteur
Noir (6)	- Moteur

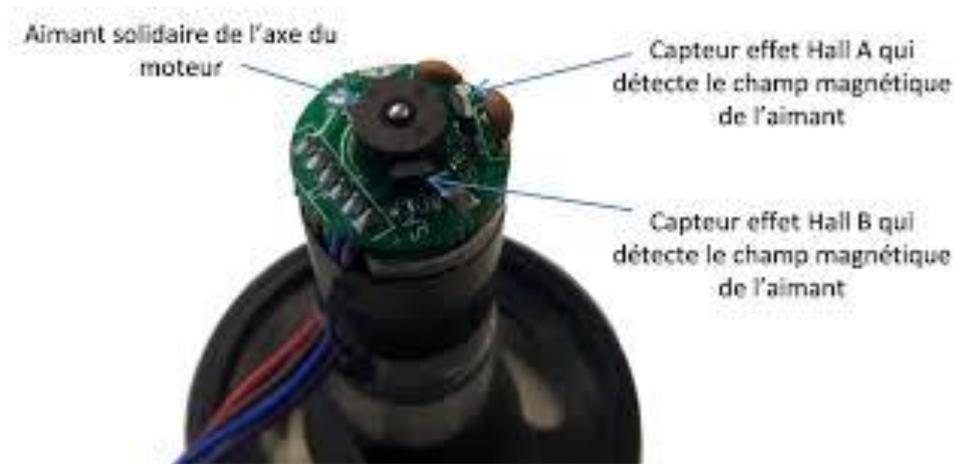
**Tableau 3.3.1.b.1** : Connecteurs du codeur

Tension nominale	12 V
Couple nominal	1.5 Kg/cm
Vitesse nominale	170 tr/min
Courant nominal	530 mA
Pas de vitesse de chargement	216
Courant à vide	150 mA
Courant de décrochage	2.5 A
Puissance nominale	4.22 W
Nombre de codeurs par tour de sortie	360

**Tableau 3.3.1.b.2** : Caractéristiques du moteur DC de type EMG30

**b. Codeur incrémental à effet Hall :** Il est constitué de :

- Un aimant solidaire du rotor du moteur
- Capteurs magnétiques à effet Hall produisant au total 360 impulsions pour chaque rotation complète du rotor. Ils sont situés sur le stator du moteur et détectent le champs magnétique tournant de l'aimant fixe sur le rotor.
- Un circuit électronique d'alimentation des constituants et de traitement (le signal est amplifié puis converti en signal carré).



**Figure 3.3.1.c :** Codeur incrémental à effet Hall

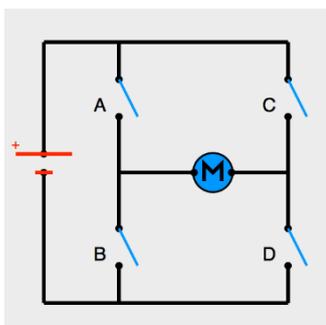
### 3.3.2 L'électronique utilisée

**a. Carte de puissance :** C'est la carte dans laquelle on trouve tous les composants électroniques utilisés pour cette réalisation. Parmi les plus importants, on trouve ce que l'on appelle un « pont en H ».



**Figure 3.3.2.a** : Carte de puissance utilisée

- **Pont en H** : Grâce au pont en H, on arrive à faire tourner le moteur dans les deux sens (en continuant de gérer sa vitesse).

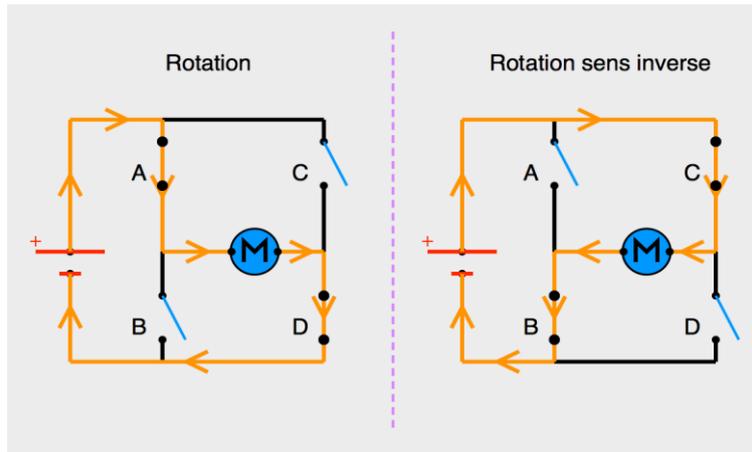


**Figure 3.3.2.a.1** : Le pont en H

Voici ce qu'on appelle un « pont en H », il tient son nom de la forme en H du circuit autour du moteur.

- **Principe de fonctionnement** : Des interrupteurs fonctionnent deux par deux. Le A est associé au D et le B est associé au C. Dans le schéma ci-dessus, rien ne se passe car tous les interrupteurs sont ouverts (ils ne laissent pas passer le courant). Le moteur est arrêté.

Voyons maintenant ce qui arrive lorsqu'on actionne en même temps les interrupteurs A et D (schéma de gauche), ou les interrupteurs B et C (schéma de droite) :



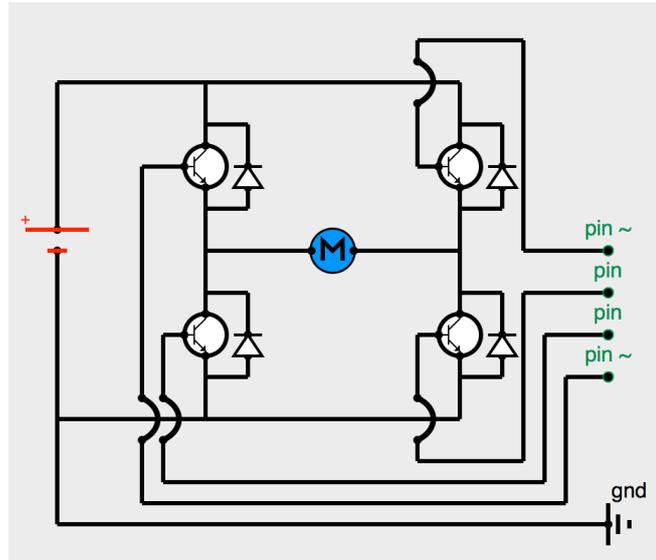
**Figure 3.3.2.a.2** : sens du courant en fonction de l'état des interrupteurs

On peut voir donc que :

- Sur le schéma de gauche : les interrupteurs A et D sont fermés. Le courant entre par la patte gauche du moteur et sort par sa droite. Le moteur tourne.
- Sur le schéma de droite : les interrupteurs B et C sont fermés. Le courant entre par la patte droite du moteur et sort par sa gauche. Le moteur tourne donc dans le sens inverse !

Un moteur lorsqu'il est en roue libre (c'est-à-dire qu'il tourne à cause de sa force d'inertie mais pas à cause du courant), il génère un courant. Ce courant peut être utilisé dans le pont en H comme frein électro-magnétique. Le moteur s'envoie son propre courant à l'envers. Cela permet de contrôler l'arrêt du moteur plutôt que de le laisser en roue libre.

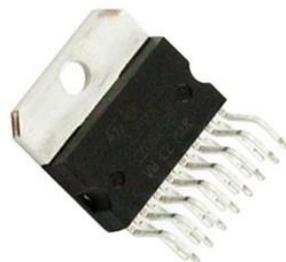
On peut réaliser un pont en H avec des transistors (NPN) et des diodes, comme suit :



**Figure 3.3.2.a.3 :** Pont en H avec transistors NPN, diodes et entrées Arduino

Mais ce n'est pas la meilleure solution car cela fait beaucoup de connexions pour commander un seul moteur DC au final. Les moteurs DC sont très utilisés en robotique, en particulier pour les déplacements. Du coup, pour nous faciliter la tâche, une puce a vu le jour dans le monde de l'électronique. Dans ce travail, on a utilisé le **L298**.

- **Module hacheur L298 :** Le L298 est un module qui dispose de 2 hacheurs 4-quadrants dans un circuit intégré, c'est-à-dire une sorte de petite boîte qui contient des composants miniaturisés et connectés entre eux afin de répondre à une fonction particulière.



**Figure 3.3.2.a.4:** L298 H bridge

- b. Arduino MEGA :** La carte Arduino Méga 2560 est basée sur un microprocesseur (ATMega2560) cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs (*Universal*

Asynchronous Receiver Transmitter). Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno. Dans ce cas, on a utilisé deux cartes, une pour la commande du bras/vision avec le « Shield » RAMPS et une autre pour le contrôle du robot mobile à l'aide d'une carte de puissance. (Comme montré sur la figure 3.3.2.b.1 ci-dessous)

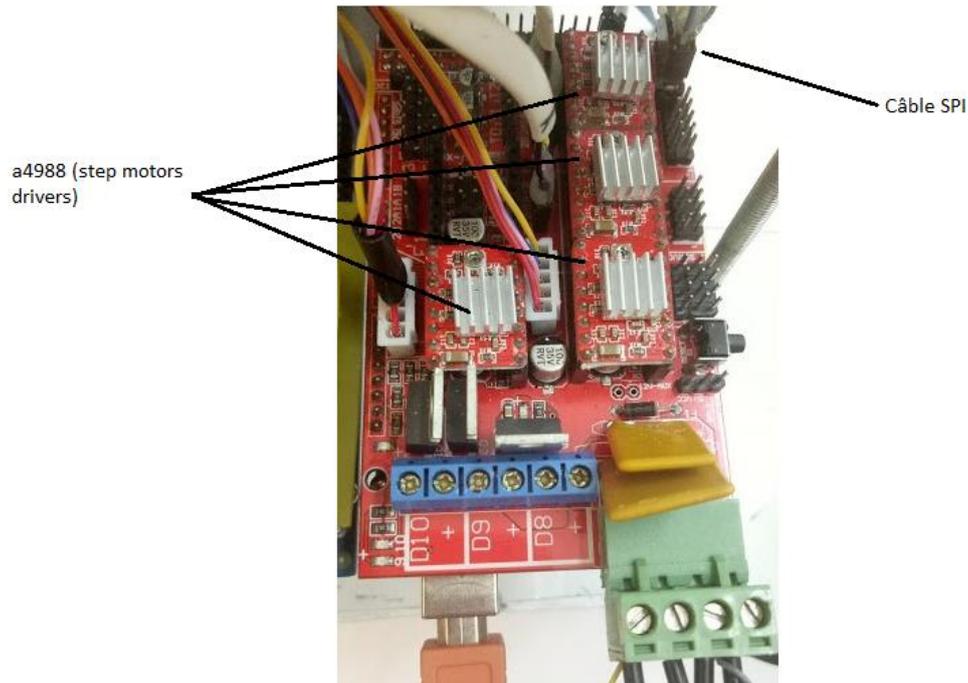


**Figure 3.3.2.b.1** : Cartes utilisées (pour la commande du robot)



**Figure 3.3.2.b.2** : Carte Arduino MEGA

**c. RAMPS 1.4** : C'est un "Shield" (carte fille d'une carte de contrôle) open-source interfaçant les éléments électroniques avec la carte de contrôle électronique Arduino 2560. On l'a utilisé parce qu'elle est bien pratique et elle contient toutes les connexions (celles du BM, de la caméra et du robot mobile).



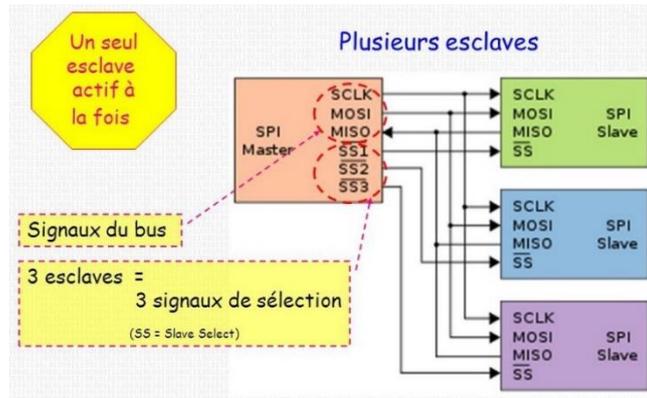
**Figure 3.3.2.c :** Carte RAMPS

- **Caractéristiques :**
  - Compatible avec Arduino 2560
  - Tension d'entrée : 12V
  - Axes pilotables : X, Y, Z + 2 extrudeurs

#### ***d. Transmission des données***

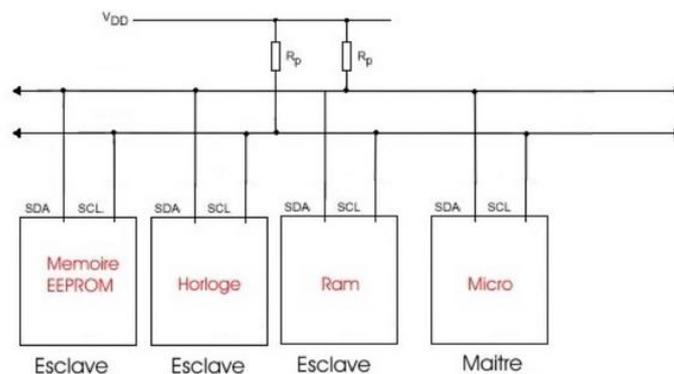
##### ***d.1 Bus SPI (Serial Peripheral Interface):***

- Bus de communication entre circuits
- Mode série synchrone
- Fonctionnement en mode « full duplex »
- De type maître/esclave
- Pas de contrôle des données ni de contrôle de flux.



**Figure 3.3.2.d.1:** Schéma de câblage d'un SPI

**d.2 Bus I<sup>2</sup>C (Inter-Integrated Circuit) :** Le bus I2c est caractérisé par une liaison en mode série réalisée à l'aide de 2 fils. C'est la société **Philips** qui a créé le concept au début des années 80. Son succès est lié à sa simplicité. Voici l'architecture type d'un bus I2C ; les données transitent par les lignes : **SDA** : signal de donnée, généré par le Maître ou l'Esclave. **SCL** : signal d'horloge généré par le Maître.



**Figure 3.3.2.d.2:** Architecture type d'un bus I2C

La communication sur le bus se fait de la manière suivante :

- Le Maître envoie sur le bus l'adresse du composant avec lequel il souhaite communiquer, chacun des esclaves ayant une adresse fixe.
- L'esclave qui reconnaît son adresse répond à son tour par un signal de confirmation, puis le Maître continue la procédure de communication (écriture/lecture).

- **La différence entre un I<sup>2</sup>C et un SPI :**

Pour le I<sup>2</sup>C, il suffit d'avoir 2 fils (clock et data), la ligne des données est bidirectionnelle. Ce n'est pas le cas pour le SPI qui comporte 4 fils (SCK, MOSI, MISO et SS), c'est une liaison point à point entre 2 périphériques avec un maître et un/plusieurs esclave(s) chaque esclave disposant d'une adresse.

	<b>I<sup>2</sup>C</b>	<b>SPI</b>
<b>Avantages</b>	<ul style="list-style-type: none"> <li>- 2 fils</li> <li>- Il peut avoir plusieurs maîtres</li> </ul>	<ul style="list-style-type: none"> <li>- La fréquence peut atteindre <b>1 MHz</b> (rapidité)</li> <li>- Il peut avoir plusieurs esclaves</li> </ul>
<b>Inconvénients</b>	<ul style="list-style-type: none"> <li>- Vitesse limitée [100KHz-400KHz]</li> </ul>	<ul style="list-style-type: none"> <li>- 4 fils (au minimum)</li> </ul>

**Tableau 3.3.2 :** Avantages et inconvénients des bus SPI et I2C (l'un par rapport à l'autre)

### 3.3.3 Asservissement du robot mobile

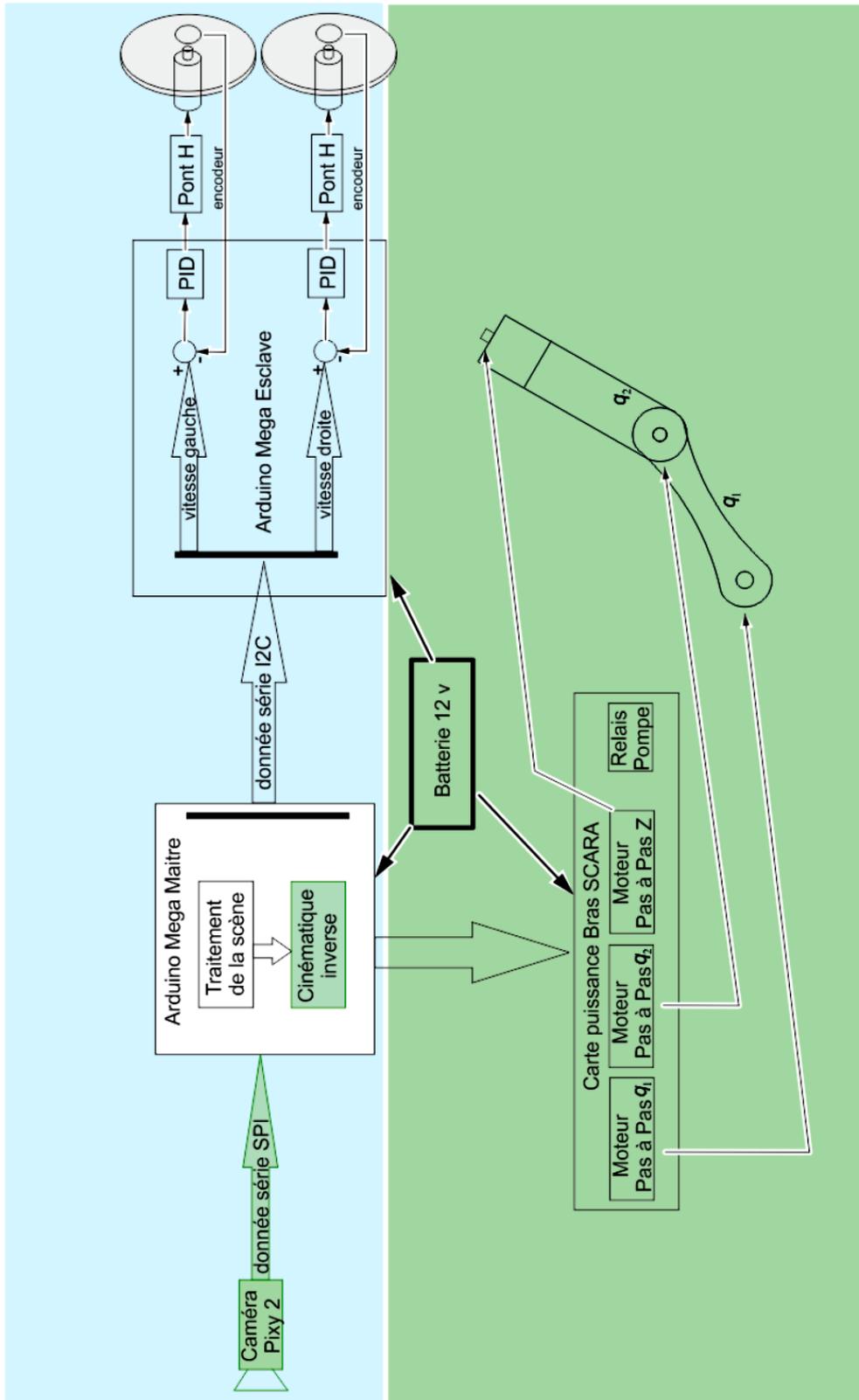


Figure 3.3.3 : Schéma synoptique global du robot

**a. Asservissement du robot mobile :** On remarque sur la figure ci-dessus (figure 3.3.3) que le robot mobile a été asservi en utilisant deux contrôleurs de type PID. Le rôle de cette commande est de faire en sorte que le robot mobile, grâce à la caméra embarquée, s'oriente vers la direction qui lui permet d'être en face de l'objet à atteindre ; si cet objet est à gauche le robot va accélérer le moteur entraînant la roue droite et décélérer celui de la roue gauche et inversement si l'objet est à sa droite l'opération inverse sera effectuée. Si l'objet est en face du robot les deux vitesses seront ajustées à une même valeur, celle-ci sera diminuée au fur et à mesure que le robot s'approche de l'objet recherché, il s'arrêtera à une distance prédéfinie.

**b. Contrôle PID :** L'algorithme de contrôle PID est utilisé pour le contrôle de presque toutes les boucles dans les industries de transformation, et il est également la base pour de nombreux algorithmes de contrôle et des stratégies avancées.

L'équation d'un régulateur PID est donnée comme suit :

$$R(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) \cdot dt + K_d \cdot \frac{de(t)}{dt}$$

- Le gain proportionnel assure une transmission instantanée du signal d'erreur (différence entre la consigne et la variable mesurée).
- Le comportement intégrateur à basse fréquence permet d'annuler l'erreur en régime permanent.
- Le comportement différentiel à haute fréquence permet d'améliorer la rapidité de la réponse du régime transitoire et limite les dépassements. Cependant, en cas de mesures bruités, il peut contribuer à déstabiliser le système.

Pour un fonctionnement correct des boucles de contrôle, la boucle PID doit être correctement réglée.

## 3.4 La partie vision

### 3.4.1 Présentation de la Pixy2

Tout comme son aîné, le capteur caméra Pixy CMUcam5, le capteur Pixy2 est compatible avec Arduino et Raspberry Pi.

Ce capteur fonctionne d'une manière très simple : il ne détecte que les images dont on a besoin et envoie les données correspondantes au processeur via 6 canaux différents : UART, SPI, I2C, USB, digital ou analogique.

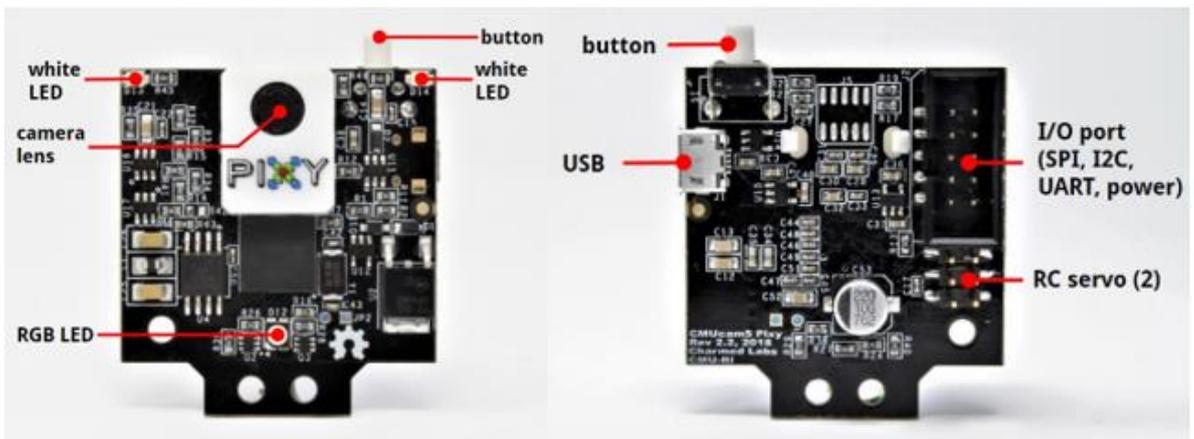
SPECS/BOARD	Pixy 2
Camera	1296x976 pixels
Number of Cores	2
Architecture	32 Bit
CPU Frequency	204 MHz
WiFi	NO
BLUETOOTH	NO
RAM	264 KB
FLASH	2048 KB
GPIO PINS	6
Busses	I2C, SPI, UART, USB
ADC Pins	0
Extras	2 Bright LEDs, Supports 2 Servos

*Figure 3.4.1.a* : Caractéristiques de la Pixy2

Comment ce capteur caméra détecte ces images ? Rien de plus simple, il utilise le principe de **reconnaissance des couleurs**. Ce capteur peut en effet détecter et enregistrer jusqu'à 7 couleurs différentes, numérotées de 1 à 7, la 1ère étant la couleur par défaut. Afin d'enregistrer les couleurs suivantes, il suffit simplement de passer l'objet que l'on souhaite être reconnu devant la lentille du capteur (une balle jaune, par exemple) tout en appuyant sur le bouton situé au sommet de la caméra.

Après avoir terminé cette petite manipulation, la caméra sera capable de reconnaître et de suivre tous les objets dont la couleur aura été mémorisée. À la différence de sa

première version, le capteur caméra Pixy2 est plus petit mais plus rapide (60 images par seconde au lieu de 50). Charmed Labs a également travaillé sur le logiciel PixyMon (version 2) afin de le rendre plus intuitif et plus réactif. Ils ont également amélioré leurs bibliothèques Arduino et Raspberry Pi en ajoutant des exemples afin de les rendre plus compréhensibles et d'étendre leurs fonctionnalités.



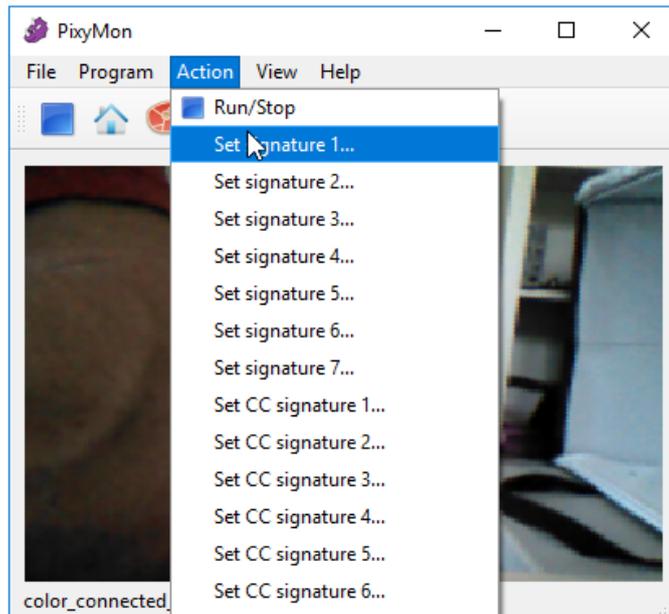
**Figure 3.4.1.b** : Le capteur caméra Pixy2

### 3.4.2 Apprentissage de Pixy2

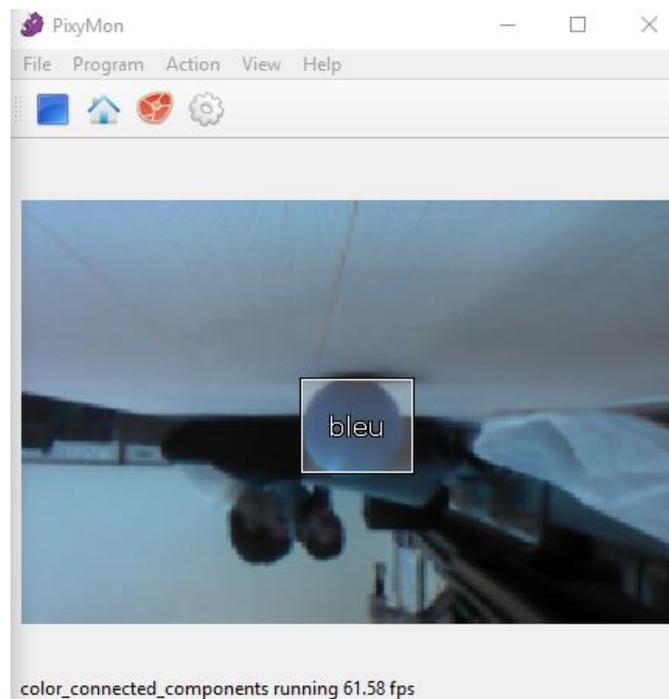
- **Le logiciel PixyMon (version 2)**

L'apprentissage des objets à Pixy2 peut se faire avec PixyMon. C'est la méthode qu'on a utilisé pour notre travail, cela est utile si l'objet est petit (petite balle dans notre cas) ou si on souhaite contrôler les pixels utilisés pour l'apprentissage. Il faut d'abord :

- Brancher le câble USB entre la Pixy2 et l'ordinateur puis exécuter PixyMon.
- Maintenir l'objet et sélectionner Action → Set signature 1 dans le menu déroulant.

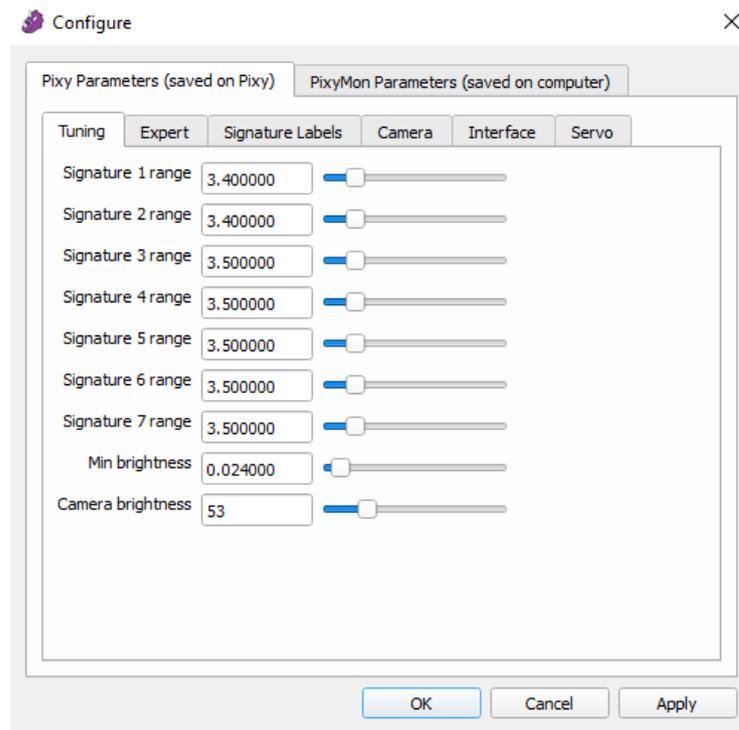


- A l'aide de la souris, cliquer et faire glisser pour sélectionner la région que Pixy2 doit utiliser pour mémoriser l'objet. Après cela, la caméra "retiendra" l'objet et le suivra automatiquement.



- **Réglages :**

Parfois les signatures de couleur qu'on fait apprendre à la Pixy2 ne sont pas tout à fait exactes, c'est-à-dire que la caméra détecte des objets qui ne sont pas ceux que l'on veut ou alors, elle ne détecte pas l'objet qu'elle a retenu, pour régler ce problème, on doit modifier les paramètres en affichant la boîte de dialogue "configure dialog" (on peut cliquer sur l'icône représentant une roue dentée ou sélectionner Fichier → Configurer).



L'apprentissage peut se faire manuellement aussi, en utilisant le bouton blanc situé au-dessus de la caméra. [13]

### 3.4.3 Applications de la Pixy2

La Pixy2 peut être utilisée pour des applications différentes, telles que :

- Reconnaissance des couleurs

- Reconnaissance des plaques de signalisation
- Détection des codes de couleurs
- Détection des codes barre
- Détection des lignes
- Détection des intersections

Dans ce cas, on a utilisé la Pixy2 pour "guider" notre robot. En captant (par reconnaissance de couleur) l'objet (une balle bleue), les données de la caméra (position, profondeur et surface de la balle par rapport au repère de la caméra  $R_c(x_c, y_c)$ ) sont transmises à travers le câble SPI au microcontrôleur (Arduino) qui commande le bras manipulateur et le robot mobile (via I<sup>2</sup>C).



**Figure 3.4.3 :** L'emplacement de la caméra dans le robot mobile

### 3.5 Conclusion

Dans ce chapitre, on a présenté les différentes parties (mécanique et électronique) de notre robot en montrant les outils utilisés dont le choix a été fait au fur et à mesure avec les tests. Par exemple, pour l'axe  $z$  du BM on a effectué beaucoup de changements ; au début on l'avait fixé à la base et on a

utilisé un moteur DC mais le mouvement était très lent et pas en cohérence avec les autres axe  $x$  et  $y$  (articulations) alors, on a changé l'emplacement (de la base à l'organe terminal) et pour plus de précision on a opté pour un autre type de moteur PAP tout en tenant compte du poids des articulations qui n'est pas très grand.

Le matériel choisi répond au besoin (cahier de charge), à la disponibilité et aux coûts.

# Chapitre 4 Tests et résultats

---

## 4.1 Introduction

Après l'étape de la conception du robot, nous allons dans ce chapitre, décrire la phase de réalisation. Nous allons présenter, en premier lieu, les méthodes utilisées pour calculer les angles de rotations des articulations du bras manipulateur et aussi la calibration des coordonnées de l'objet capté par la caméra, pour les transformer en coordonnées dans le repère du bras. Nous allons ensuite donner un aperçu sur le travail accompli à travers des captures d'écran et des résultats de simulation.

## 4.2 Commande du bras manipulateur

### 4.2.1 Validation de la cinématique inverse sur Arduino

On a choisi de commander notre BM avec une carte Arduino, alors on a créé un programme (Arduino) pour dessiner un cercle de rayon ( $r = 2$ ) centré au point de coordonnées ( $x = 18, y = 5$ ), en calculant les valeurs des angles  $\theta_1$  et  $\theta_2$  (cinématique inverse) :

$$\left\{ \begin{array}{l} \theta_1 = \tan^{-1} \left( \frac{y_p}{x_p} \right) \pm \cos^{-1} \left( \frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times \sqrt{x_p^2 + y_p^2}} \right) \\ \theta_2 = \pm \cos^{-1} \left( \frac{x_p^2 + y_p^2 - L_1^2 - L_2^2}{2 \times L_1 \times L_2} \right) \end{array} \right.$$

```

my_a988_multi_scara_ramps | Arduino 1.8.8
Fichier Édition Croquis Outils Aide
my_a988_multi_scara_ramps $

void loop() {
  //mouve (26,0);
  //elay(2000);
  /*mouve (15,-5);
  delay(2000);
  mouve (15,5);
  delay(2000);
  mouve (15,-10);
  delay(2000);
  mouve (15,10);*/
  for (i=1;i<2000;i++){
    mouve (2*cos(0.0314*i)+18,2*sin(0.0314*i)+5);
    Serial.print(th1);Serial.print('\t');Serial.println(th2);
    delay(3);
  }
  while (1);
}

```

**Figure 4.2.1.a** : Programme Arduino utilisé

Cependant, Arduino n'est pas assez puissant et a une capacité limitée alors pour vérifier si la cinématique inverse a été bien calculée, on a procédé à un test sur Matlab, sachant que sur Matlab -grâce à la boîte à outils de robotique- les calculs de la cinématique directe ou inverse se font avec grande précision.

- On a pris les valeurs des angles  $\theta_1$  et  $\theta_2$  (affichées sur le moniteur série) :

```

COM8 (Arduino/Genuino Mega or Mega 2560)
-0.28 1.02
-0.28 1.01
-0.28 1.01
-0.27 1.01
-0.27 1.01
-0.26 1.01
-0.26 1.01
-0.26 1.01
-0.25 1.01
-0.25 1.01
-0.25 1.01
-0.25 1.01
-0.24 1.01
-0.24 1.01
-0.24 1.01
-0.24 1.01
-0.24 1.02
-0.23 1.02
-0.23 1.02
-0.23 1.03
-0.23 1.03
-0.23 1.03
-0.23 1.04

```

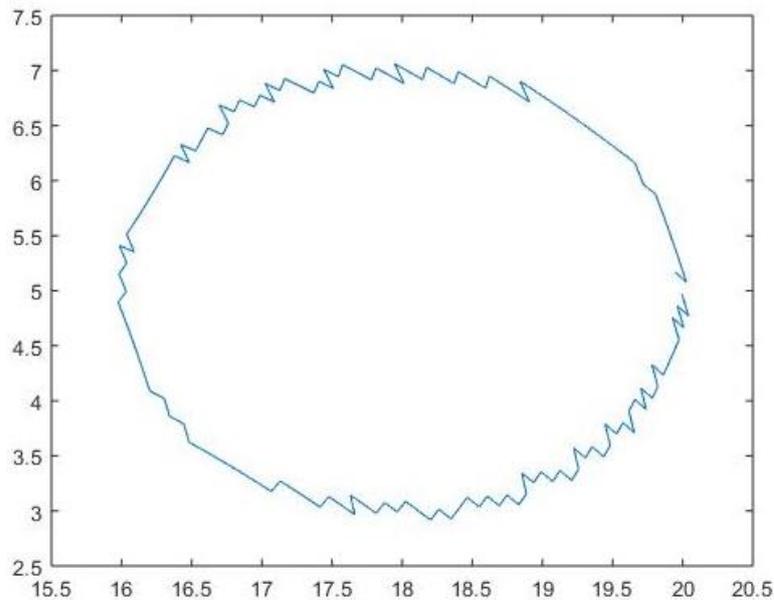
**Figure 4.2.1.b** : Valeurs des angles  $\theta_1$  et  $\theta_2$  affichées sur le moniteur série

- Ensuite, on a copié ces valeurs sur la fenêtre de commande (command window) de MATLAB en créant un vecteur contenant ces valeurs ( $u = [\theta_i \theta_j]$ ).
- Et comme on a les valeurs des angles, on a calculé  $x$  et  $y$  (La cinématique directe) en utilisant les formules démontrées dans le chapitre 2 :

$$\begin{cases} x = L_1 \times \cos(\theta_1) + L_2 \times \cos(\theta_1 + \theta_2) \\ y = L_1 \times \sin(\theta_1) + L_2 \times \sin(\theta_1 + \theta_2) \end{cases}$$

```
>> t1=u(:,1);
>> t2=u(:,2);
>> x=11.3*cos(t1)+12.3*cos(t1+t2);
>> y=11.3*sin(t1)+12.3*sin(t1+t2);
>> plot(x,y)
```

- Après affichage de la figure, on visualise un cercle comme suit :



**Figure 4.2.1.c :** Trajectoire circulaire poursuivie par le bras manipulateur

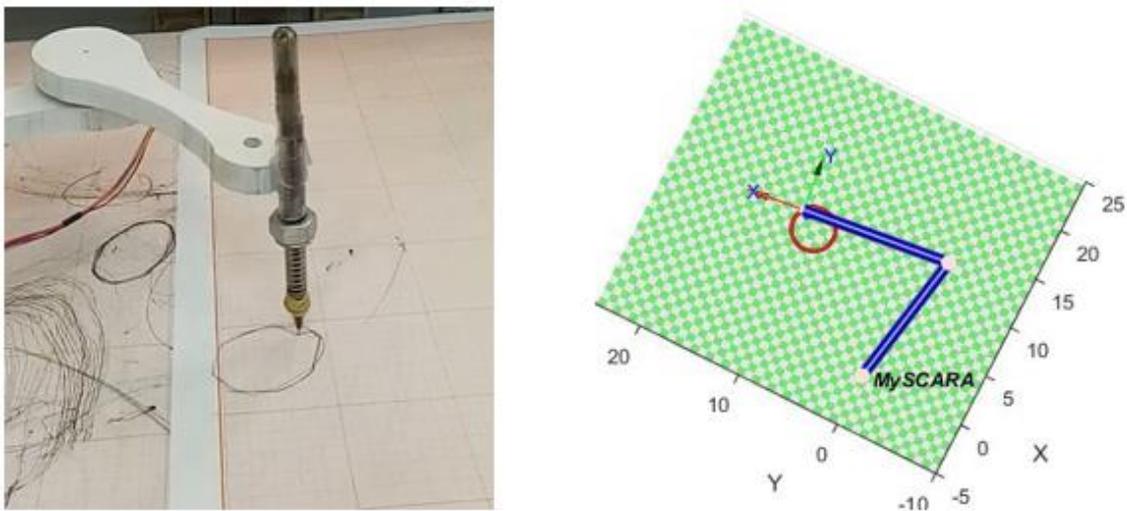
C'est exactement ce qui a été implémenté dans le programme Arduino.

Après ce résultat, on confirme que les calculs de la cinématique inverse (sur Arduino) ont été effectués correctement.

Il faut aussi tenir compte des petits mouvements non contrôlés qu'il y'a au niveau des articulations ce qui peut jouer sur la précision. La partie matérielle

(Hardware) peut être la cause mais dans ce cas on est sûre que la partie logicielle (software) fonctionne très bien.

Tel qu'il a été cité précédemment, on veut que le bras manipulateur suive une trajectoire circulaire dont les données ont été implémentées dans le programme. Pour visualiser le mouvement, on a pris une photo réelle et une capture de la figure de simulation sur Matlab.



**Figure 4.2.1.d :** trajectoire circulaire poursuivie par le BM (résultat réel et simulation sur Matlab)

#### **4.2.2 Positionnement du BM à des coordonnées préprogrammées**

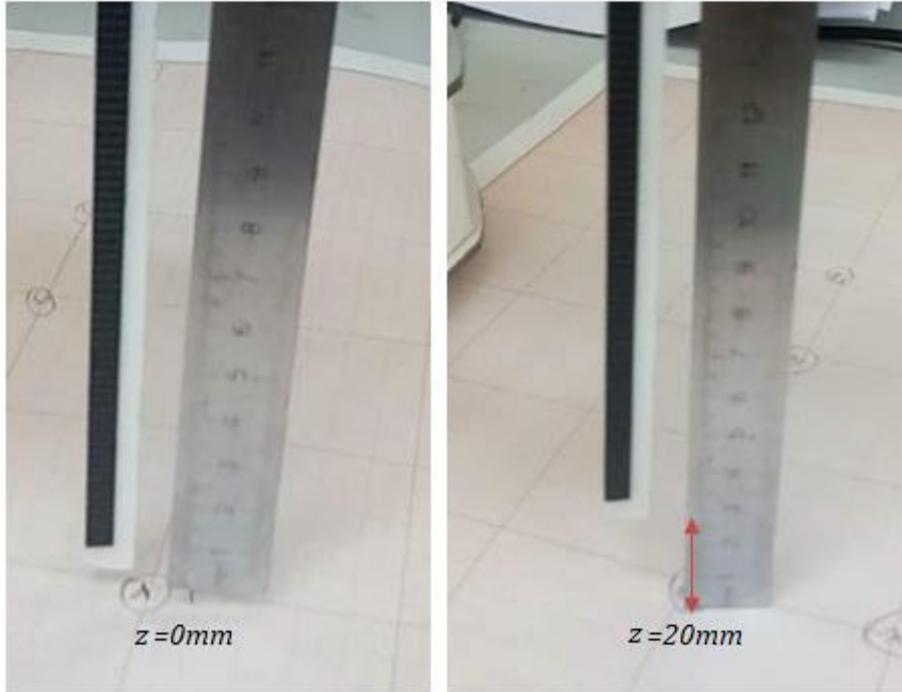
On a voulu tester la précision des mouvements des articulations de notre bras manipulateur (surtout celle de l'effecteur final) et pour cela, on a effectué quelques modifications dans le même programme utilisé précédemment (partie 4.2.1) ; au lieu de dessiner un cercle, on veut que toutes les articulations de notre BM se positionnent à des points

(coordonnées  $x, y$  et  $z$  dans l'espace 3D en mm) qu'on a choisi aléatoirement et qu'on a implémenté dans le programme en question (dans une boucle). On trouve dans le tableau ci-dessous les points choisis :

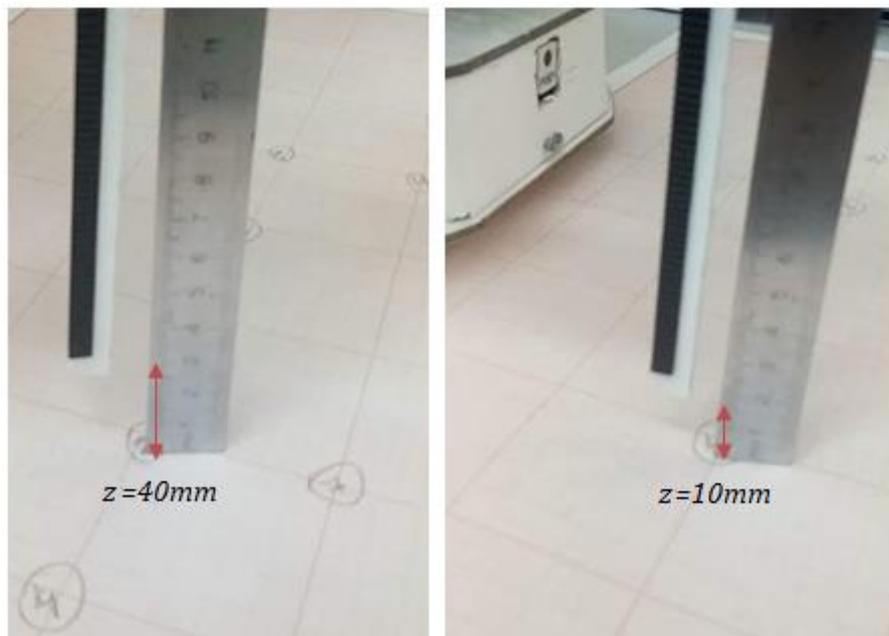
	$x$	$y$	$z$
1	256	0	0
2	150	-50	20
3	150	50	40
4	150	-100	10
5	150	100	0
6	200	100	80
7	200	-50	60
8	200	-150	40
9	200	150	15

**Tableau 4.2.3** : Coordonnées  $x, y$  et  $z$  (utilisées dans le programme)

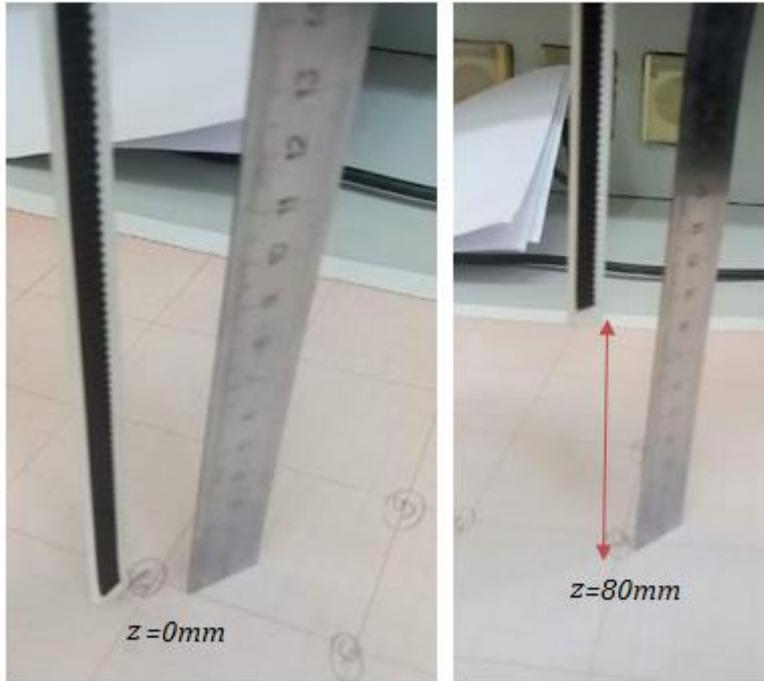
- Voici quelques photos prises lors de ce test :



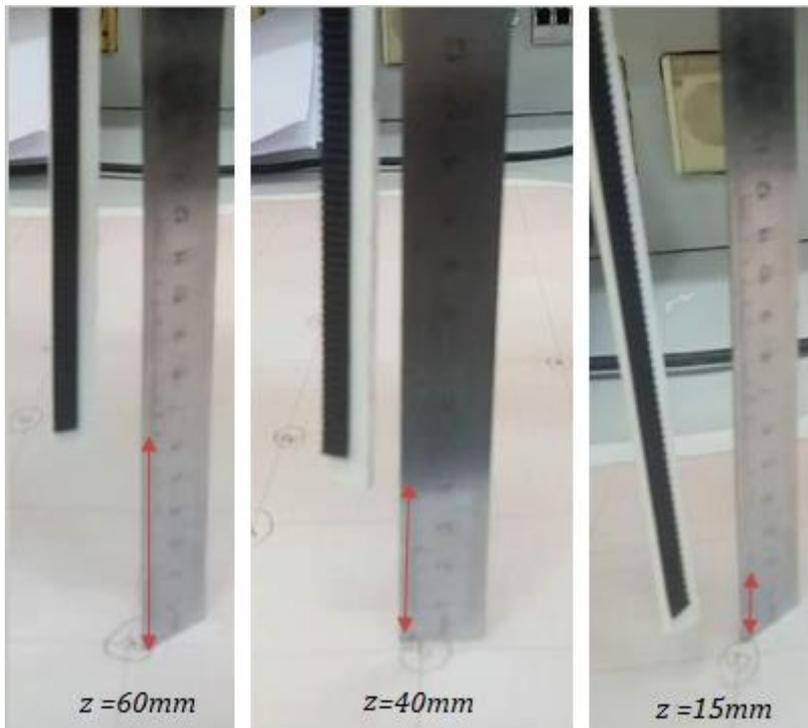
- On voit (de gauche à droite) dans cette figure l'effecteur final positionné dans le point **1** ( $256, 0, 0$ ) puis dans le point **2** ( $150, -50, 20$ ), là on peut voir le changement de position sur l'axe  $z$  ( $z = 20\text{mm}$ ). On place une règle en parallèle avec la tige de l'effecteur final pour mesurer



- Passage du point **3** ( $150, 50, 40$ ) au point **4** ( $150, -100, 10$ )

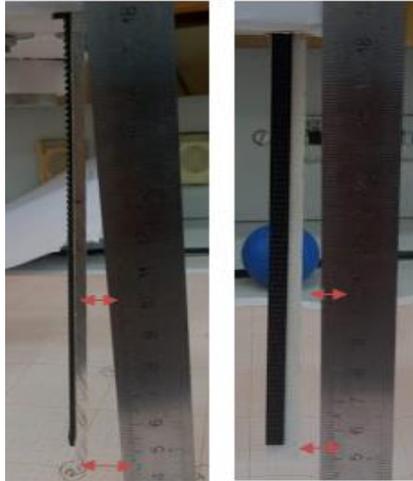


- Passage du point 5 (**150, 100, 0**) au point 6 (**200, 100, 80**).



- Passage du point 7 (**200, -50, 60**), au point 8 (**500, -150, 40**) puis finalement au point 9 (**200, 150, 15**).

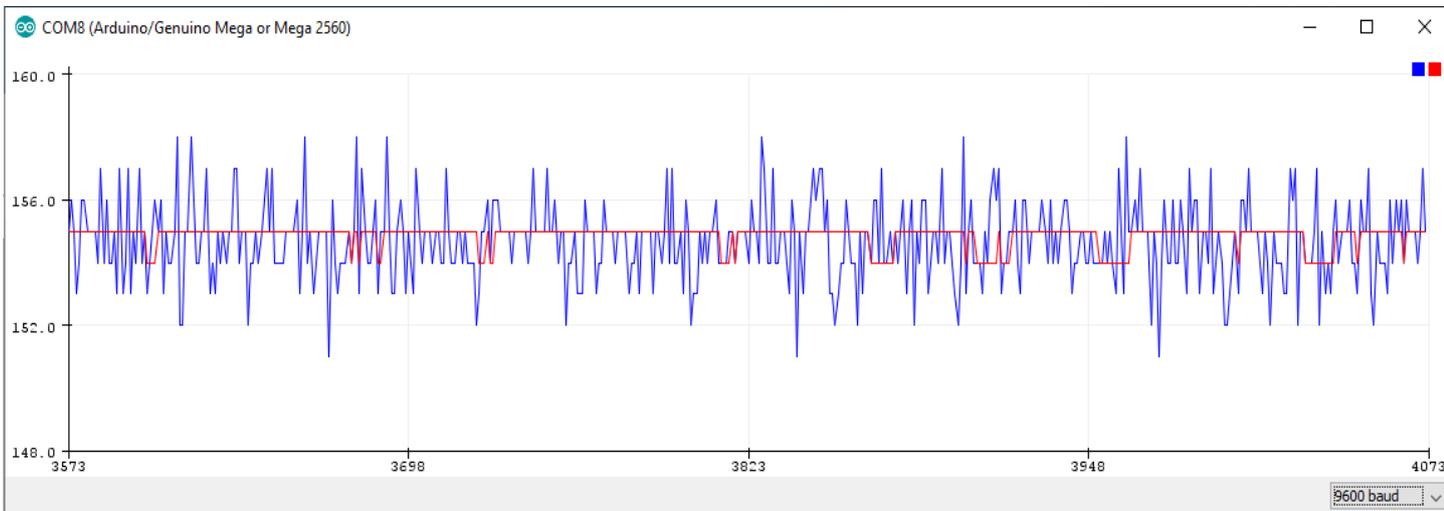
Puisque le bras a touché à tous les points donnés, on peut dire que le système fonctionne bien, sauf qu'en regardant les photos, on remarque que la tige de l'effecteur final ne se met pas en position exacte et cela est dû au fait que la tige soit un peu penchée vers l'intérieur du BM ce qui joue un peu sur la précision, mais ceci n'est pas un grand problème car on peut prendre ce petit écart en considération (en calculs).



### 4.2.3 Calibrage de la caméra

La suppression du bruit est un des traitements essentiels en vision. Pour notre travail, on a utilisé un médian qui est un filtre non-linéaire, il arrive à combiner à la fois une suppression efficace du bruit impulsionnel et la conservation des détails suffisamment importants. Le principe de ce type de filtrage est d'éliminer (dans un échantillon) toutes les valeurs maximales (trop grande) et les valeurs minimales (trop petites) et calculer la moyenne des valeurs moyennes.

On a filtré trois signaux ; la surface de la balle (captée par la caméra), les  $x$  et  $z$  du bras manipulateur.



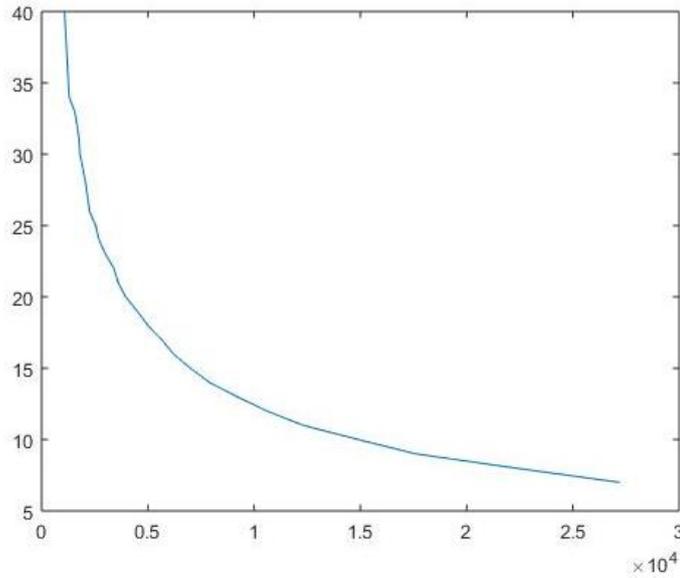
**Figure 4.2.3.a** : Filtrage médian

En captant l'objet, la caméra capte en réalité sa surface (surface = longueur \* hauteur) qui varie selon la position de la balle sur l'axe  $x$  (par rapport à la caméra). Afin de bien calibrer le capteur, on peut utiliser différentes méthodes. On a dû trouver une relation entre ces deux paramètres (surface et position de l'objet sur l'axe  $x$ ), alors on a pris des valeurs de surface à des positions différentes (33 valeurs), on a copié ces valeurs dans la fenêtre de commande (*Command Window*) sur Matlab :

```
D=[7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40]
```

```
S=[27200,22300,17600,14900,12300,10600,9200,7900,7000,6200,5650,5000,4500,3950,3600,3400,3000,2700,2540,2256,2162,2070,1936,1804,1760,1672,1554,1296,1260,1224,1188,1152,1116,1080]
```

En affichant la figure de la surface en fonction de la distance, on constate qu'il s'agit d'une fonction non linéaire et pour linéariser cette courbe on peut utiliser la méthode du calcul des multi-segments.



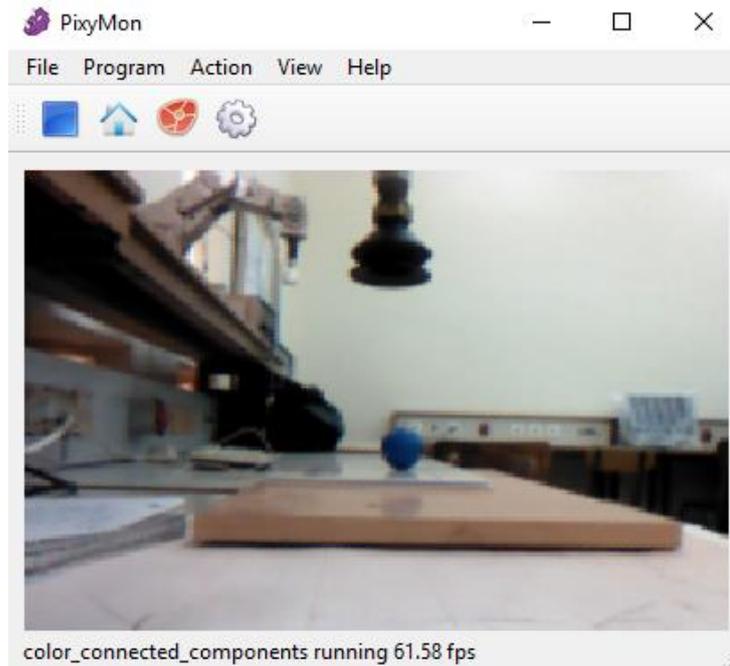
**Figure 4.2.3.b :** Courbe de la surface en fonction de la distance

En utilisant la fonction "*polyfit*" sur Matlab, on obtient un polynôme du 6<sup>ème</sup> degré (*polyfit(s, d, 6)*) comme suit :

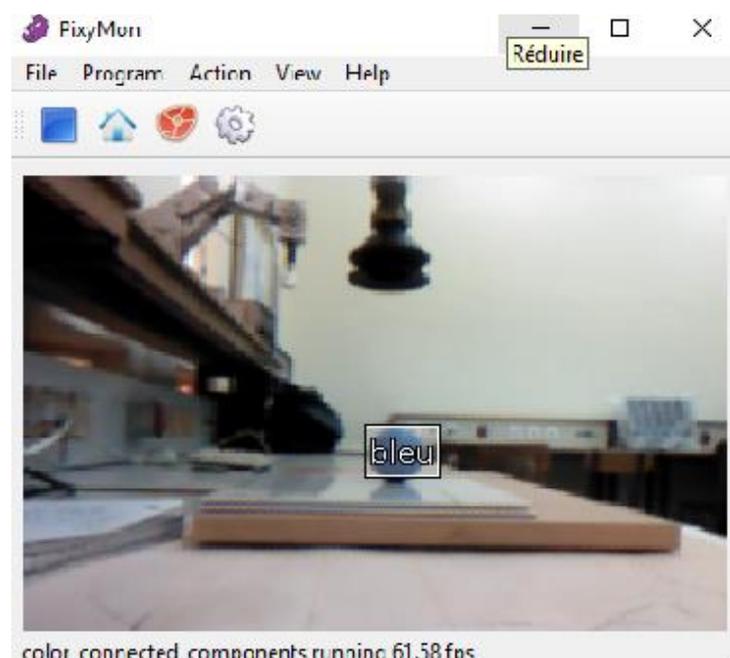
$$p = 56.1164x^6 - 0.0210x^5 + (4.6184 \times 10^{-6})x^4 - (5.3096 \times 10^{-10})x^3 + (3.1943 \times 10^{-14})x^2 - (9.5058 \times 10^{-19})x + (1.1008 \times 10^{-23})$$

On remarque qu'on arrive à une puissance de  $10^{-23}$  et cela nécessite trop de calculs pour arriver à une telle précision et éventuellement, trop d'espace pour stocker toutes les valeurs obtenues et comme expliqué dans la partie précédente (4.2.1) Arduino n'est pas assez puissant pour effectuer de telles opérations.

La caméra utilisée sert à indiquer les coordonnées de l'objet à atteindre au robot mobile et ce dernier se dirige vers l'objet en question, le suit quelque-soit sa position et s'arrête à une distance préprogrammée. Sauf que dans notre cas, on a travaillé avec une petite balle, qui, à partir d'une certaine distance n'est plus visible pour la caméra.



**Figure 4.2.3.c :** Capture de la balle à une distance de 700mm



**Figure 4.2.3.d :** Capture de la balle à une distance de 600mm

#### 4.2.4 Calibration des axes $x$ , $y$ et $z$ du bras manipulateur

Il est à noter que les axes  $x$ ,  $y$  et  $z$  du bras manipulateur doivent être calibrés pour que l'effecteur final puisse bien se positionner au point où se trouve la balle captée par la caméra. On a procédé à une sorte de

conversion. Pour transmettre les données de la caméra vers le bras manipulateur il faut faire une sorte de conversion, parce que la caméra nous donne des pixels, qui, pour le bras manipulateur devraient être des millimètres (mm).

Après plusieurs tests, on a fait une calibration comme suit :

- Pour l'axe  $x$  ( $x_{bras}$ ) :

On a affiché sur le moniteur série de Arduino, la largeur (largeur filtrée) de la balle à des positions différentes (distances) par rapport à la caméra.

Ensuite, on a copié ces valeurs sur la fenêtre de commande de Matlab :

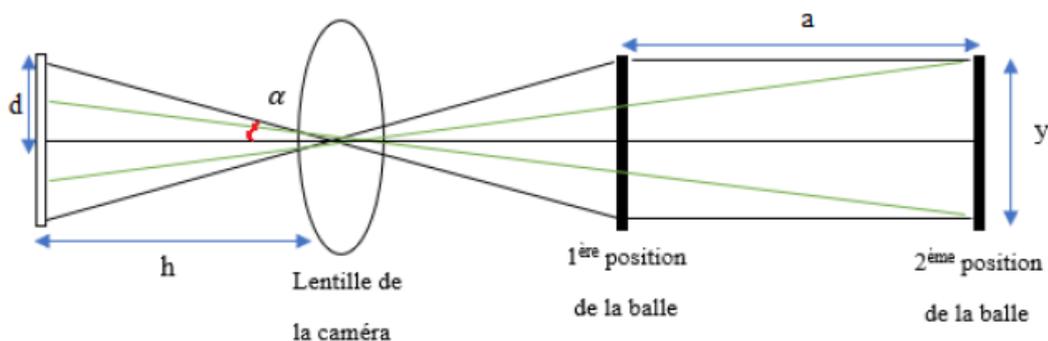
$$d = [140,150,160,170,180,190,200,210,220,230,240,250,260,270,280]$$

$$l = [140,122,116,104,100,92,86,82,76,70,68,66,64,60,56]$$

En utilisant la fonction "**polyfit**" de Matlab, on a créé un polynôme du 2<sup>ème</sup> degré :

$$x_{bras} = -5.08 \times l + 0.0177 \times l^2 + 507$$

- Pour l'axe  $y$  ( $y_{bras}$ ) :



$$\alpha = \frac{d}{h} = \frac{y}{a} \Rightarrow y = a \times \frac{d}{h}$$

Quand on change la position de la balle par rapport à l'axe  $x$  uniquement, même si on garde la même position sur l'axe  $y$ , les données que reçoit la caméra sont

différentes ce qui peut fausser les résultats. Afin de régler ce problème, il a fallu calculer  $y_{bras}$  en fonction de  $x_{bras}$  calculé précédemment, comme suit :

$$y_{bras} = (150 - x_f) \times x_{bras} \times 0.037$$

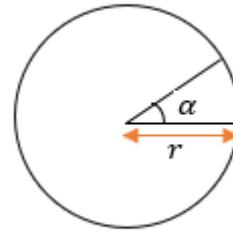
On a fait cette opération  $(150 - x_f)$  pour que l'effecteur final se mette sur le centre de la balle (juste au-dessus de la balle), car le centre de la caméra correspond à **150 pixels**, l'étendue maximale étant de **300 pixels**, on soustrait la valeur  $x$  donnée par la caméra. Le centre de la caméra est décalé du centre du bras manipulateur de **40mm**. ( $\frac{1}{h} = 0.037$ , c'est une valeur heuristique)

- Pour le  $z$  ( $z_{bras}$ ) :

Le moteur pas-à-pas utilisé effectue  $200pas/tour$  ( $200pas/360^\circ$ )

Donc, on a :  $200 \rightarrow 360$

$$Pas \rightarrow \alpha$$



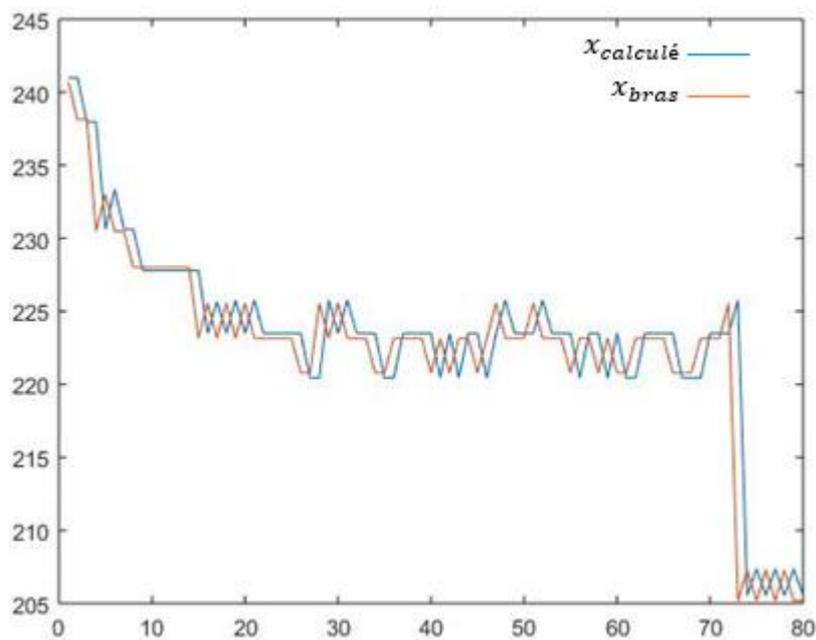
$$\text{Et : } z = r * \alpha, \text{ alors : } Pas = \frac{200}{360} * \alpha \rightarrow Pas = 0.0055 * \alpha$$

$$Pas_{(mm)} = 5.55 * z \rightarrow Pas_{(mm)} = -5.55 * z_{bras}$$

On a ajouté le signe (-) pour que l'axe se dirige vers le bas.

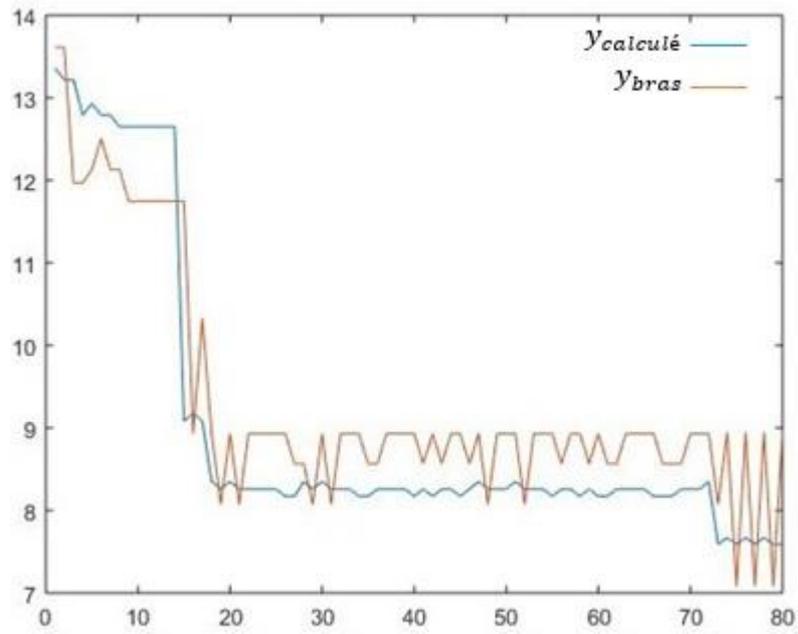
En utilisant Matlab, on a vérifié si la calibration des axes  $x, y$  et  $z$  du bras manipulateur a été bien calculée, pour ce faire on a réalisé les tests suivants :

- D'abord, on a mis le robot en marche et on a en même temps affiché les valeurs de  $\theta_1, \theta_2, x_{bras}, y_{bras}$  et  $z_{bras}$  sur le moniteur série (Arduino).
- Ensuite, on a copié ces valeurs sur Matlab. Puisqu'on a les valeurs des angles  $\theta_1$  et  $\theta_2$  on a calculé  $x$  et  $y$  du bras manipulateur en utilisant la cinématique directe.
- Enfin, on a voulu faire une comparaison entre les deux résultats (valeurs de  $x, y$  et  $z$  affichés sur le moniteur série et ceux calculés avec la cinématique directe sur Matlab), alors on a affiché les figures ci-dessous :

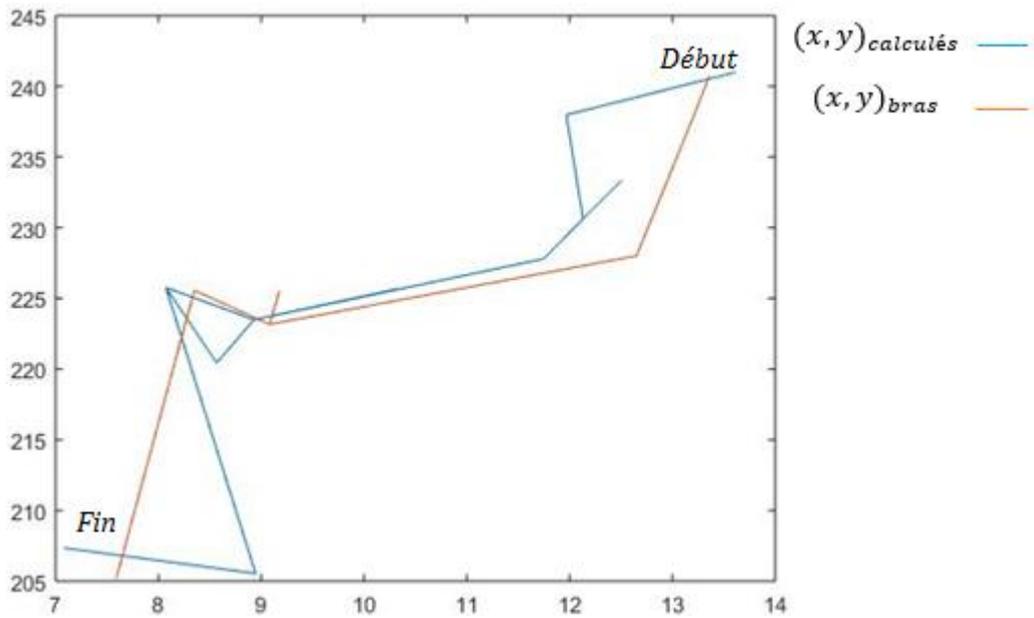


**Figure 4.2.4.a** : Allure de  $x_{calculé}$  et  $x_{bras}$

- On remarque que les deux courbes sont parfaitement superposées, ce qui veut dire que les calculs ont été bien faits.



**Figure 4.2.4.b :** Allure de  $y_{calculé}$  et  $y_{bras}$



**Figure 4.2.4.c :** Allure de  $(x_{calculé}, y_{calculé})$  et  $(x_{bras}, y_{bras})$

### 4.3 Commande du robot mobile

Tel que cela a été cité précédemment cette partie du projet a fait l'objet d'un autre mémoire de Master réalisé en 2016 [14]. Le but de ce travail était de construire un robot mobile capable de suivre un objet de couleur prédéfinie de la manière la plus fluide possible. Pour arriver à cela il a fallu déterminer les paramètres adéquats de deux contrôleurs de type PID générant les vitesses des moteurs gauche et droit pour faire avancer et orienter le robot plus ou moins vite en fonction de la position de l'objet par rapport à la face avant du robot. Les deux figures suivantes montrent les variations des consignes proposées simultanément au moteur gauche et droit en fonction de la position latérale et la surface de l'objet capté par la caméra. Le résultat de la commande proposée est de faire en sorte que le robot s'arrête à une distance correspondant, dans ce cas, à une surface de 5000 pixels, ce qui équivaut à une distance de 20 cm par rapport à la caméra.

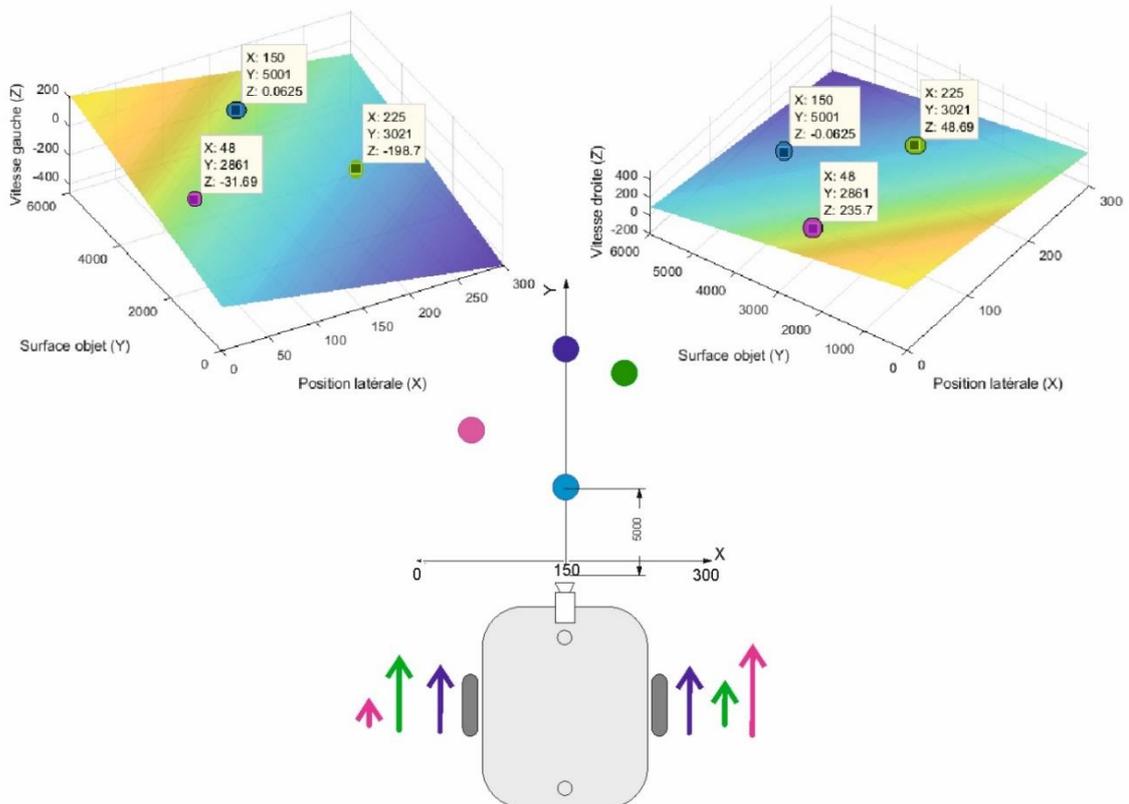


Figure 4.3 : Vitesse des roues gauche et droite en fonction de la position de la balle

La période d'échantillonnage étant de 5 ms, nous pouvons calculer la vitesse du robot de la manière suivante :

Afin d'arriver au résultat illustré par la figure 4.3 nous avons adopté un algorithme simple qui donne plus ou moins d'importance au déplacement linéaire ou angulaire selon la position de l'objet à suivre ; Si par exemple, l'objet est à droite et très proche de la caméra le robot doit avancer lentement en tournant à droite, c'est-à-dire qu'il faut augmenter la vitesse gauche et diminuer la vitesse, et parfois inverser le sens de rotation, de la roue droite, cet exemple peut être vérifié sur la figure 4.3 avec la position de couleur verte. Il faut noter aussi qu'une vitesse négative pour le côté gauche fait avancer le robot vers l'avant, ceci est du fait que les moteurs sont placés en opposition de part et d'autres du centre du robot.

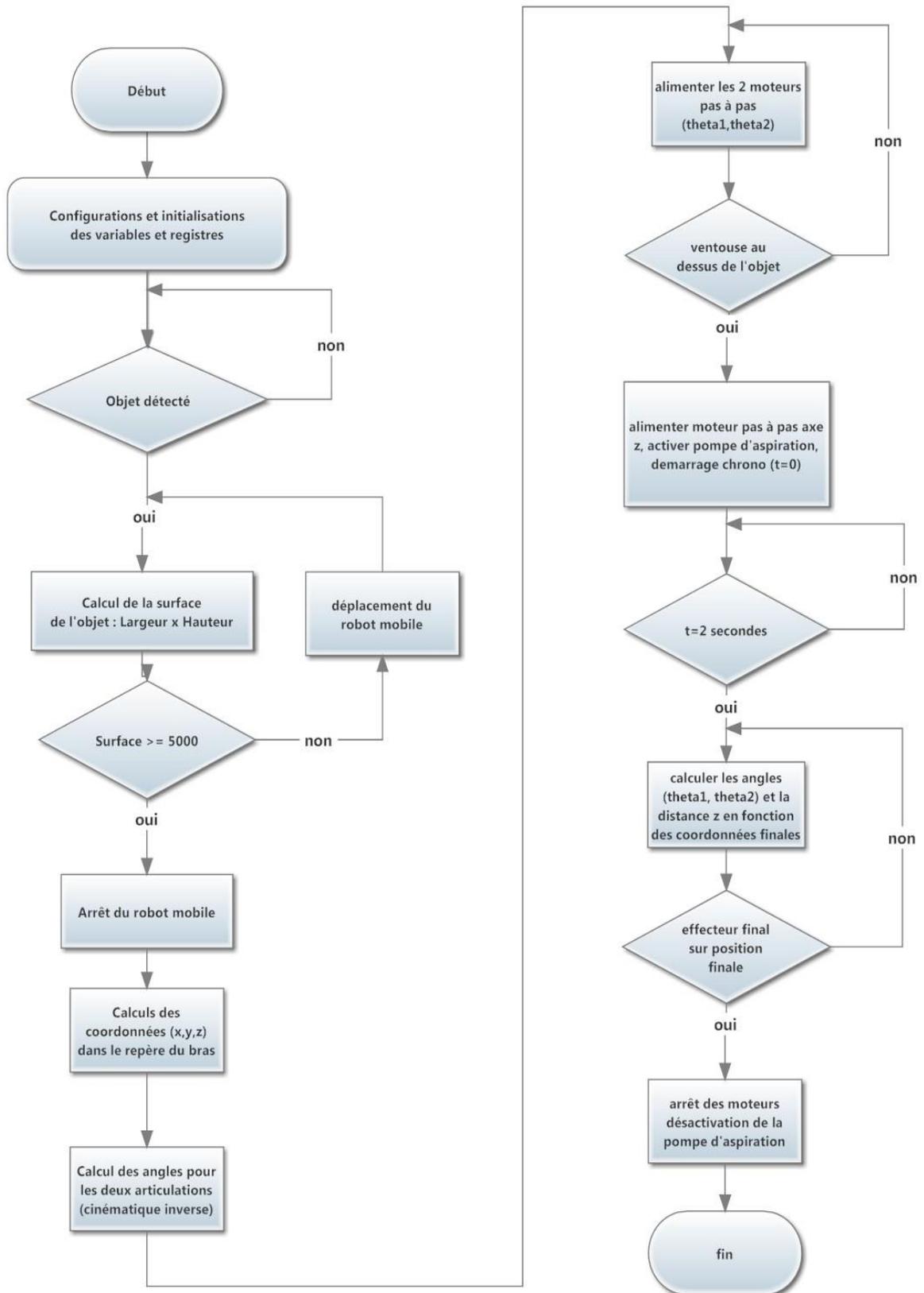
Les résultats précédents ont été obtenus en adoptant l'algorithme donnant les consignes de vitesses gauche et droite suivant :

$$vitesse_{gauche} = (150 - x) * k_1 - (5000 - surface_{objet}) * k_2$$

$$vitesse_{droit} = (150 - x) * k_1 + (5000 - surface_{objet}) * k_2$$

Les essais que nous avons effectués nous ont permis de choisir les gains  $k_1 = 1$  et  $k_2 = 1/16$ , ces valeurs ont donné un résultat satisfaisant qui font que le robot se déplace d'une manière fluide sans faire d'oscillations.

## 4.4 Organigramme du programme utilisé



## **4.5 Conclusion**

Dans ce chapitre, les résultats expérimentaux et de simulation sont présentés. On a présenté les différentes méthodes utilisées pour le calcul des angles de rotation des articulations et la calibration des axes  $x, y$  et  $z$  du bras manipulateur. La phase de réalisation a été décrite à travers les différentes captures et figures présentées.

## Conclusion générale

---

Au cours de ce projet dont le but était de réaliser un bras manipulateur mobile, nous avons abordé différents aspects relatifs à ce genre de travaux notamment l'utilisation de la vision artificielle pour faire évoluer les robots dans les environnements non structurés propre à l'homme. En effet, il fallait d'abord concevoir la partie mécanique en système modulaire et facilement démontable, elle est constituée de la plateforme mobile dont les mouvements sont assurés par des moteurs à courant continu et du bras manipulateur avec ses différentes articulations.

La particularité de notre robot réside dans le fait qu'il est doté d'une caméra servant à indiquer les coordonnées de la balle que le robot doit suivre. De ce fait, le champ de vision étant limité à  $60^\circ$  à l'horizontal et à  $40^\circ$  à la verticale, la plateforme mobile délimite alors l'espace de configuration ce qui rend le robot plus autonome.

La première partie du travail étant consacrée au bras manipulateur, on en a conçu un de type SCARA, dont les articulations sont actionnées par des moteurs pas-à-pas. Lors de la conception du bras manipulateur, on a procédé à plusieurs tests et éventuellement pas mal de modifications. En ce qui concerne les articulations, on a renforcé la première articulation rotoïde (l'axe  $x$ ) avec une deuxième articulation et un deuxième moteur en mouvement synchronisé avec le premier. Pour l'axe  $z$ , on a du changé l'emplacement comme expliqué dans le chapitre 3.

La deuxième partie du travail a été dédiée à l'étude des mouvements du robot. La vitesse du robot mobile a été asservie en utilisant deux contrôleurs de type PID et la position est obtenue grâce à la caméra embarquée ; c'est-à-dire qu'il se positionne en fonction de la position de la balle. Quant au bras manipulateur, il a fallu faire une calibration et une conversion des pixels en millimètres.

A travers les résultats obtenus et compte tenu des problèmes rencontrés, on peut considérer que le cahier de charge a été respecté d'une manière très convenable et l'objectif a été atteint. Concernant la précision des mouvements, la réponse est satisfaisante, les erreurs sont négligeables pour les applications que nous envisageons.

En industrie, on peut trouver ce modèle de bras manipulateurs mais à des prix très élevés ce qui nous motive à concevoir des robots de ce type. Pour obtenir des robots aussi robustes que ceux utilisés dans l'industrie, le premier point sur lequel il faut agir c'est le remplacement des moteurs utilisés par d'autres actionneurs plus performants. Le deuxième point concerne le microcontrôleur utilisé qui peut être remplacé par un microprocesseur plus puissant. On peut aussi munir le robot de nouvelles fonctionnalités en exploitant les autres caractéristiques de la caméra utilisée.

## Annexes

---







# Bibliographie

---

- [1] Wikipedia, site web: <https://www.wikipedia.org>.
- [2] B. Espiau : « La robotique, Histoire et perspective », « Une thèse sur le thème de la robotique ».
- [3] Nizar CHELLY : « Chapitre 1 : Généralités sur les bras manipulateurs », département du génie électrique, ISET, Kairaouan, 2013/2014.
- [4] Alain Pruski : « Robotique générale », 1988.
- [5] P. Coiffet : « Robot Hibilis », « robots sapiens », édition Herms, 1993.
- [6] Alain Liégeois : « Modélisation et commande des bras manipulateurs », « Techniques de l'ingénieur », page 2, juin 2000.
- [7] Pr. Boualem KAZED : « cours de commande des robots manipulateurs », Spécialité : automatique et systèmes. Université Saad Dahleb Blida 1, 2018/2019.
- [8] Jean Louis Boimond : « cours de robotique », ISTIA, Université d'Angers.
- [9] Extrait : « conception, modélisation et commande en robotique », site web : <https://www.techniques-ingenieur.fr>.
- [10] David Filiat: "Robotique mobile", Engineering school, ENSTA Pariteck, 2011.
- [11] Site web: [www.petercorke.com](http://www.petercorke.com)
- [12] <http://ressources.univ-Lemans.fr/AccesLibre/UM/Pedago/physique/02/electri/pasapas.html>
- [13] Site web: <https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start>
- [14] IRAIN E.M., MERAGA K., « Asservissement d'un robot mobile par vision embarquée », Mémoire de Master en Système de Vision et Robotique, Dept. Electronique, Univ. Blida, Octobre 2016.